ITISE 2019

International Conference on Time Series and Forecasting

PROCEEDINGS OF PAPERS

Volumen 2

ITISE 2019
International Conference on Time Series and Forecasting

Proceedings of Papers
25-27 September 2019
Granada (Spain)

Editors and Chairs
Olga Valenzuela
Fernando Rojas
Héctor Pomares
Ignacio Rojas

I.S.B.N: 978-84-17970-78-9
Legal Deposit: Gr 1209-2019
Edit and Print: Godel Impresiones Digitales S.L.

Preface

We are proud to present the set of final accepted papers for the 6th International conference on Time Series and Forecasting (ITISE 2019) held in Granada (Spain) during September, 25th-27th, 2019.

The ITISE 2019 seeks to provide a discussion forum for scientists, engineers, educators and students about the latest ideas and realizations in the foundations, theory, models and applications for interdisciplinary and multidisciplinary research encompassing disciplines of computer science, mathematics, statistics, forecaster, econometric, etc, in the field of time series analysis and forecasting.

The aims of ITISE 2019 is to create a friendly environment that could lead to the establishment or strengthening of scientific collaborations and exchanges among attendees, and therefore, ITISE 2019 solicits high-quality original research papers (including significant work-in-progress) on any aspect time series analysis and forecasting, in order to motivating the generation, and use of knowledge and new computational techniques and methods on forecasting in a wide range of fields.

The list of topics in the successive Call for Papers has also evolved, resulting in the following list for the present edition:

1. **Time Series Analysis and Forecasting**.
    - Nonparametric and functional methods
    - Vector processes
    - Probabilistic Approach to Modeling Macroeconomic Uncertainties
    - Uncertainties in forecasting processes
    - Nonstationarity
    - Forecasting with Many Models. Model integration
    - Forecasting theory and adjustment
    - Ensemble forecasting
    - Forecasting performance evaluation
    - Interval forecasting
    - Econometric models
    - Econometric Forecasting
    - Data preprocessing methods: Data decomposition, Seasonal adjustment, Singular spectrum analysis, Detrending methods, etc.
2. **Advanced method and on-Line Learning in time series**.
    - Adaptivity for stochastic models
    - On-line machine learning for forecasting
    - Aggregation of predictors
    - Hierarchical forecasting
    - Forecasting with Computational Intelligence
    - Time series analysis with computational intelligence

- Integration of system dynamics and forecasting models

3. **High Dimension and Complex/Big Data**.

   - Local Vs Global forecast
   - Techniques for dimension reduction
   - Multiscaling
   - Forecasting Complex/Big data

4. **Forecasting in real problem**.

   - Health forecasting
   - Telecommunication forecasting
   - Modelling and forecasting in power markets
   - Energy forecasting
   - Financial forecasting and risk analysis
   - Forecasting electricity load and prices
   - Forecasting and planning systems
   - Real time macroeconomic monitoring and forecasting
   - Applications in: energy, finance, transportation, networks, meteorology, health, research and environment, etc.

After a careful peer review and evaluation process (each submission was reviewed by at least 2, and on the average 2.9, program committee members or additional reviewer). In this proceedings we are presetting the abstract of the contribution to be presented during ITISE-2019 (accepted for oral, poster or virtual presentation,according to the recommendations of reviewers and the authors' preferences).

In this edition of ITISE, we are honored to have the following invited speaker:

1. Prof. Per Bjarte Solibakke, Professor and Associate Dean for Education, Faculty of Economics, Norwegian University of Science and Technology — NTNU Department of International Business.Vice Dean for Education, Faculty of Ecdonomics and Management, Department of International Business.

2. Prof. Thorsten Lehnert, Full professor of Finance. Luxembourg School of Finance (LSF)

3. Prof. Dieter Nautz, Professor Freie Universitt Berlin. Fachbereich Wirtschaftswissenschaft. Chair of Econometrics.

4. Prof. Dr. Stephan Schlter, Professor. Faculty of Mathematics, Natural and Economic Sciences University of Applied Sciences Ulm .

5. Prof. J. Hinaunye Eita, Professor and Head of Academics School of Economics College of Business and Economics . University of Johannesburg .

This new edition of ITISE was organized at the Universidad de Granada, with the help of the Spanish Network Time Series (RESET). We wish to thank to our main sponsor the institutions Faculty of Science, Dept. Computer Architecture & Computer Technology and CITIC-UGR from the University of Granada for their support. We wish also to thank to the Dr. Veronika Rosteck and Dr. Eva Hiripi, Springer, Associate Editor, for their interest in the future editing a book series of Springer from the best papers of ITISE 2019.

We would also like to express our gratitude to the members of the different committees and to the reviewer for their support, collaboration and good work.

September, 2019
Granada

ITISE Editors and Chairs
Olga Valenzuela
Fernando Rojas
Hector Pomares
Ignacio Rojas

.

Program Committee

Tatyana Afanaseva	Ulyanovsk State Technical University
Dorel Aiordachioaie	University Dunarea de Jos of Galati
Cagdas Hakan Aladag	Hacettepe University
Jose M. Amigo	Universidad Miguel Hernandez
Josu Arteche	University of the Basque Country UPV/EHU
Marcel Ausloos	GRAPES
Rosangela Ballini	IE - DTE - UNICAMP
Oresti Banos	University of Granada
Josep Lluís Carrion-I-Silvestre	Universitat de Barcelona
German Castellanos	Universidad Nacional de Colombia
João P. S. Catalão	University of Porto
Miguel Damas	University of Granada
Lee Chang-Yong	Kongju National University
Marijana Cosovic	University of East Sarajevo, Faculty of Electrical Engineering
Pierpaolo D'Urso	Sapienza University of Rome
Ricardo de A. Araújo	Laboratório de Inteligência Computacional do Araripe / Instituto Federal do Sertão Pernambucano
Lola Gadea	University of Zaragoza
Alberto Guillen	University of Granada
Jesus Gonzalo	U. Carlos III de Madrid
Ferda Halicioglu	Istanbul Medeniyet University
Luis Javier Herrera	University of Granada
Tzung-Pei Hong	Department of Computer Science and Information Engineering, National Univesity of Kaohsiung
Plamen Ch. Ivanov	Boston University
Ivan Izonin	Lviv Polytechnic National University
Samrad Jafarian-Namin	Yazd University
Vinayakam Jothiprakash	
Emina Junuz	Faculty of information technologies
Sreekanth K J	KISR
Rebecca Killick	Lancaster University
Alexey Koronovskiy	Saratov State University, Faculty of Nonlinear Processes
Dalia Kriksciuniene	Vilnius University
Elmar Lang	University of Regensburg
Hooi Hooi Lean	Universiti Sains Malaysia
Thorsten Lehnert	Luxembourg School of Finance
Chunshien Li	Department of Information Management, National Central University
Carlos Lima	University of Brasilia
Hui Liu	Central South University, China & University of Rostock, Germany
Songan Mao	Purdue University
Francisco Martínez-Álvarez	Universidad Pablo de Olavide
Janusz Miśkiewicz	University of Wrocław
Antonio Montañés	
Miquel Montero	Universitat de Barcelona

Fionn Murtagh	University of Huddersfield
Guy Mélard	Université libre de Bruxelles
P. C. Nayak	National Institute of Hydrology
Juan M. Palomo-Romero	University of Córdoba
Eros Pasero	Politecnico di Torino
Fernando Perez De Gracia	Universidad de Navarra
Irina Perfilieva	University of Ostrava
Hector Pomares	University of Granada
María Dolores Pérez Godoy	Departamento de Informática. Universidad de Jaén
Vadlamani Ravi	IDRBT, Hyderabad
Antonio Jesús Rivera Rivas	Departamento de Informática. Universidad de Jaén
Paulo Rodrigues	Banco de Portugal
Ignacio Rojas	University of Granada
Heather Ruskin	Dublin City University
Kalle Saastamoinen	National Defence University of Finland
Reza Sadeghi	Department of Computer Science and Engineering, Kno.e.sis Research Center, Wright State University
Francois Schmitt	CNRS
Thanasis Sfetsos	NCSR Demokritos
Leonid Sheremetov	Mexican Petroleum Institute
Yixiao Sun	University of California San Diego
Leopold Sögner	Institute for Advanced Studies
Ryszard Tadeusiewicz	AGH University of Science and Technology, Krakow, Poland
Mohsen Talebsafa	University of Texas at Arlington
Chor Foon Tang	Universiti Sains Malaysia
Alicia Troncoso	Universidad Pablo de Olavide
Mehdi Vafakhah	Tarbiat Modares University
Olga Valenzuela	University of Granada
Dimitris Varoutas	National and Kapodistrian University of Athens, Facutly of Informatics & Telecommunications
Claudia Villalonga	Universidad Internacional de La Rioja
Martin Wagner	Faculty of Statistics, Technical University Dortmund
Michael Wolf	University of Zurich
Slawomir Zadrozny	Systems Research Institute, Polish Academy of Sciences

Table of Contents

Session A.3: Energy forecasting (Part I)

Session B.3: Forecasting theory and adjustment

Session A.4: Dimension reduction techniques in Time Series

Session B.4:Real macroeconomic monitoring and forecasting (Part I)

Session B.7: Forecasting with Many Models

Session A.8: Econometric models (Part II)

Session B.8: Financial forecasting and risk analysis

Session A.9: Forecasting Complex/Big data (Part I)

Session B.9: Vector processes

Time Series Causality Based on Complex Networks for the Study of Air-Sea and Climate-Epidemics Coupled Systems

Teddy Craciunescu[1], Andrea Murari[2] , Michela Gelfusa[3], E. Peluso[3]

[1] National Institute for Laser, Plasma and Radiation Physics, Bucharest-Magurele,,Romania
teddy.craciunescu@gmail.com
[2] Consorzio RFX, Padova, Italy
[3] University of Rome "Tor Vergata", Rome, Italy

Abstract. A recently developed measure for the characterization of interconnected dynamical systems is used for the study of several coupled phenomena related to the influence of the El Niño Southern Oscillation on other atmospheric systems and to the influence of climatic factors on malaria epidemics. The method is based on the representation of time series as weighted cross-visibility networks. The weights are introduced as the metric distance between connected nodes. This allows the representation of the adjacency matrix as an image. The structure of the networks, depending on the coupling strength, is quantified via the image entropy. The results illustrate the potential of the method for real life problems.

Keywords: Dynamical systems coupling, Climatic processes, Malaria epidemic, Complex Networks

1 Introduction

The synchronization between systems connected through some form of coupling is a common phenomenon occurring in a wide variety of fields. The identification of the existence of causal relations and the evaluation of the coupling strength is an important problem and a wide variety of methods have been proposed. A significant class of methods are based on statistical principles like mutual information [1], transfer entropy [2-3] or synchronization likelihood [4]. Phase synchronization methods represent another major option [5-6]. The cross-convergent maps method [7] uses a different approach based on the spatial vicinity of the temporal successive points in phase space. Analytical methods (see e.g. [8]) have been also proposed,

with the advantage of high-speed calculations. A comparison of several synchronization measures is reported in [9]. Recently we have proposed a synchronization measure based on complex networks [10]. In this paper we show the efficiency of this method in the study of several coupled phenomena occurring in climatology and of their influence on pandemic occurrences.

With regard to the structure of the paper, next section describes the synchronization measure method. The following Section 3 is devoted to the its application to the study of the climatic influence of El Niño Southern Oscillation and to the influence of climatic factors on malaria epidemics. Section 4 of the paper is devoted to the conclusions.

2 The analysis of synchronization experiments by using complex networks

The transformation of time series into graphs is introduced to allow the study of time series dynamics by mean of the organization of networks. The visibility graphs (VG) [11] is a popular approach based on the representation of time series using vertical bars; seeing this representation as a landscape, every bar in the time series is linked with those that can be seen from the top of the bar. The VG concept has been recently extended to the study of the coupling between time series by the cross-visibility algorithm [12]. Considering a pair of time series $\{x_i\}_{i=1,N}$ and $\{y_i\}_{i=1,N}$, where N is the total number of points in the time series, the cross-visibility network can be constructed by the following rules:

$$y_k \leq y_i + \frac{x_j - x_i}{j-i}(k-i), \quad i < \forall k < j \tag{1}$$

or

$$y_k \geq y_i + \frac{x_j - x_i}{j-i}(k-i), \quad i < \forall k < j \tag{2}$$

Time series can be graphically represented as a set of bars, where the height of the bar is proportional to the time series values. A node is inserted in the complex network for each bar. The nodes are connected if they are visible to each other by mean of the landscape created by the bars located in between, corresponding to both time series. Eq. (1) accounts for the visibility from the top view, while Eq. (2) accounts for the visibility from the beneath view. The network constructed in this way can be represented by the adjacency matrix A whose elements are:

$$a_{ij} = \begin{cases} 1, if\ nodes\ i\ and\ j\ are\ connected \\ 0, otherwise \end{cases} \tag{3}$$

The key element of our approach consists of modifying the adjacency matrix by weighting the connections with the metric distance between two connected values in the time series:

$$a_{ij}^{w} = \begin{cases} dist(y_i - y_j), if\ Eq.(1)\ or\ Eq.(2) is\ satified \\ 0,\ otherwise \end{cases} \quad (4)$$

where $dist(y_i - y_j)$ is the metric distance. The Euclidean distance could be a choice but it should be considered that it implicitly assumes that the data points are infinitely precise values. In many applications, measurements are affected by various noise sources and therefore the use of the geodesic distance on Gaussian manifolds as a metric distance provides usually significantly better results [13].
The weighted adjacency matrix (WAM) can be interpreted as an image whose structure is dependent on the coupling between the time series. The entropy of the image [14] can be used to reveal the structure changes:

$$p_i = \frac{H(i)}{\sum_i H(i)}, \quad i = 1, \dots, N_H \quad (8)$$

where H is the histogram of pixel intensities in the image, $H(i)$ is the number of pixels with a certain intensity and N_H the number of intensity bins in the image.

When the time series tends to synchronize due to increased coupling, WAM evolves to a simpler structure, which translates into a relatively monotonic decrease of the image entropy. The entropy, as a measure of the degree of complexity, can therefore be used to define a measure of synchronization:

$$Q = -H(CVN) \quad (9)$$

where the minus sign has been introduced in order to have an increase of Q with the coupling strength, coherent with most synchronization measures.

3 Application to climatic studies

3.1 Climatic influences of El Niño Southern Oscillation

The identification of causal relations between time series has become an increasing focus of interest in climatology. In particular a significant number of papers have ben dedicate to the influence of the El Niño Southern Oscillation (ENSO), the most important coupled ocean-atmosphere phenomenon causing global climate variability on interannual time scales, on various phenomena such as e.g. changes in level of atmospheric CO2 [14], rainfall-sensitive vegetation [15], rainfall and river discharge [17], global temperature variations [18].

In this paper, we are addressing first the influence of the ENSO irregular cyclicities on the high variability of rainfall and river discharge in the north-western Argentine Andes. This problem has been investigated for the first time, by mean of the cross-recurrence plots, in [17].

A standardized measure of ENSO is the Southern Oscillation Index (SOI), which gives an indication of the development and intensity of El Niño or La Niña events in the Pacific Ocean. The method used by the Australian Bureau of Meteorology is based on the mean sea level pressure (MSLP) difference between Tahiti and Darwin P_{diff}:

$$SOI = \frac{P_{diff} - P_{diffav}}{SD(P_{diff})} \tag{10}$$

where: P_{diffav} is the long term average of P_{diff} for the month in question, and $SD(P_{diff})$ is the long term standard deviation of P_{diff} for the month in question.

SOI is usually computed on a monthly basis and the data used in this paper has been retrieved from the National Weather Service Organization, Climate Prediction Center [19]. The time evolution of SOI is presented in Fig. 1.

For the assessment of the ENSO influence on local rainfall in NW Argentina, the monthly precipitation from the stations located in San Salvador de Jujuy (JUY) and Salta (SAL), which are influenced by different local winds, have been downloaded using the KNMI Climate Explorer, which is a part of the WMO Regional Climate Centre at KNMI [20]. The local precipitation data is presented in Fig. 2.

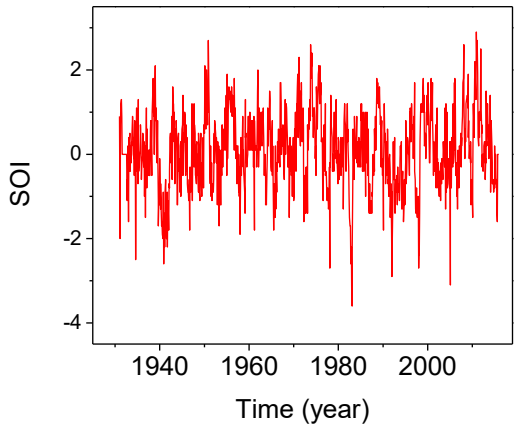


Fig. 1. Time evolution of the the Southern Oscillation Index (SOI) retrieved from the National Weather Service Organization, Climate Prediction Center [19].

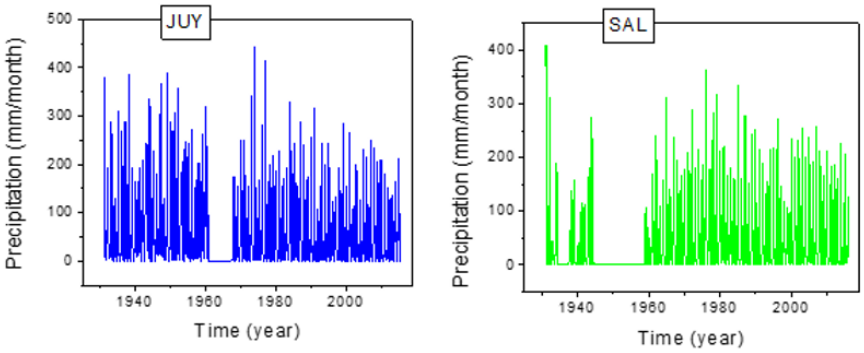


Fig. 2. Local precipitation data in San Sebastian de Jujuy (left) and Salta (right) for the time interval 1931-2016.

The ENSO time series has been used together with each of the local precipitation time series in order to construct the weighted cross-visibility networks, by means of Eqs. 1-2 and the corresponding WAM matrices. For each network the coupling measure Q, given by Eq. 9, has been calculated considering different time lags between the time series. The evolution of Q in respect with the time lag is presented in Fig. 3. The peaks appearing in this evolution reveal high similarity between the dynamics of ENSO and rainfalls and confirm the findings reported in [17].

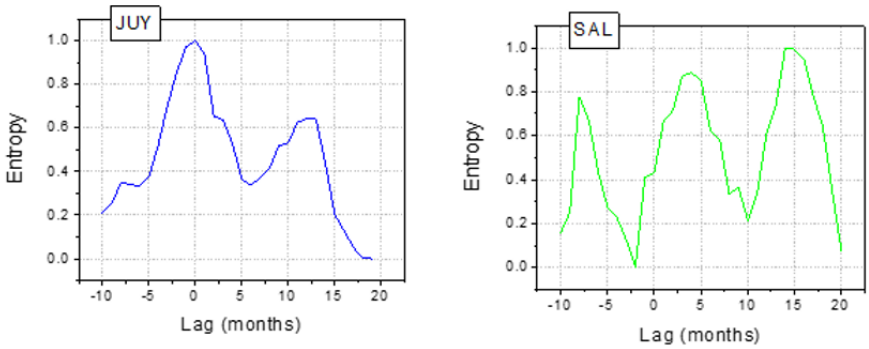


Fig. 3. The evolution of the coupling measure Q for the networks constructed using SOI index and the precipitation in JUY (left) and SAL (right)

Another interesting causal influence of ENSO is related to the Indian Ocean Dipole (IOD). ENSO has been linked to the sea surface temperature (SST) anomalies over other ocean basins via the "atmospheric bridge" [21]. IOD, which is also an air-sea coupled mode, determines SST periodic oscillation [22-23] and has influences on the climate of Australia and of the countries surrounding the Indian Ocean Basin. The general view assumes that IOD has a self-generating mechanism determined by the internal atmosphere–ocean coupling [24]. However, significant evidences on

the influence of ENSO on IOD have been already accumulated together with indications about the reverse effect [25]. It exists a mounting interest in this problem as the ENSO–IOD interlink increased since 1970 together with the enhancement of the Walker circulation [26].

The existence of the linkage between ENSO and IOD is studied in this paper by analyzing the causal influence between the Nino-4 index, which captures SST anomalies in the central equatorial Pacific, and the Indian Ocean SST and also between the IOD index and the Pacific Ocean SST.

The IOD intensity is represented by mean of the Dipole Mode Index (DMI), which is the SST gradient between the western equatorial Indian Ocean (50E-70E and 10S-10N) and the south eastern equatorial Indian Ocean (90E-110E and 10S-0N) [27]. The Nino-4 monthly index and the SST gridded data has been retrieved from [19], while monthly DMI series has been retrieved from [27]. The time period studied is in between the years 1958-2010.

The maps of the variation of the coupling measure Q, have been calculated for the pairs [DMI index, the tropical Pacific SST] and [Nino4, Indian Ocean SST], respcetively.

The causal relation between the IOD index and the tropical Pacific SST shows a clear influence by an El Nino-like pattern. More interesting are the maps showing the causal link between Nino4 and the Indian Ocean SST (Fig. 4). The influence of the the Indian Ocean on El Nino by mean of IOD is clearly revealed by the two positive poles in Fig. 4 (right).

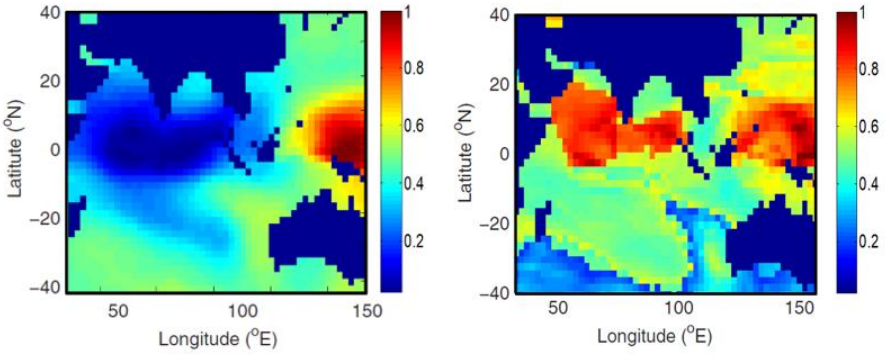


Fig. 4. Map of the synchronization measure Q revealing the causation between Nino4 and the Indian Ocean SST (left) and vice-versa (right).

3.2 The influence of climatic factors on malaria epidemic

The analysis of the causal relation between climatologic phenomena and the outbreak of various disease by means of time series analysis represents a relatively new topic. An effort has been spent on the study of the ENSO-driven climate variability connection with major outbreaks of leptospirosis, dengue, pandemic events. For a review of the present understanding of ENSO health associations the reader is referred to [28]. The coupling between climatic processes and malaria epidemic is also a major topic of interest, as this disease has huge social, economic, and health impact. Malaria incidence is significant in the tropical regions but the global warming could shift its area of influence towards more densely populated regions. The linkage between malaria epidemic and climatic factors has been explored mainly by means of process-based mathematical models and also with various methods inferring information directly from time series.

In this paper, the synchronization measure based on complex networks has been applied to the study of the coupling between the number of malaria cases and the health status of the vegetation, measured by the normalized difference vegetation index (NDVI) in the Rangamati district, Bangladesh. NDVI quantifies the vegetation fluctuations by measuring the difference between near-infrared radiation and visible light. Healthy and dense vegetation strongly absorbs the visible light received from the sun while it strongly reflects the near-infrared light [29]. This case has been previously studied by Haque et al. [30]. An association between the two processes has been retrieved with a time lag lower than 3 months by using a generalized linear negative binomial regression model. In this paper we are using the data reported in [30]. The number of malaria cases are related to the Rangamati district hospital (Bangladeshi Highlands, 22^o 40' N, 92^o 11' E) and it has been collected from January 1989 to December 2008. The NDVI index has been prepared based on the data library of the International Research Institute (IRI) of Lamont Doherty Earth Observatory (LDEO) at Columbia University, USA [30]. As in both cases the data is not publicly available, for the purpose of this paper, the data has been digitized form Fig. 2 in [30]. The time series have been resampled in a 4800 data points. The time series are presented in Fig. 5. The coupling has been studied for a time lag of maximum 6 months.

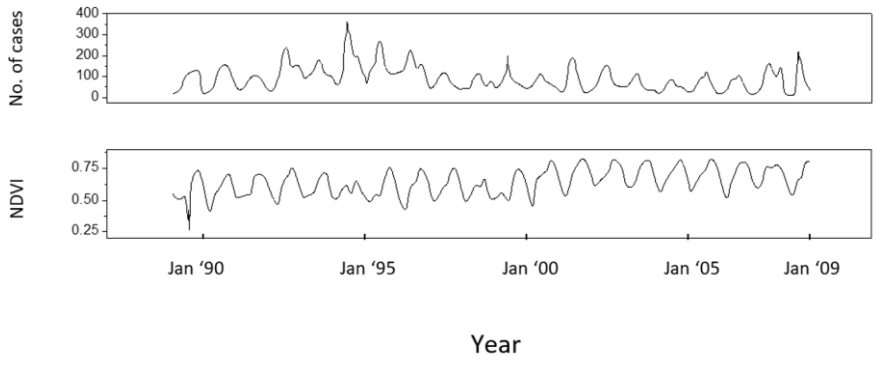


Fig. 5. Time series corresponding to the number of malaria cases recorded at the Rangamati district hospital (top) and to the evolution of NDVI (bottom).

The evolution of the coupling measure is presented in Fig. 6. The evolution of Q reaches a maximum at 2.3 months, indicating a coupling between the two processes. The results confirm Haques's et al. findings [30]. The figure shows also the structure of the complex network, for the time lag corresponding to the maximum coupling and to another time lag respectively. The networks have been created using the prefuse force directed lay-out in Cytoscape 3.7.1 [31]. For the maximum coupling the network complexity clearly increases, developing several clusters.

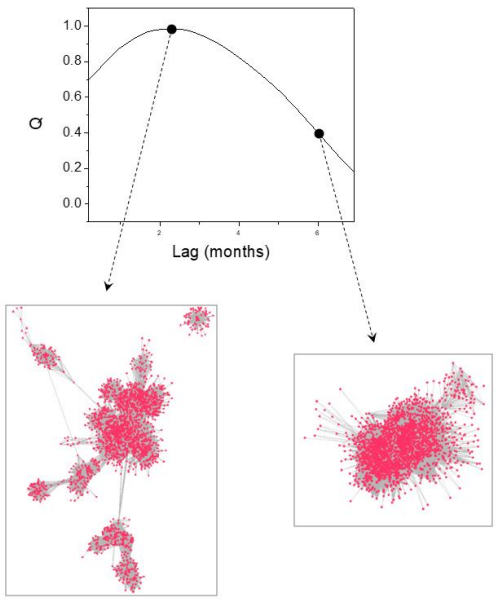


Fig. 6. The evolution of the coupling measure Q for the networks constructed using the number of malaria cases and the NDVI time series. The two inserts show the structure of the complex network at the time lag corresponding to the maximum coupling (left) and to another time lag, respectively (right).

4 Conclusion

A new method for the evaluation of coupling strength between dynamical systems, based on the entropy quantification of the topology of a cross-visibility network, has been applied to several climatic and epidemic coupled phenomena. The results show the potential of the approach to handle the investigation of real life complex systems.

References

1. T.M. Cover, J.A.Thomas, "Elements of Information Theory", Wiley, New York 1991, ISBN: 978-0-471-24195-9.
2. T. Schreiber, "Measuring Information Transfer", *Phys Rev Lett*, vol. 85, pp. 461–464, 2000.
3. K. Hlavácková-Schindlera, M. Palub, M.Vejmelkab, J. Bhattacharya, "Causality detection based on information-theoretic approaches in time series analysis", *Phys. Rep.*, vol. 441, pp. 1-46, 2007.
4. C.J. Stama, B.W. van Dijk, "Synchronization likelihood: an unbiased measure of generalized synchronization in multivariate data sets", *Physica D,* vol. 163, pp. 236–251, 2002.
5. A. Shabunin, V. Demidov, V. Astakhov, V. Anishchenko, "Information theoretic approach to quantify complete and phase synchronization of chaos", *Phys. Rev. E*, vol. 65, art. no. 056215, 2002.
6. M. Palus, A. Stefanovska, "Direction of coupling from phases of interacting oscillators: An information-theoretic approach.", *Phys. Rev. E*, vol. 67, art. no. 055201 (R), 2003.
7. G. Sugihara, R. May, H. Ye, C.-H. Hsieh, E. Deyle, M. Fogarty, S. Munch, "Detecting Causality in Complex Ecosystems", *Science*, vol. 338, no. 6106, pp. 496-500, 2012.
8. X. San Liang, "Unraveling the cause-effect relation between time series", *Phys. Rev. E*, vol. 90, art. no. 052150, 2014.
9. T. Kreuz, F. Mormann, R.G. Andrzejak, A. Kraskov, K. Lehnertz, P. Grassberger, "Measuring synchronization in coupled model systems: A comparison of different approaches", *Physica D*, vol. 225 pp. 29–42, 2007.
10. T. Craciunescu, A. Murari, M. Gelfusa, "Improving entropy estimates of complex network topology for the characterization of coupling in dynamical systems", submitted to *Entropy*.
11. L. Lacasa, B. Luque, F. Ballesteros, J. Luque, J.C. Nun, "From time series to complex networks: The visibility graph", *PNAS*, vol. 105, no.13, pp. 4972-4975, 2008.
12. S. Mehraban, A.H. Shirazi, M. Zamani, G.R. Jafari, "Coupling between time series: A network view", EPL vol. 103, art. no. 50011, 2016.
13. T. Craciunescu, A. Murari, "Geodesic distance on Gaussian Manifolds for the robust identification of chaotic systems", Nonlinear Dyn., vol. 86, pp. 677-693, 2016.
14. J.-L. Starck, F. Murtagh, P. Querre, F. Bonnarel, "Entropy and astronomical data analysis: Perspectives from multiresolution analysis", A&A, vol. 368, pp. 730–746, 2001.
15. S. Arjasakusuma, Y. Yamaguchi, Y. Hirano, X. Zhou, "X. ENSO- and Rainfall-Sensitive Vegetation Regions in Indonesia as Identified from Multi-Sensor Remote Sensing Data", ISPRS Int. J. Geo-Inf. Vol. 7, no. 103, 2018..

16. A. Attanasio, A. Pasini, U. Triacca, “Has natural variability a lagged influence on global temperature? A multi-horizon Granger causality analysis”, Dynamics and Statistics of the Climate System vol. 1, no. 11, 2016.
17. N. Marwan, M.C. Romano, M. Thiel, J. Kurths, “Recurrence plots for the analysis of complex systems”, Physics Reports vol. 438, pp. 237–329, 2007.
18. C. Papagiannopoulou, D.G. Miralles, S. Decubber, M. Demuzere, N.E.C. Verhoest, W.A. Dorigo, W. Waegeman, “A non-linear Granger-causality framework to investigate climate–vegetation dynamics”, Geosci. Model Dev., vol. 10, pp. 1945–1960, 2017.
19. NOAA ESRL Physical Sciences Division Data. Available online: https://www.esrl.noaa.gov/psd/data/gridded/rsshelp.html (accessed on 23 10 2018).
20. KNMI Climate Explorer, http://climexp.knmi.nl/ (accessed on 23 10 2018).
21. P.J. Webster, A.M. Moore, J.P. Loschnigg, R.R. Leben, “Coupled ocean-atmosphere dynamics in the Indian Ocean during 1997-98”, *Nature* vol. 23, no. 401, pp. 356-60, 1999.
22. N.H. Saji, B.N. Goswami, P.N. Vinayachandran, T. Yamagata, “A dipole mode in the tropical Indian Ocean.”, *Nature* vol. 401, pp.360–363, 1999.
23. A.P. Fischer, P. Terray, E. Guilyardi, S. Gualdi, P. Delecluse, “Two independent triggers for the Indian Ocean dipole zonal mode in a coupled GCM”, J. Climate, vol. 18, pp. 3428–3449, 2005.
24. H. Annamalai, S.P. Xie, J.P. McCreary, R. Murtugudde, “Impact of Indian Ocean sea surface temperature on developing El Niño”, J. Climate vol. 18, pp. 302–319, 2005.
25. L. Fan, Q. Liu, C. Wnag, F. Guo, “Indian Ocean Dipole Modes Associated with Different Types of ENSO Development”, *J. Climate*, vol. 30, pp. 2233-2249, 2017.
26. Y. Yuan, C. Li, “Decadal variability of the IOD-ENSO relationship”, Chin. Sci. Bull., vol. 53, pp. 1745–1752, 2008.
27. Japan Agency for Marine-Earth Science and Technology (JAMSTEC), http://www.jamstec.go.jp/e/
28. G.R. McGregor, K.Ebi, “El Niño Southern Oscillation (ENSO) and Health: An Overview for Climate and Health Researchers”, *Atmosphere*, vol. 9, art. no. 282, 2018.
29. J. Weier, D. Herring, Measuring Vegetation (NDVI & EVI). NASA Earth Observatory, Washington DC, 2000.
30. U. Haque, M. Hashizume, G.E. Glass, A.M. Dewan, H.J. Overgaard5, T. Yamamoto, The Role of Climate Variability in the Spread of Malaria in Bangladeshi Highlands, PLoS ONE, 5(12), e14341, 2010.
31. Cytoscape 3.4.0. Cytoscape Developers, NRNB, 2016.

Short-term Temperature Forecasts using Deep Learning – an Application to Data from Ulm, Germany

David Kreuzer, Michael Munz, Samuel Peifer and Stephan Schlüter
Technische Hochschule Ulm

Abstract

With the increasing importance of solar energy we need more precise forecasts of its amount in order to guarantee network stability. Forecasting models are normally based on solar radiation, the major impact factor, but also on temperature which impacts the level of efficiency of the solar modules. Since conventional weather models (like the Lorenz model) are extremely chaotic, data driven models are getting more popular. Among those models the field of deep learning, i.e. neural networks with multiple hidden layers, is becoming more relevant. Especially convolutional neural networks which do not have the need of a feature extraction stage are a solid alternative to classic approaches. One of the main reasons is that computation power for massive parallel computing (i.e. GPGPU computing) is increasing. Authors like Dong et al. (2018) or Xiaoyun et al. (2016) use a recurrent neural network to predict temperature, wind speed or radiation, for example. Especially long short-term memory (LSTM) networks are often used, due to the fact that recurrent networks generally show good results when dealing with time series data (Lopez et al., 2016) and LSTM cells, in particular, have the "ability to bridge very long time lags" (Hochreiter & Schmidhuber, 1997), which is crucial when dealing with seasonal data. To the best of the authors knowledge, nobody applied the combination of convolutional and LSTM layers in this context so far, which is what we do in this work in order to forecast temperature data. Therefore we use weather data from Ulm between 2015 and 2018.

The motivation for applying a convolution LSTM network is the assumption that individual measurements are highly correlated. Hence, the use of 2D-convolutional operations is likely to lead to more stable results, since interdependencies between the channels are taken into account. The recurrent connections of the LSTM, again, incorporate the time dependencies. As input data we use wind speed, wind direction, temperature, humidity, dew point temperature, air pressure, global radiation, and diffuse radiation, whereby we have a temporal granularity of 10 minutes. To ensure a better generalization of the network, the data is split day by day and shuffled, meaning that the order of the days, being fed to the network, varies in every epoch. To compensate the risk of losing the annual seasonality and to simplify training, domain knowledge is used. The Integration of prior knowledge is done by incorporating additional information to the networks input representation, specifically month, daytime, and monthly mean temperature according to every data point. We are using L2-regularisation in all layers and dropout only in the dense layers. For training the sliding window approach is used. The model is created with Python 3.6 and Tensorflow.

To benchmark our deep neural network, the results are compared with classical methods of time series forecasting such as naive forecasts or the seasonal autoregressive integrated moving average (SARIMA) model.
The SARIMA is chosen as we expect the temperature to show distinct daily seasonality. The model is an autoregressive model which accounts for dynamics in the error term but also for trends and seasonal effects, which are captured by a lag term. The degree of integration is determined using the partial autocorrelation function; all further parameters are identified by (conditional) maximum likelihood estimation (see e.g. McNeil et al., 2016).
For the case study we choose three different time horizons, namely 6, 12 and 24 hours. As performance measures we use the root mean squared error (RMSE) and the standard deviation of the mean absolute error (MAE) for the first up to the 144th time step (i.e. 24 hours). Both models show good results for consecutive days with similar patterns. Since the neural network also uses information from other channels, it shows better results than the SARIMA model when the weather changes drastically in comparison to the day before. Nevertheless feeding more information is not always beneficial – we see that the SARIMA model often outperforms the neural network in the first few hours. However, for larger forecasting horizons, the neural network delivers more accurate results. Regarding the prediction time, neural network prediction is independent from the newest data and has to be trained only once. After training, which takes much longer than the training of the SARIMA, it only requires data from the last six hours for prediction, while SARIMA always has to be fit to the data of the last week.

Literature

Hochreiter S, & Schmidhuber J (1997). Long Short-term Memory. Neural Computation; 9, pp. 1735-1780.
Dong D, Sheng Z, Yang T (2018). Wind Power Prediction Based on Recurrent Neural Network with Long Short-Term Memory Units. 2018 International Conference on Renewable Energy and Power Engineering (REPE), Toronto, ON, Canada, pp. 34-38.
Lopez L, Valle C, Allende H (2016). Recurrent Networks for Wind Speed Forecasting. International Conference on Pattern Recognition Systems (ICPRS-16), Talca, pp. 1-6.
McNeil A J, Frey R, Embrechts P. Quantitative Risk Management: Concepts, Techniques, and Tools. Princeton University Press: 2006.
Xiaoyun Q, Xiaoning K, Chao Z, Shuai J, Xiuda M (2016). Short-term prediction of wind power based on deep Long Short-Term Memory. 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), Xi'an, pp. 1148-1152.

Multiple change-point estimation of multi-path panel time series data via EM algorithm *

Jaehwi Kim[1,2][0000−0001−6303−1531] and Jaehee Kim*[1,3][0000−0002−2115−0142]

[1] Department of Statistics, Duksung Womens University, 33, Samyang-ro 144-gil, Dobong-gu, Seoul, S. Korea
[2] un7743@naver.com
[3] jaehee@duksung.ac.kr

Abstract. This paper addresses the problem of estimating multiple common change-points in multi-path panel data with the unknown number of change-points. In many applications, we have panel data which consist of many related univariate time-series. We present a novel approach using EM (Expectation Maximization) to detect change-points in such panel data to pool the information across time-series separated by the change-point. The variable dimension of parameters due to the unknown number of parameters causes mathematical complexity and computational difficulty in searching the change-points and resulting appropriate segments. We suggest the tail-cutting algorithm which is recursive and repetitive as an alternative for binary search algorithm. Simulations to evaluate the performance of the estimators are provided for panel time series data. We also demonstrate the usefulness of this method on the electricity usage data and PM10 data.

Keywords: EM algorithm · Mixture distribution · Multi path change point model · Panel time series data · Tail-cutting algorithm ·

1 Introduction

The term panel data is used for any data set with repeated observations over time for the same individuals. Panel data is better suited than cross-sectional data for studying the dynamics of change when it is well suited to understanding transition behavior. Panel data enables the study of more complex behavioural models for example the effects of technological change, or economic cycles We are interested in structural change-point detection for panel time series data. Panel data change-point detection research is not yet much done since it is both mathematically and computationally challenging to uncover hidden change-points in the observed sequences. For an overview of some of the methods used on univariate time series see Jandhyala et al. (2013)[1]. Detecting multiple change points in univariate time series has been widely discussed in various contexts

* *Corresponding Author. This research was supported by the Korea Research Foundation (KNRF) (No. 2018R1A2B26001664). Also it was supported by Korea Electric Power Corporation (Grant number: R18XA01).

including Inclan and Tiao (1994)[2], Chen and Gupta (1997)[3], Lavielle and Moulines (2000)[4], Ombao et al. (2001)[5], and Davis et al. (2006, 2008)[6], [7].

Many applications of change-point arise when repeated observations are made in time, on different patients for example. Joseph and Wolfson (1992, 1993)[8, 9] provided change-point inference when the data consist of several sample paths. In the context of a single sample path it is well known that the maximum likelihood estimator of the change-point is not consistent as the number of observations on either side of the change-point tends to infinity. By regarding each path as arising from a mixture of distributions, apart from the mathematical and statistical difficulties, there are also well known pitfalls in the actual solution of the likelihood equations. The EM algorithm can offer a tractable solution in the panel data change-point estimation.

There are some recent work with multivariate time series or panel data. Kirch et al. (2015)[10] described how to find multiple change-points in EEG data. Preuss et al. (2015)[11] proposed a nonparametric procedure using periodograms for multiple structural breaks in the autocovariance function based on MOSUM type statistics. Cho and Fryzlewicz (2015) [12] developed a method in segmenting the second-order structure of a high dimensional multivariate time series and showed an application to finance data. Cho (2016) [13] proposed multiple change-points detection based on double cusum statistics for panel data. Cao and Wu (2015)[14] considered large scale multiple testing procedure including p-values for multiple change-point estimation, and applied to a genome data. Vert and Bleakley (2010)[15] presented a fast algorithm for multidimensional multiple change-points utilizing LASSO. Qian et al. (2019) [16] proposed multiple change-points detection via integrating empirical Bayesian information and Gibbs sampling to find the optimal change-point configuration. Bardwell et al. (2019) [17] presented an approach to detect the most recent change-points in panel time series data using profile likelihood and dynamic programming. Such methods are applicable to detect changes in many areas such as finance, bioinformatics, and signal processing.

In this paper, we are interested in multiple change-points estimation in panel time series data. We propose the multiple change-point detection based on EM algorithm (MDEM) and the tail-cutting search to find the optimal set of change-points. The EM algorithm can define the location of unknown change-points as latent variables in the mixture distribution. EM has been widely studied in Dempter et al. (1977)[18] . Our contribution is utilizing EM computation and tail-cutting search for multiple change-points estimation in panel time series data when any parameter can change and getting convergence to a global maximization.

The plan of the paper is as follows. In Section 2, we define multi-path panel time series data with the unknown number of common parameter change-points and provide the expressions for the EM algorithm. Section 3 describes the tail-cutting algorithm as our proposal. Section 4 provides the result of simulation and real data analysis. Finally, we conclude in Section 5.

2 Multi-path change-point model for panel time series data

2.1 Structure of panel data

We consider the observations in the form of $M \times N$ array whose row consists of time series.

$$\mathbf{V} = \begin{pmatrix} Y_{11} & Y_{12} & \cdots & Y_{1N} \\ Y_{21} & Y_{22} & \cdots & Y_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ Y_{M1} & Y_{M2} & \cdots & Y_{MN} \end{pmatrix} \tag{1}$$

A change will be said to occur at common τ_k, $k = 1, \ldots, K$, for $i = 1, 2, \ldots, M$. We make the following assumptions:

(i) The observations $\{Y_{it}\}$ within row i ($i = 1, 2, \ldots, M$), follow different AR(p) model separated by the change-point.
(ii) The rows are independent mutually.
(iii) The sequence $\{\tau_k\}$, $k = 1, 2, \ldots, K$ is a set of independent and identically distributed random variables within the range of time point integers $\{p+1, \ldots, N\}$ and have a distribution function $G(\cdot)$ corresponding to the probability function $P_T(\tau) = P(T = \tau)$.
(iv) The distributional parameters such as mean $\mu_1, \ldots, \mu_K, \mu_{K+1}$; variance $\sigma_1, \ldots, \sigma_K, \sigma_{K+1}$; AR coefficients $\phi_1, \ldots, \phi_K, \phi_{K+1}$ are unknown.

For simplicity, we consider firstly the panel time series AR(1) model has one common change-point, for $i = 1, \ldots, M$,

$$y_{it} = \begin{cases} \mu_1 + \phi_1 y_{i,t-1} + \epsilon_{it},\ \epsilon_{it} \sim N(0, \sigma_1^2) & \text{for } t = 2, \ldots, \tau - 1, \\ \mu_2 + \phi_2 y_{i,t-1} + \epsilon_{it},\ \epsilon_{it} \sim N(0, \sigma_2^2) & \text{for } t = \tau + 1, \ldots, N. \end{cases} \tag{2}$$

For stationarity, each AR parameter must lie in the region S where

$$S = \{\phi_1, \phi_2 \in R : -1 \leq \phi_1, \phi_2 \leq 1\}. \tag{3}$$

At least a pre-specified fraction ξ_0 of the observations, for $l = 1, \ldots, K+1$, $C_l = \left\{ y_{[\xi_0(l-1)]} \leq y \leq y_{[\xi_0(l)]} \right\}$ where $[\cdot]$ denotes the integer part. For example, a safe choice for ξ_0 is 0.10.

2.2 EM algorithm for mixture distribution

The joint likelihood of the data $\{Y_{ij}\}$ in (1) is given by

$$L(\mathbf{y}|\Omega) = \prod_{i=1}^{M} \sum_{\tau_i=3}^{N-1} \left\{ \prod_{t=2}^{\tau_i - 1} f(y_{it}|\mu_1, \phi_1, \sigma_1^2) \prod_{t=\tau_i+1}^{N} f(y_{it}|\mu_2, \phi_2, \sigma_2^2) \right\} \times \theta_{\tau_i} \tag{4}$$

where $\Omega = \{\mu_1, \mu_2, \phi_1, \phi_2, \sigma_1^2, \sigma_2^2, \theta\}$ is the parameter set, $f(\cdot|\mu, \phi, \sigma^2)$ denotes the probability function for y_{it}, and $\theta_{\tau_i} = P(\hat{\tau}_i = \tau)$. Equivalently, the likelihood can be written with independent error terms $\{\epsilon_{it}\}$ such as

$$L(\epsilon|\Omega) = \prod_{i=1}^{M} \sum_{\tau_i=3}^{N-1} \left\{ \prod_{t=2}^{\tau_i-1} f(\epsilon_{it}|\mu_1, \phi_1, \sigma_1^2) \prod_{t=\tau_i+1}^{N} f(\epsilon_{it}|\mu_2, \phi_2, \sigma_2^2) \right\} \times \theta_{\tau_i} \tag{5}$$

where $\epsilon_{it} = y_{it} - \mu_k - \phi_k y_{i,t-1}$, for $k = 1$ or $k = 2$.

The unknown true change-points $\{\tau_k\}$ are regarded as missing data and the ensuing maximization is carried out via the EM algorithm. The quantities $P_T(\cdot)$ are dealt as mixing constants in a standard finite mixture problem when the mixture distribution have unknown parameters.

The maximum likelihood estimation is done by maximization of $L(\mathbf{y}|\cdot)$ with respect to $mu_1, \mu_2, \phi_1, \phi_2, \sigma_1, \sigma_2$ and θ. Here the time-points (i, t), $i = 1, 2, \ldots, M$, $t = 3, 4, \ldots, N-1$ have indicator function $I(t = \tau_i)$, but their results are latent variable. We denote latent variable by C, and solve the missing data problem through EM algorithm. The equation for C is as follows:

$$\theta_{\tau_i} = P\left[C_{i\tau_i} = 1\right], \qquad C_{i\tau_i} = \begin{cases} 1, & \tau_i : \text{change} - \text{point} \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Let Ω's dimension be variable according to the number of change-points. The complete data log-likelihood for observed variable ϵ and latent variable C is

$$l_c(\Omega) = \log L_c\left(\Omega; \mathbf{y}, \mathbf{C}\right) = \sum_{i=1}^{M} \sum_{\tau_i=3}^{N-1} c_{i\tau_i} \left\{\log \theta_{\tau_i} + \log g_{\tau_i}(y_{it})\right\} \tag{7}$$

where

$$g_{\tau_i}\left(y_{it}\right) = \prod_{t=2}^{\tau_i-1} f(y_{it}|\mu_1, \phi_1, \sigma_1^2) \prod_{t=\tau_i+1}^{N} f(y_{it}|\mu_2, \phi_2, \sigma_2^2). \tag{8}$$

The EM algorithm of Dempster et al. (1977) [9] can be applied as follows:
E-step: Given the current estimate at the kth step, $\hat{\mu}_1^{(k)}, \hat{\mu}_2^{(k)}, \hat{\phi}_1^{(k)}, \hat{\phi}_2^{(k)}, \hat{\sigma}_1^{2(k)}, \hat{\sigma}_2^{2(k)}$ and $\hat{\theta}^{(k)}$, target function Q is

$$Q\left(\Omega; \hat{\Omega^{(k)}}\right) = E_{C|Y,\hat{\Omega}^{(k)}}[l(\Omega)] \to E\left[C|Y, \hat{\Omega^{(k)}}\right] = P\left[C = 1|Y, \hat{\Omega}^{(k)}\right]. \tag{9}$$

Compute the probability for change-point at τ_k as

$$\begin{aligned} c_{i\tau_i}^{(k+1)} &= P\left[C = 1|Y, \hat{\Omega}^{(k)}\right] \\ &= \frac{\frac{\hat{\theta}_{\tau_i}^{(k)}}{(\hat{\sigma}_1^{2(k)})^{\frac{\tau_i-1}{2}} (\hat{\sigma}_2^{2(k)})^{\frac{N-\tau_i+1}{2}}} h^*\left(y_{it}|\hat{\mu}^{(k)}, \hat{\phi}^{(k)}, \hat{\sigma}^{2(k)}\right)}{\sum_{\tau_i=3}^{N-1} \frac{\hat{\theta}_{\tau_i}^{(k)}}{(\hat{\sigma}_1^{2(k)})^{\frac{\tau_i-1}{2}} (\hat{\sigma}_2^{2(k)})^{\frac{N-\tau_i+1}{2}}} h^*\left(y_{it}|\hat{\mu}^{(k)}, \hat{\phi}^{(k)}, \hat{\sigma}^{2(k)}\right)} \end{aligned} \tag{10}$$

where

$$h^*\left(y_{it}|\hat{\mu}^{(k)},\hat{\phi}^{(k)},\hat{\sigma}^{2(k)}\right)=\exp\left\{-\sum_{j=2}^{\tau_i-1}\frac{\left(y_{it}-\hat{\mu}_1^{(k)}-\hat{\phi}_1^{(k)}y_{i,t-1}\right)^2}{2\hat{\sigma}_1^{2(k)}}-\sum_{j=\tau_i+1}^{N}\frac{\left(y_{it}-\hat{\mu}_2^{(k)}-\hat{\phi}_2^{(k)}y_{i,t-1}\right)^2}{2\hat{\sigma}_2^{2(k)}}\right\}.$$

M-step: We compute parameter estimates to optimize $Q\left(\Omega;\hat{\Omega}^{(k)}\right)$.

$$\hat{\theta}_{\tau_i}^{(k+1)}=\frac{1}{M}\sum_{i=1}^{M}c_{i\tau_i}^{(k)}, \tag{11}$$

$$\hat{\mu}_1^{(k+1)}=\frac{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=2}^{\tau_i-1}c_{i\tau_i}^{(k)}\left(y_{it}-\hat{\phi}_1^{(k)}y_{i,t-1}\right)}{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}c_{i\tau_i}^{(k)}(\tau_i-2)},$$
$$\hat{\mu}_2^{(k+1)}=\frac{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=\tau_i+1}^{N}c_{i\tau_i}^{(k)}\left(y_{it}-\hat{\phi}_2^{(k)}y_{i,t-1}\right)}{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}c_{i\tau_i}^{(k)}(N-\tau_i)},$$

$$\hat{\phi}_1^{(k+1)}=\frac{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=2}^{\tau_i-1}c_{i\tau_i}^{(k)}(y_{it}-\hat{\mu}_1^{(k+1)})(y_{i,j-1}-\mu_1^{(k+1)})}{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=2}^{\tau_i-1}c_{i\tau_i}^{(k)}y_{i,t-1}^2},$$
$$\hat{\phi}_2^{(k+1)}=\frac{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=\tau_i+1}^{N}c_{i\tau_i}^{(k)}(y_{it}-\hat{\mu}_2^{(k+1)})(y_{i,t-1}-\mu_2^{(k+1)})}{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}c_{i\tau_i}^{(k)}y_{i,t-1}^2},$$

$$\hat{\sigma}_1^{2(k+1)}=\frac{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=2}^{\tau_i-1}c_{i\tau_i}^{(k)}\left(y_{it}-\hat{\mu}_1^{(k+1)}-\hat{\phi}_1^{(k)}y_{i,t-1}\right)^2}{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}c_{i\tau_i}^{(k)}(\tau_i-2)},$$
$$\hat{\sigma}_2^{2(k+1)}=\frac{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}\sum_{t=\tau_i+1}^{N}c_{i\tau_i}^{(k)}\left(y_{it}-\hat{\mu}_2^{(k+1)}-\hat{\phi}_2^{(k)}y_{i,t-1}\right)^2}{\sum_{i=1}^{M}\sum_{\tau_i=3}^{N-1}c_{i\tau_i}^{(k)}(N-\tau_i)}.$$

One alternates between the E-step and the M-step until a convergence criterion is met.

The distribution of $\hat{\phi}$ is explained in [19], $\hat{\phi}$'s are compared by Chi-squared test since

$$\hat{\phi}_1\sim N\left(\phi_1,\frac{1}{n_1}\left(1-\phi_1^2\right)\right),\qquad \hat{\phi}_2\sim N\left(\phi_2,\frac{1}{n_2}\left(1-\phi_2^2\right)\right). \tag{12}$$

Therefore we can use the test

$$T = \left(\frac{\hat{\phi}_1}{\sqrt{\left(1-\hat{\phi}_1^2\right)/n_1}} - \frac{\hat{\phi}_2}{\sqrt{\left(1-\hat{\phi}_2^2\right)/n_2}} \right)^2 \sim \chi^2(1) \tag{13}$$

where n_1 and n_2 are the number of observations in each subsegment respectively. If the test rejects the null hypothesis that the two subsegments have the same parameters, then the change-point is suggested.

3 Segmentation algorithm for multiple change-points search

We focus on the problem of multiple change-points in panel data with the unknown number of change-points.

3.1 multiple change-points model for mean, variance and AR(1) parameters change

For panel time series data we consider the common multiple change-points model.

$$y_{it} = \begin{cases} \mu_1 + \phi_1 y_{i,t-1} + \epsilon_{it} & \text{for} \quad \epsilon_{it} \sim N(0,\sigma_1^2), \quad t = 2,\ldots,\tau_1 \\ \mu_2 + \phi_2 y_{i,t-1} + \epsilon_{it} & \text{for} \quad \epsilon_{it} \sim N(0,\sigma_2^2), \quad t = \tau_1+1,\ldots,\tau_2 \\ \vdots & \vdots \\ \mu_k + \phi_k y_{i,t-1} + \epsilon_{it} & \text{for} \quad \epsilon_{it} \sim N(0,\sigma_k^2), \quad t = \tau_{k-1}+1,\ldots,\tau_k \\ \mu_{k+1} + \phi_{k+1} y_{i,t-1} + \epsilon_{it} & \text{for} \quad \epsilon_{it} \sim N(0,\sigma_{k+1}^2), \quad t = \tau_k+1,\ldots,N \end{cases} \tag{14}$$

where $i = 1,\ldots,M$, $k = 1,\ldots,K$.

3.2 Tail-cutting algorithm

The EM algorithm is fundamentally a method of estimating parameters, and the number of parameters to be estimated depends on the number of change-points. It has some limitations for change-point estimation due to its parameter estimation ability in the given segment. According to the variable parameter dimension with the unknown number of K change-points, we need the computational search strategy combined with the EM algorithm.

As an alternative for the binary search, we propose the tail-cutting algorithm in the followings:

(i) First, find one change-point from the whole segment. (This is a tentative point used to separate the segment.) Based on this point, the data is divided into two subsegments.

(ii) Then test whether the two subsegments have different parameters. If the parameters of the two subsegments are not significantly different, the partitioning operation ends without splitting the two subsegments.

(iii) Perform the same procedure as in (i), (ii) for each subsegment. We repeat it for the subsegments located at both ends of the divided sub segments until they are no longer divided.

(iv) After the process is stopped when there is no more possible to be divided into subsegment, the tentative points are determined as change-points. Then we disregard the first and last subsegments.

(v) We repeat steps (i) - (iv) with the segment with the end removed until no more change-points are detected.

This algorithm detects change-points from the both ends. Unlike the binary search, the change-point can be changed and re-decided to be optimized because searching procedures is done again. For the detecting or stopping rule, two-sample comparison test is performed for two subsegment divided by the estimated change-point. The Chi-squared test (13) is performed in each change-point decision to test whether the two segments are separable.

4 Empirical Study

4.1 Performance measure

For the performance of change-point estimators the following measures are used such asProp, Prop2, false discovery rate (FDR), d, mean of the number of detected change-points (MDC).

$$\text{Prop} = \frac{1}{K}\sum_{k=1}^{K} P\left[\hat{\tau}_k = \tau_k\right], \quad \text{Prop2} = \frac{1}{K}\sum_{k=1}^{K} P\left[\hat{\tau}_k \in (\tau_k - 5, \tau_k + 5)\right],$$

$$\text{FDR} = E\left[\frac{FD}{\hat{K}+1}\right],\ d = \max_{k}\min_{ke}\left|\hat{\tau}_{ke} - \tau_k\right|,\ \text{MDC} = \frac{1}{N}\sum_{i=1}^{N}\hat{K}_i$$

where K is the number of change-points, $\hat{K}$ is the number of estimated change-points, FD is the number of false discovered change-points. $\hat{\tau}_{ke}$ is the estimated change-point.

4.2 Simulation

We demonstrate the accuracy of our proposed method and compare with MRC (Bardwell, 2019) method. The data are generated from Model (14). Two cases are considered with one, two, three and five change-points. We set $N = 150$ time-points, $M = 50$ subjects in 100 repetitions for each simulation. Let $\eta = \tau/N$ be the relative change-point position in the data. Table 1 and Figure 1 provide the result of Case I and Table 2 and Figure 2 provide that for Case II. The results shows that the proposed method has high proportion of prop and prop2. Overall our method seems better than MRC method. Figures show that the probability distribution for the estimated change-points has peak on the true change-points.

Table 1. Case I. Change-point estimation results when only AR(1) parameter changes in panel data with $M = 50$, $N = 150$ in 100 repetitions

	Parameters							
η	0.6		0.2, 0.6		0.2, 0.5, 0.8		0.2, 0.36, 0.54, 0.7, 0.84	
μ	0							
ϕ	0.1, 0.8		0.1, 0.5, 0.3		0.1, 0.5, 0.35, 0.8		0.1, 0.5, 0.25, 0.8, 0.6, 0.3	
σ^2	1							
K	1		2		3		5	
Method	MDEM	MRC	MDEM	MRC	MDEM	MRC	MDEM	MRC
Prop (%)	90	0	27	0.5	37.33	0	23	0.4
Prop2 (%)	91	0	39	0.5	55.33	0.33	67.6	0.4
FDR	0.710	0.762	0.646	0.767	0.536	0.758	0.635	0.777
d	8.39	36.03	29	33.86	32.31	94.65	26.88	66.12
MDC	3	3.44	3.53	3.60	3.88	3.44	5.44	3.87

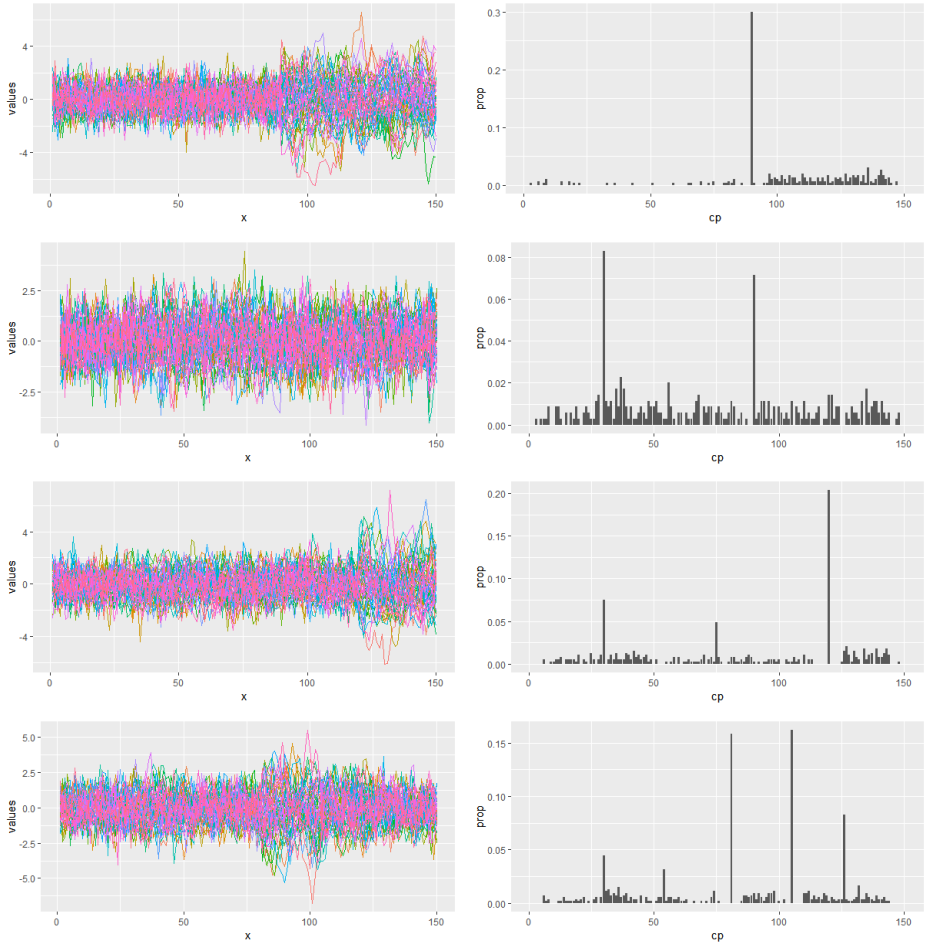


Fig. 1. Case I. Simulation data with change-points (left) and probability plot of change-point estimates (right)

Table 2. Case II. Change-point estimation results when mean, variance, and AR(1) parameter changes in panel data with $M = 50$, $N = 150$, in 100 repetitions

	Parameters							
η	0.6		0.2, 0.6		0.2, 0.5, 0.8		0.2, 0.36, 0.54, 0.7, 0.84	
μ	0, 2		0, 2, 1		0, 3, 1.5, 0		0, 3, 1.5, 0,2, 1	
ϕ	0.1, 0.5		0.1, 0.5, 0.35		0.1, 0.6, 0.25, 0.8		0.1, 0.6, 0.25, 0.8, 0.1, 0.5	
σ^2	2, 1		2, 1, 1.5		2, 1, 3, 0.5		2, 1, 3, 0.5, 1, 2.5	
K	1		2		3		5	
Method	MDEM	MRC	MDEM	MRC	MDEM	MRC	MDEM	MRC
Prop (%)	97	0	95.5	0	68.67	0	93.4	0
Prop2 (%)	100	0	99.5	27	75	27.67	99	0
FDR	0.279	0.722	0.343	0.746	0.296	0.774	0.209	0.767
d	9.45	17.35	0.25	80.75	29.12	91.11	1.07	111.58
MDC	2.03	3.63	3.74	3.35	4.08	3.68	6.28	3.61

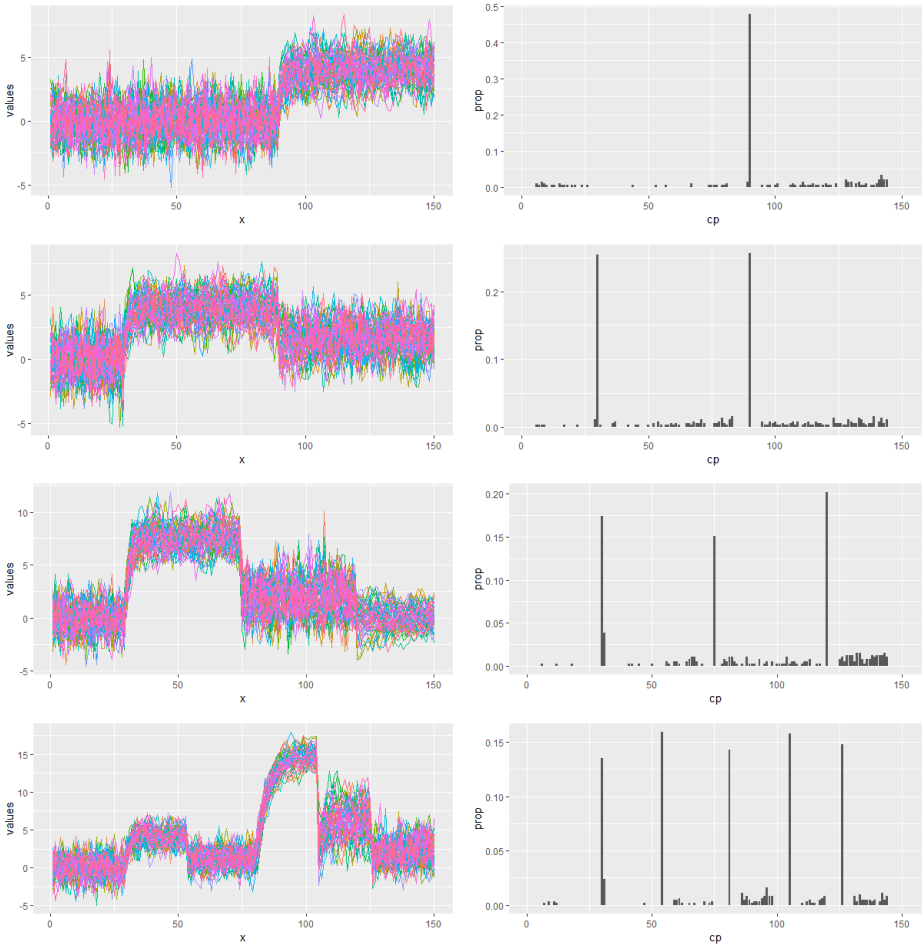


Fig. 2. Case II. a simulation data with change-points (left) and bar plot of change-point estimates (right)

4.3 Real panel data application

In this section, we apply our method to two real panel datasets. The aim of the analysis is to find some change-points and to explore the patterns. This information helps us to understand the structure and to control it. In the case of time series analysis, it enables forecasting of the future process, by treating the last (recent) estimated segment as being stationary.

The electricity consumption data from S factory. We use 15-minute interval electricity consumption data on June 18, 2018 from S factory located in Pohang, S. Korea. This panel data consists of four main units, auxiliary units, and coilers in the factory (6 devices, $M = 6$) with $N = 96$ time points. The data is log-transformed to reduce the scale and to stabilize the variance. Figure 3 shows ten estimated change-points of the electricity consumption panel data. The obtained information from change analysis gives us to understand the usage structure and to control or predict.

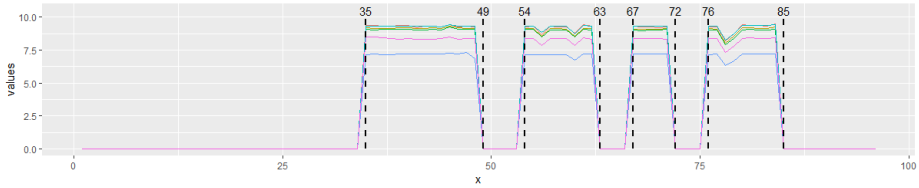


Fig. 3. S factory electricity consumption data and estimated change-points

PM10 data in S. Korea. We apply our method to PM10 data of Republic of Korea from January 2010 to September 2018 (105 time points). This panel data consists of PM10 amount that recorded each month of 78 cities and counties. The data is log-transformed and missing values are estimated using a spline method [20]. The PM10 levels in Korea seem the lowest in summer and high between autumn and spring periodically . Therefore we use a seasonal-trend decomposition procedure based on loess (STL) [21] to remove periodicity. Then the remainder is put into change-point estimation procedure. Since ARMA(1,1) can be fitted for the whole panel data, AR(1) model can be used for the PM10 panel data. Figure 4 shows the change-point estimation results of PM10 data with three change-points. The more refined analysis is possible with the subsegments divided by the estimated change-points. By time order, AR(1) coefficient ϕ value of each segment divided by the estimated change-point is as follows:

$$\phi_1 = 0.1314, \quad \phi_2 = 0.0283, \quad \phi_3 = 0.3967, \quad \phi_4 = 0.0160.$$

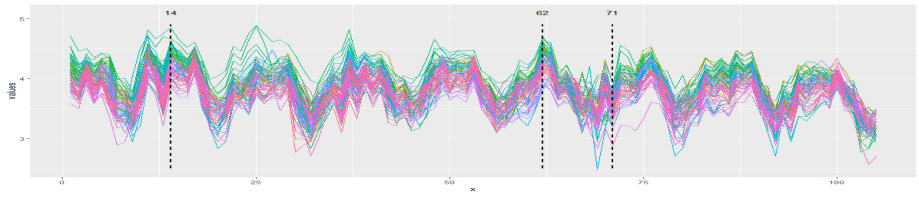


Fig. 4. PM10 data and estimated change-points

5 Conclusion

We have developed a novel method using EM algorithm to detect change-points in panel time series data. After finding the change-points, we are also able to identify the series affected by different changes, which leads to a greater understanding of occurrence of structural change. Difficulty in estimating the change-point distribution may arise when the size of change is small and there are few data paths. Our empirical results suggest that our method is robust to the panel data with short-term autocorrelation with the common change-point. Furthermore, our general approach can easily be extended to allow for modeling of panel data change-points with other dependent structure. Such an approach seems to be needed for time series with substantial or long-range dependencies. Our method ignores any dependence across time series, either in the form of cross-correlation. The former is an open and intriguing area of future research for the change-point community. The latter problem is not yet much discussed and therefore related research is expected.

References

1. Jandhyala, V., Fotopoulos, S. Macneill, I. and Liu P. (2013) Inference for single and multiple changepoints in time series, Journal of Time Series Analysis, 34, 423446
2. Inclan, C. and Tiao, G. C. (1994) Use of cumulative sums of squares for retrospective detection of changes of variance. J. Am. Statist. Ass., 89, 913923.
3. Chen, J. and Gupta, A. K. (1997) Testing and locating variance change-points with application to stock prices. J. Am. Statist. Ass., 92, 739747.
4. Lavielle, M. and Moulines, E. (2000) Least-squares estimation of an unknown number of shifts in a time series. J. Time Ser. Anal., 21, 3359.
5. Ombao, H. C., Raz, J. A., von Sachs, R. and Malow, B. A. (2001) Automatic statistical analysis of bivariate nonstationary time series. J. Am. Statist. Ass., 96, 543560.
6. Davis, R. A., Lee, T. C. M. and Rodriguez-Yam, G. A. (2006) Structural break estimation for non-stationary time series. J. Am. Statist. Ass., 101, 223239.
7. Davis, R. A., Lee, T. C. M. and Rodriguez-Yam, G. A. (2008) Break detection for a class of nonlinear time series models. J. Time Ser. Anal., 29, 834867.
8. Joseph, L. and Wolfson, D. B. (1992). Estimation in multi-path change-point problems, Comm. Statist. Theory and Methods, 21,897-913.

9. Joseph, L. and Wolfson, D. B. (1993) Maximum likelihood estimation in the multi-path change-point problem. Ann. Inst. Statist. Math. 45, 3, 511- 530
10. Kirch, C., Muhsal, B., and Ombao, H.: Detection of changes in multivariate time series with application to EEG data. Journal of the American Statistical Association. **110**(511), 1197–1216 (2015)
11. Preuss, P., Puchstein, R., and Dette, H.: Detection of multiple structural breaks in multivariate time series. Journal of the American Statistical Association. 110(510), 654–668 (2015)
12. Cho, H., and Fryzlewicz, P.: Multiplechangepoint detection for high dimensional time series via sparsified binary segmentation. Journal of the Royal Statistical Society: Series B (Statistical Methodology). **77**(2) 475–507 (2015)
13. Cho, H.: Change-point detection in panel data via double CUSUM statistic, Electronic Journal of Statistics **10**, 2000–2038 (2016)
14. Cao, H. and Wu, W. B. (2015) Changepoint estimation: another look at multiple testing problems, Biometrika, 1-7
15. Vert, J. P., and Bleakley, K.: Fast detection of multiple change-points shared by many signals using group LARS. In Advances in neural information processing systems. 2343–2351 (2010)
16. Qian, G., Wu. Y. and Xu M. (2019)Multiple change-points detection by empirical Bayesian information criteria and Gibbs sampling induced stochastic search, Applied Mathematical Modelling, 72, 202-216.
17. Bardwell, L., Fearnhead, P., Eckley, I. A., Smith, S., and Spott, M.: Most recent changepoint detection in Panel data. Technometrics. **61**(1), 88–98 (2019)
18. Dempster, A. P., Laird, N. M., Rubin, D. B.: Maximum likelihood from incomplete data via the EM algorithm, Journal of Royal Statistical Society Ser. **B**(39), 1–38 (1977)
19. Brockwell, P. J., and Davis, R. A.: Time Series: Theory and Methods. 2nd edn. Springer, New York (1990)
20. Moritz, S., Bartz-Beielstein, T.: imputeTS: time series missing value imputation in R. The R Journal. **9**(1), 207–218 (2017)
21. Cleveland, R. B., Cleveland, W. S., McRae, J. E., Terpenning, I.: STL: a seasonal-trend decomposition. Journal of official statistics. **6**(1), 3–73 (1990)

Time Series Generation using a One Dimensional Wasserstein GAN

Kaleb E. Smith[1] and Anthony O. Smith[1]

Florida Institute of Technology, Melbourne FL 32901, USA

Abstract. Time series data is an extremely versatile data type that can represent many real world events; however the acquisition of event specific time series requires special sensors, devices, and to record the events, and the man power to translate to one dimensional (1D) data. This is a costly labor effort and in many cases events are not frequent enough which results in a lack of time series data describing these events. This paper looks to address that issue of a shortage of event time series data by implementing a one dimensional Wasserstein Generative Adversarial Network utilizing a residual network architecture in the generator. With this framework, we learn a specific time series data distribution, which gives us the ability to autonomously generate synthetic data resembling actual real world events. We demonstrate this by covering a multitude of different time series data types (sensors, medical, imaging, etc). We justify our method visually by comparing the mean envelope of the real data versus the actual generated synthetic data, and we also look to justify it by a machine learning approach, having our synthetic data classified by a state-of-the-art time series classifier into its appropriate class. Our method shows to perform well on generating time series data that stays in between the mean envelope of the original data set and classifies within 4% of the original data set test sets.

Keywords: Generative Adversarial Networks · time series data · synthetic data generation · deep learning · Wasserstein GAN · .

1 Introduction

Time series data is one of the most important data types for understanding how the real world functions around us day to day. Time series can represent how complex events respond through periods in time, with some examples being weather, traffic patterns, stock markets, fiances, industry growth, sales forecast, movie selection and many more. With this type of data capturing such important information, it would be ideal to have algorithms that can perform machine learning tasks (classify, cluster, or forecasting) for event classification, so that we are better equip with knowledge based off of collected past events.

One such type of algorithm which has shown great success in machine learning tasks on time series data in the community today is deep learning algorithms. Deep learning algorithms run data through several stacked layers of neural network architecture (with complex activation functions and possibly convolutional

filters) each of which passes its representation of the data to the next layer. The ability to process large numbers of features makes deep learning very powerful when dealing with unstructured data. However, deep learning algorithms are prone to overfitting on less complex problems or lack of data because they require access to a large amount of data to be effective.

Another issue with these types of the deep learning methods is the lack of ability to collect the data needed for training. Sometimes this data can be sporadic and the events do not occur as often as liked for a meaningful amount of data to be used for the training, i.e hurricanes or stock market crashes. There is also the labor tasking of affording the man power to be able to log these events with the appropriate label while the time series data is happening. This is costly and most times not as efficient as needed and can add time to the application wished to be learned by the algorithm.

With deep learning algorithms growing more popular by the day its essential to be able to provide enough data to make sure they excel in the given task. When data is not available, it is suitable to look at an intelligent method to synthesize any amount of data that would be needed in order flood the algorithm with enough training data to be efficient.

This paper looks to address this problem and generate relevant time series data similar to the time series class of interest. Our method explores the use of deploying a generative adversarial network which demonstrates a promising ability to generate synthetic time series data in a broad range of data types from a benchmark time series data set. Our method shows visually our generated samples stay in the mean envelope of the original data set, and our generated samples classify within a 4% variance to the original data set test samples performed by a state-of-the-art (SOTA) classification algorithm.

2 Related Works

Generative Adversarial Networks (GANs) have become one of the most popular research topics since introduced in Goodfellow's original paper in 2014 collecting over eight thousand citations to date [1]. Most of these citations though revolve around the concept of two dimensional (2D) data or image generation to strengthen image related tasks (synthesizing video, helping in image recognition, scene classification, etc). However when it comes to 1D data GANs have very few studies done.

One of the first approaches on the use of GANs for time series generation was done by Morgren, utilizing both recurrent neural networks (RNN) and GANs to synthesize music data [2]. The author shows how a network and adversarial training can be used to train, and be highly flexible and expressive with continuous sequence data for tone length, frequencies, intensities, and timing; varying from the original approach of using RNNs through symbolic representations only.

Esteban et al. utilized the work done with RNNs and GANs by using a recurrent conditional generative adversarial network (RCGAN) to generate 1D data [3]. Their paper showed the concept of 1D generation was possible with

simple data like sinusoidal and real world medical data like oxygen saturation, heart rate monitoring, etc. To build off of this for synthesizing medical data, Hartmann et al. chose to focus electroencephalographic (EEG) data utilizing an improved Wasserstein GAN (WGAN) to generate their data [4]. These authors used a progressively growing implementation of the WGAN to achieve better generated results. Our work follows closest to Hartmann's, except we do not use a convolutional generator nor do we do progressive growing in our network.

An interesting network for time series generation comes from a newer paper by Brophy et al. whom mapped their time series data to 2D images and then applied a GAN to generate more 2D images and then map those synthetic images back to 1D to create 1D synthesized data [5]. This proved to be quick and provide comparable results in the time series domains applied that leveraged this approach. Lastly, there is a hybrid method utilizing both variational autoencoders (VAEs) and GANs to synthesize piano music [6].

An older method that did not catch on for time series generation was using RNNs to generate the time series like forecasting. Kulkarni et. al used RNNS to generate mobility traffic in their application, showing promising results by utilizing the hidden layers in the RNN to extract meaningful patterns that were later used to generate the new samples [7]. Alzantot et. al also used multiple RNNs and a Mixture Density Network to generate time series sensor data and then use another RNN to discriminate if it was real or fake. Though this seems like a GAN, the authors claim it is not based off of the way the networks were trained [8]. These methods rely heavily on the RNNs in their processes, however show poor results to those that have now been shown in GANs. This could be due to the complexity of their data they wish to emulate or the fact that not enough is learned during an RNN in training to be able to generate newer time series data.

To note, some of the most ground-breaking time series generation with GANs come from methods that use autoregressive models. These methods are state-of-the-art in music generation and have shown some promising abilities in their 1D generation. However, they are computationally expensive and show to be bounded to music data types and features only associated with that data type [9–11]. We do not explore autoregressive models in our methodology, but see their potential in future research.

3 Methodology

Our method utilizes a Wasserstein GAN with gradient penalty (WGAN-GP), using a residual network as the generator and a fully connected convolutional network for the discriminator [16, 14]. We look to leverage the recent success that deep networks have shown in mapping useful features from time series data in hopes to be able to generate meaningful 1D data.[12, 14].

WGAN was developed in 2017 with the hope to reduce a flaw in GANs called mode collapse. Mode collapse is when the generator generates a limited variety of samples, or even the same sample, regardless of the input, and never learns

the distributions from the data. This is partially due to the discriminator not forcing more variety in the generator, and if the generator gets collapsed to a single sample it is stuck there and will not learn a way out. The result is the same data sample being generated regardless of the random input.

WGAN fights this by using the Wasserstein distance, also called Earth Mover distance (EM distance) instead of Kullback-Leibler Divergence (KL) or Jensen-Shannon Divergence. These two divergences measure similarities between two probability distributions while the EM distance measures minimum energy cost of moving/merging one mass distribution to the shape of the other distribution. The EM distance loss for GANs can be seen in Eq 1.

$$W(P_r, P_\theta) = \sup_{||f||_L \leq 1} E_{x \sim P_r}[f(x)] - E_{x \sim P_\theta}[f(x)] \tag{1}$$

Where the supremum is over all the 1-Lipschitz functions $f : .\mathcal{X} \to \mathcal{R}$. Note that if we replace $||f||_L \leq 1$ for $||f||_L \leq K$ (consider K-Lipschitz for some constant K), then we end up with $K \cdot W(P_r, P_\theta)$. This means we have a parameterized family of functions that are all K-Lipschitz for some K, then the optimization problem becomes:

$$\max_{w \in \mathcal{W}} E_{x \sim P_r}[f_w(x)] - E_{z \sim p(z)}[f_w(g_\theta(z))] \tag{2}$$

where $E_{x \sim P_r}[f_w(x)] - E_{x \sim P_g}[f_w(x)]$ is the loss for the discriminator/critic and $-E_{z \sim p(z)}[f_w(g_\theta(z))]$ is the loss for the generator. The EM distance has two properties of value to deep networks; first the function is continuous anywhere and second the gradient of the function is almost everywhere.

In the original WGAN paper, they use weight clipping to restrict the maximum weight value in f to ensure that f is a 1-Lipschitz function. Though it demonstrated good results, it still suffered from poor convergence and the quality of images was not great. This is associated with the weight clipping not letting the model really learn fully and being constricted in its generalization. To combat this, gradient penalty (GP) was developed and shown to enhance convergence and generate better quality images [17]. The idea of WGAN-GP is to merge the limit to the loss function and set the Lagrange multiplier as a constant in WGAN which is the GP term, seen in Eq 3.

$$\min_G \max_D E_{x \sim P_r}[log(D(x))] + E_{\tilde{x} \sim P_g}[log(1 - D(\tilde{x}))] \tag{3}$$

Our method uses the sophistication of the WGAN-GP algorithm, but instead of the 2D realm of image generation we construct our WGAN-GP using residual networks (ResNet) rather than vanilla convolutional networks. Our decision is due to ResNet's ability to map better and deeper features from images and time series data [14, 15]. ResNets solved the issue of the vanishing gradient problem of deeper networks by introducing residual blocks, which essentially let the network bypass one or more layers in its training process. This allows for deeper networks to not experience vanishing gradients from the identity skip connections in the

residual blocks. Another benefit of ResNets is their ability to have deeper networks but fewer parameters to learn, which allows for faster convergence during training. We use ResNet in the generator portion of our WGAN-GP to learn the 1D distributions of our time series data and generate meaningful data.

4 Experiments

Our experiments were conducted on a benchmark time series data set from University of California Riverside (UCR) [13]. This data set contains over 128 different univariate time series data sets and 30 multivariate time series data sets. We focus our experiments on the univariate subset. The UCR data set contains different types of time series data (i.e. how the time series was generated) spanning electronic devices, motion, ECG, HAR, images, spectrograms, simulations, audio, and sensor data sources. In our experiments we look to go across these different types as well to see if the GAN does achieve better generated samples from specific data set types.

To train our GAN, we use both the training and testing data concatenated and train for 250 epochs for each data set. Our generator is a small ResNet based off of ResNet 18 architecture, with LeakyReLU activation functions with a 0.2 threshold, no batch normalization, and Adam as the optimizer. Our hardware was a Intel(R) Core(TM) i7-7820X CPU @ 3.60GHz, 128G of RAM, and a NVIDIA GV100 graphics processing unit. Our tests took anywhere from 5-15 minutes per class per data set depending on the number of samples for training.

We wanted to visually compare our generated time series data to the original data. To do this, we look to show the generated data plotted against the mean envelope of the original data and visually see how well it matches up. We found through our data sets chosen, our 1D-WGAN-GP does well when compared to the original data. There are some outliers in the data though where certain peaks are extended further than the max or min of the original data set, seen in both Tab. 1 and Tab. 2. It's promising in many of the data sets to see a tight fit around the mean envelope; Herring, coffee, and meat showing the best results. These three data sets are spectrograms and image data types, and GANs success in 2D images/spectrograms could be a factor in why these do so well when they are turned into 1D data and then generated through our 1D-WGAN-GP.

There are some cases however where the GAN produces synthetic data that visually just looks like random time series in that class. For example, InlineSkate and BeetleFly show samples generated inside the mean envelope, but when inspected closer, the data does not look to follow any particular structure associated with the mean of the class. This can also be seen in DiatomSizeReduction, with the first class being volatile compared the the envelope, yet not demonstrating the same in the second and third class. One possible reason for these types of behavior in the generation of the new classes could be from the amount of training data in each class.

It is no surprise that the generation of more difficult data sets are easier accomplished when a extensive amount of training data is given to the GAN,

and when this is low the results are typically lower quality. One way going forward would be to look into data augmentation methods for time series data to generate more augmented data based off the original data set to try to help the GAN have more information to generalize the distribution to create better synthetic data.

One thing that needs explored more is the data sets where the generated time series data escapes the mean envelope. This is typically seen in the maximum of the generated samples and not so much on the minimum. One possible explanation could be the distribution of the random vector of noise fed into the generator to create the time series data. Looking at different distributions or different scales might result into better, tighter generated signals.

The second experiment we conducted was looking to see how the generated time series data compares when classified to the original time series data. For this we trained a Fully convolutional networks-Long Short Term Memory network (FCN-LSTM) on the data sets chosen and then used our generated data as test data to see if they categorized in the appropriate class [18]. A comparison of the original data set classification accuracy to the generated data set accuracy is shown in Tab. 3 along with the difference between data sets.

When looking at the table there is a very tight variance to the original data set accuracy, with some data sets having better accuracy than the original and some having poorer, however, none deviate more than 3.66% in either direction. This experiment validates two things. First, it shows that our generated time series data effectively fools the FCN-LSTM into thinking it was data similar to the classes it was trained on. This fortifies a use case for the GAN, generating for samples when an algorithm needs more training data to perform better classification. Second, it reinforces the visualization experiment with unique machine learning statistical means. To clarify, this table does not show an improvement based solely on our generated samples, rather we are comparing if the algorithm can classify our generated samples as well as it classifies the testing samples of the UCR data set.

5 Conclusion

With time series data being such an important aspect to machine learning and relating how events correlate to humans in every day life, it is a necessity for advanced algorithms that can perform tasks from classification to forecasting. It has been shown in the community today that deep learning methods have overtaken the field in performance metrics when working on time series data.

Where they fail, however, is when the training data is limited or the problem is not complex. As an effect from this, there is a cost to have agents or researchers to collect this data, label this data, and be able to put it in usable formats for the algorithms to work. This is where we look to leverage our 1D-WGAN-GP to generate more synthetic samples similar to the data we wish to use with deep

learning algorithms. To this we have shown in our paper our method has visually excelled in producing new samples that are similar to the original data set and even classify appropriately when used with a SOTA algorithm for time series classification.

Future work on this task would be to improve the generation of the time series data by looking at different loss functions or possibly different architectures for the generator network. These could look into different 1D distances in different domains or probability distributions or simply ensembling the loss functions into a cohesive loss function that carries the best trait of a multitude of loss functions in GAN training. We also feel exploring simple yet effective time series machine learning methods in prepossessing or learning and incorporating them within the GAN structure would produce better generated samples and possibly be able to further advance the 2D realm of generated imagery as well. There also needs to be an in depth analysis of how much the generated data actually affects the deep learning algorithm's performance on classification, clustering, or forecasting.

Our method shows promising results for generating the time series data and in theory should have enough variance from the original data for the new samples to not hinder the deep learning method causing overfitting, but rather strengthen the deep learning algorithm to perform better at its task.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680)
2. Mogren, O. (2016). C-RNN-GAN: Continuous recurrent neural networks with adversarial training. arXiv preprint arXiv:1611.09904.
3. Esteban, C., Hyland, S. L., Rtsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633.
4. Hartmann, K. G., Schirrmeister, R. T., Ball, T. (2018). EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals. arXiv preprint arXiv:1806.01875.
5. Brophy, E., Wang, Z., Ward, T. E. (2019). Quick and Easy Time Series Generation with Established Image-based GANs. arXiv preprint arXiv:1902.05624.
6. Akbari, M., Liang, J. (2018, April). Semi-recurrent CNN-based VAE-GAN for sequential data generation. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 2321-2325). IEEE.
7. Kulkarni, V., Garbinato, B. (2017, November). Generating synthetic mobility traffic using RNNs. In Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery (pp. 1-4). ACM.
8. Alzantot, M., Chakraborty, S., Srivastava, M. (2017, March). Sensegen: A deep learning architecture for synthetic sensor data generation. In 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 188-193). IEEE.
9. Donahue, C., McAuley, J., Puckette, M. (2018). Adversarial audio synthesis. arXiv preprint arXiv:1802.04208.
10. Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., Roberts, A. (2019). Gansynth: Adversarial neural audio synthesis. arXiv preprint arXiv:1902.08710.

11. Marafioti, A., Holighaus, N., Perraudin, N., Majdak, P. (2019). Adversarial Generation of Time-Frequency Features with application in audio synthesis. arXiv preprint arXiv:1902.04072.
12. Sadouk, L. (2018). CNN Approaches for Time Series classification. In Convolutional Neural Network. IntechOpen.
13. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery, 31(3), 606-660.
14. Wang, Z., Yan, W., Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.
15. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
16. Arjovsky, M., Chintala, S., Bottou, L. (2017). Wasserstein gan. arXiv preprint arXiv:1701.07875.
17. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A. C. (2017). Improved training of wasserstein gans. In Advances in Neural Information Processing Systems (pp. 5767-5777).
18. Karim, F., Majumdar, S., Darabi, H., Chen, S. (2017). LSTM fully convolutional networks for time series classification. IEEE Access, 6, 1662-1669.

Table 1. Some generated data sets from 1D-WGAN-GP compared to the mean envelope of the original data. Plotted per class for comparison, data sets with multiple classes were done by plotting the first three classes of the data set.

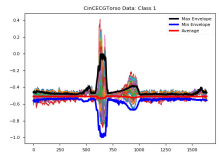
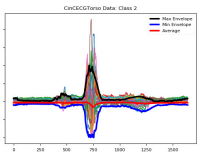
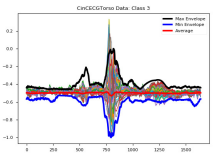
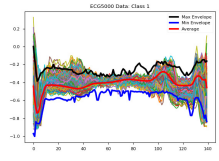
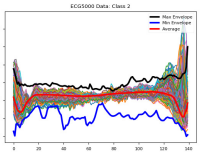
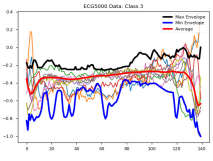
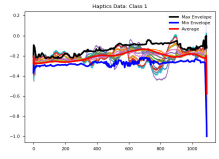
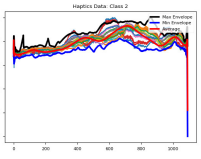
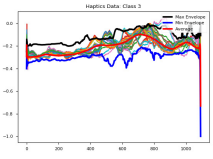
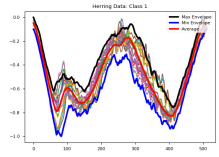
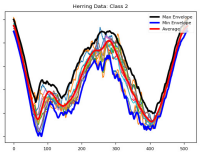
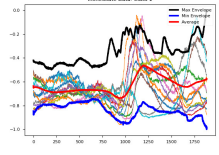
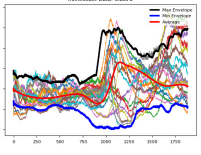
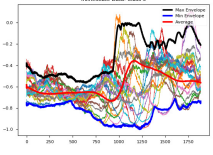
Data Set	Class1	Class2	Class3
CinCECGTorso			
ECG5000			
Haptics			
Herring			
InlineSkate			

Table 2. Continuation of the generated data sets from 1D-WGAN-GP compared to the mean envelope of the original data. Plotted per class for comparison and only up to three classes for visualization purposes.

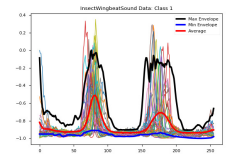
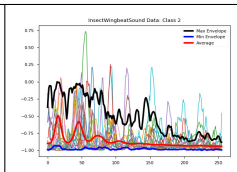
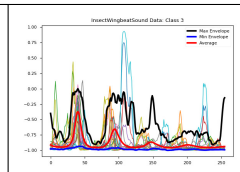
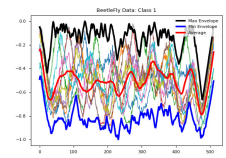
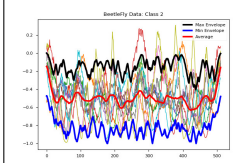
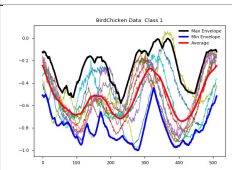
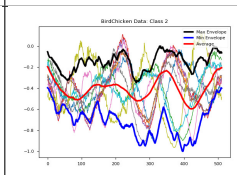
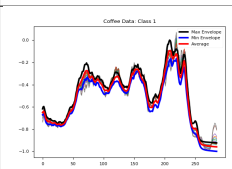
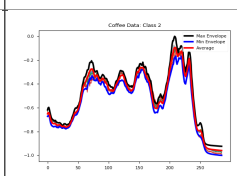
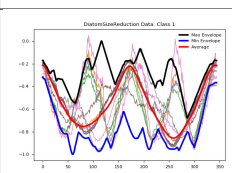
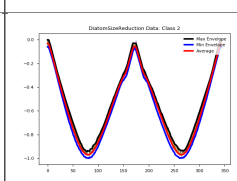
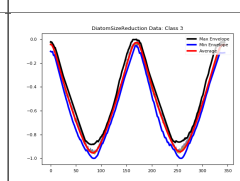
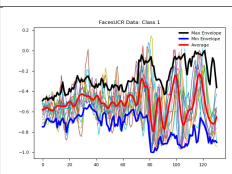
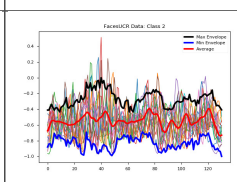
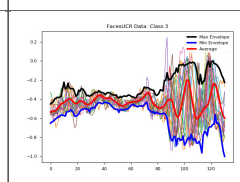
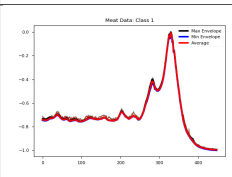
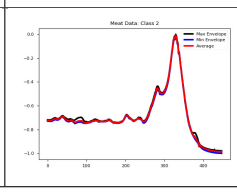
Data Set	Class1	Class2	Class3
InstWingbSnd			
BeetleFly			
BirdChicken			
Coffee			
DiaSizRedu			
FacesUCR			
Meat			

Table 3. FCN-LSTM classification results on original data sets and generated data sets, in percent accuracy.

Data Set	**Original Data % Acc.**	**Generated Data % Acc.**	**% Difference**
CinCECGTorso	90.94	94.23	+3.29
ECG5000	94.84	93.85	-0.99
Haptics	57.47	55.25	-2.22
Herring	76.56	78.62	+2.06
InlineSkate	46.55	49.34	+2.79
InsectWingsbeatsound	68.23	70.24	+2.01
BettleFly	100.0	100.0	0.00
BirdChicken	100.0	100.0	0.00
Coffee	100.0	100.0	0.00
DiatomSizeReduction	97.71	96.32	-1.39
FacesUCR	98.98	98.90	-0.08
Meat	100.0	100.0	0.00
ChloConc	100.0	97.83	-2.17
Fish	98.29	95.85	-2.44
FordA	97.33	98.71	+1.38
Ham	83.81	81.23	-2.58
GunPoint	100.0	100.0	0.00
Mallat	98.34	94.68	-3.66
NonInvThor1	96.54	94.85	-1.69
ShapesAll	91.83	91.60	-0.23
Wafer	100.0	100.0	0.00
ArrowHead	92.00	93.47	+1.47
Car	96.70	96.87	+0.17
CBF	100.0	99.58	-0.42
DistPhxAGp	86.00	85.75	-0.25
ECG5Days	99.42	99.17	-0.25
FacesAll	96.57	97.24	+0.67
OSULeaf	99.59	97.68	-1.91
Plane	100.0	100.0	0.00
ShapeletSim	100.0	100.0	0.00
StarlightCur	97.71	96.32	-1.39
ToeSeg1	99.12	96.85	-2.27
Yoga	91.90	90.28	-1.62

Forecasting using Big Data: The case of Spanish Tourism Demand

Author: Miguel Ángel Ruiz Reina [1]

Programa de Doctorado: Economía y Empresa

Facultad de Ciencias Económicas y Empresariales – Universidad de Málaga (UMA)

- Google queries gives information to forecasting Tourism Flows.
- VAR, VECM, SARIMA and ARDL with seasonality needs strong mathematical assumptions.
- Impulse-Response explain dynamics to some external change.
- Hierarchical Neural Networks models weak mathematical assumptions.
- Matrix U2 Theil is proposed as a novel technique for the selection of forecasting models in Big Data.
- The models have been tested with time horizon $h = 1, 3, 6, 12$.
- The results are contrasted and statistically significant.

Abstract:

The development of new technologies, especially in the digital field applied to Tourism Industry, has meant a revolutionary change in Data Analysis. Specifically, due to data obtained from social networks, environment, images, videos, sound or any tool from the Internet of Things (IoT). In this article, we will work in an applied way on the hotel demand in Spain and through the dynamic correlations generated worldwide. The Big Data set will come from Google applications. Granger-Causality test extended to seasonality allows determine causality relationships and through VAR (VECM) analysis when consumers decide to confirm their hotel booking. The endogenous or exogenous dynamics effects will be analysed by Impulse-Response Analysis. All this analysis will be compared with the results of ADRL multivariate models with seasonality, and univariate techniques such as SARIMA and Hierarchical Neural Networks. The best model of forecasting will be selected by Matrix U2 Theil. Tourism market is provided of knowledge through Data Analysis.

Acknowledgements: The author wishes to acknowledge the support given by the University of Malaga. Ph.D. Program in Economics and Business, effective from July 16, 2013. Especially to Professor Antonio Caparrós Ruiz from the Department of Statistics and Econometrics of the University of Málaga, for reviewing this work.

Keywords:

VAR; VECM; ARDL; Hierarchical Neural Networks; Seasonality; Matrix U2 Theil; Forecasting; Tourism Demand; Spain; Google; Big Data; Impulse-Response

1 Corresponding author.
E-mail address: ruizreina@uma.es
Full Postal address: Universidad de Málaga (UMA), Facultad de Ciencias Económicas y Empresariales. Departamento de Economía Aplicada (Econometría y Estadística). Calle El Ejido, 6, 29071 Málaga, Spain.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on Automatic Control, 19*(6), 716-723.

Andreas, N., & Fredrik, T. (2019). Supervised Machine Learning. Lecture notes for the Statistical Machine Learning course. Department of Information Technology, Uppsala University.

Andrews, D. (1988). Chi-Square Diagnostic Tests for Econometric Models: Theory. *Econometrica, 56*(6), 1419-1453. doi:10.2307/1913105

Artola, C., Pinto, F., & de Pedraza, P. (2015). Can internet searches forecast tourism inflows. *International Journal of Manpower, 36*(1), 103-116. doi:10.1108/IJM-12-2014-0259

Babu, S., & Subramoniam, S. (2016). Tourism Management in Internet of Things Era. *Journal of IT and Economic Development, 7*(1), 1-14.

Brown, R., Durbin, J., & Evans, J. (1975). Techniques for Testing the Constancy of Regression Relationships over Time. *Journal of the Royal Statistical Society, 37*(2), 149-192.

Buse, A. (1982). The Likelihood Ratio, Wald, and Langrange Multiplier Test: An Expository Note. *The American Statitician, 36*(3), 153-157.

Butler , D. (2013). Whern Google got Flu wrong. *Nature*(494), 155-156. doi:10.1038/494155a

Callen, T. (2017). *International Monetary Fund. Retrieved from https://www.imf.org/external/pubs/ft/fandd/basics/gdp.htm*.

Camacho, M., & Pacce, M. (2016). Forecasting travelers in Spain with Google queries. *16/21working paper*.

Camacho, M., & Pacce, M. (2017). Forecasting travellers in Spain with Google's search volume indices. *Tourism Economics*. doi:DOI: 10.1177/1354816617737227

Chancellor, S., & Counts, S. (2018). Measuring Employment Demand Using Internet Search Data. *CHI '18 Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* Montreal.

Chen, C., Shi, C., & Chen, J. (2018). Research on Tourism Network Index Model Based on Baidu Index: A Case Study of Sanya. retrieved from https://doi.org/10.1051/matecconf/201822805017. *MATEC Web of Conferences*, *228.*

Choi, H., & Varian, H. (2009). *Predicting the Present with Google Trends. Retrieved from https://static.googleusercontent.com/media/www.google.com/es//googleblogs/pdfs/google_predicting_the_present.pdf.* Google Inc.

Cisneros, J., & Fernández, A. (2015). Cultural tourism as tourits segment for reducing seasonality in a coastral area: the case study oif Andalusia. *Current Issues in Tourism, Vol. 18*(Num. 8), 765-784. doi:10.1080/13683500.2013.861810

Cook. (2019). Economist Network. *Forecast Evaluation using Theil´s Inequality Coefficients*. School of Management, Swansea University. Retrieved from https://www.economicsnetwork.ac.uk/showcase/cook_theil

Cook, S., Conrad, C., Fowkles, A., & Mohebbi, M. (2011). Assessing Google Flu Trends Performance in the United States during the 2009 Influenza Virus A (H1N1) Pandemic. *PLOS ONE, 6*(8), Published online. doi:10.1371/journal.pone.0023610

Davidson, A., & Yu, Y. (2005). The internet and the occidental tourist: analyses of Taiwan's tourism websites from the perspective of western tourists. *Information Technology & Tourism, 7*, 91-102.

Diebold, F., & Mariano, R. (1995). Comparing Predictive Accuracy. *Journal of Business and Economic Statistics, 13*, 253-63.

Dinis, G., Costa, C., & Pacheco, O. (2016). The Use of Google Trends Data as Proxy of Foreign Tourist Inflows to Portugal. *International Journal of Cultural and Digital Tourism*, 66-75.

Doan, A., Halevy, A., & Ives, Z. (2012). Principles of Data Integration. 173-207. doi:https://doi.org/10.1016/B978-0-12-416044-6.00007-7

Drago, C. (2017). Forecasting the Measured Perceived Touristic Interest Using Autoregressive Neural Networks and Big Data: the Case of Florence. *AIQUAV.*

Dunham, I. (2015). Big Data: A Revolution That Will Transform How We Live, Work, and Think. *The AAG Review of Books*, 19-21. doi:DOI: 10.1080/2325548X.2015.985533

Edgar, T., & Manz, D. (2017). Research Methods for Cyber Security. 153-173. doi:https://doi.org/10.1016/B978-0-12-805349-2.00006-6

Engle, R., & Granger, C. (1987). Co-Integration and Error Correction: Representation, Estimation, and Testing. *Econometrica, 55*(2), 251-276. doi:DOI: 10.2307/1913236

Ettredge, M., Gerdes, J., & karuga, G. (2005). Using web-based search data to predict macroeconomic statistics. *Communications of the ACM, 48*(11), 87-92. doi:10.1145/1096000.1096010

Garín, T. (2007). German demand for tourism in Spain. *Tourism Management*(28), 12-22.

Gawlik, E., Kabaria, H., & Kaur, S. (2011). *Predicting tourism trends with Google Insights. Retrieved from http://cs229.stanford.edu/proj2011/GawlikKaurKabaria-PredictingTourismTrendsWithGoogleInsights.pdf.* Retrieved from http://cs229.stanford.edu.

Giacomini, R., & White, H. (2006). Test of Conditional Predictive Ability. *Econometrica, 74*(6), 1545-1578. doi:https://doi.org/10.1111/j.1468-0262.2006.00718.x

Ginsberg, J., Mohebbi, M., Patel, R., Brammer, L., Smolinski, M., & Brilliant, L. (2009). Detecting influenza epidemics using search engine query data. *Nature International Journal of Science*(457), 1012-1014.

Goel, S., Hofman, J., Lahaie, S., Pennock, D., & Watts, D. (2010). Predicting consumer behavior with Web search. *Proceedings of the National Academy of Sciences of the United States of America*. doi:https://doi.org/10.1073/pnas.1005962107

Goldberger, A. (1991). A Course in Econometrics. *Harvard University Press, Cambridge, Massachusetts*, 178.

González, R. (2017). Hacking the citizenry?: Personality profiling, 'big data' and the election of Donald Trump. *Anthropology Today, 33*(3), 9-12. doi:https://doi.org/10.1111/1467-8322.12348

Götz, T., & Knetsch, T. (2017). *Google data in bridge equation models for German GDP.* Deutsche Bundesbank.

Granger, C. (1969). Investigating causal relations by econometric models and cross spectral methods. *37*(3), 424-438. doi:10.2307/1912791

Gunter, U., & Onder, i. (2016). Forecasting city arrivals with Google Analytics. *Annals of Tourism Research*. doi:10.1016/j.annals.2016.10.007

Guo, Y., Liu, H., & Chai, Y. (2014). The embedding convergence of smart cities and tourism Internet of Things in China: an advance respective. *Advances in Hospitality and Tourism Research (AHTR), 2*(1), 54-69.

Hanck, C., Arnold, M., Gerber, A., & Schmelzer, M. (2019). *Introduction to Econometrics with R. Retrieved from https://www.econometrics-with-r.org/index.html.*

Harvey, D., Leybourne, S., & Newbold, P. (1997). Testing the Equality of Prediction Mean Squared Errors. *International Journal of Forecasting, 13*, 281-291.

Helft, M. (2008). Is There a Privacy Risk in Google Flu Trends? *The New York Times*.

Huang, X., Zhang, L., & Ding, Y. (2016). The Baidu Index: Uses in predicting tourism flows eA case study of the Forbidden City. *Tourism Management, 58*, 301-306. doi:http://dx.doi.org/10.1016/j.tourman.2016.03.015

Hyndman, R., & Kochler, A. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting, 22*(4), 679-688. doi:https://doi.org/10.1016/j.ijforecast.2006.03.001

INE. (2017). *INE. Retrieved from https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736169169&menu=ultiDatos&idp=1254735576863*.

Jansen, B. J. (2006). Review of "The search: how google and its rivals rewrote the rules of business and transformed our culture" John Batelle, Peeguin Group. *Information Processing and Management: an International Journal, 42*(5), 1399-1401.

Kožić, I., Sorić, P., & Sever, I. (2018). Interdependence of international tourism demand for Mediterranean countries: Impact of demand shocks. *International Journal of Tourism Research*, 97-107. doi: https://doi.org/10.1002/jtr.2244

Li, C., Song, H., & Wit, S. (2005). Recent Developments in Econometric Modeling and Forecasting. *Journal of Travel Research, 44*(1), 1-29. doi:10.1177/0047287505276594

Li, S., Qiu, R., & Chen, L. (2008). Cyberspace attention of tourist attractions based on Baidu index: temporal distribution and precursor effect. *Geography and Geo-Information Science, 24*(6), 102-107.

Li, X., Pan, B., Law, R., & Hyang, X. (2017, April 1). Forecasting tourism demand with composite search index. *Tourism Management, 59*, 57-66. doi:DOI: 10.1016/j.tourman.2016.07.005

Li, X., Pan, B., Law, R., & Hyang, X. (2017). Forecasting tourism demand with composite search index. *Tourism Management, 59*, 57-66. doi:DOI: 10.1016/j.tourman.2016.07.005

Li, X., Shang, W., Wang, S., & Ma, J. (2015). A MIDAS modelling framework for Chinese inflation index forecast incorporating Google search data. *Electronic Commerce Research and Applications, 14*(2), 112-125. doi:10.1016/j.elerap.2015.01.001

Lim, C., Alananzeh, O., & Hua, K. (2019). Perceptions of Risk and Outbound Tourism Travel Intentions among Young Working Malaysians. *Human and Social Sciences, 46*(1), 365-379.

Lin, Y. (2011). The Application of the Internet of Things in Hainan Tourism Scenic Spot. *Seventh International Conference on Computational Intelligence and Security.*

Lindholm, A., Wahlström, N., Lindsten, F., & Schön, T. (2019). *Supervised Machine Learning.* Department of Information Technology, Uppsala University.

Liu, J., Li, X., & Guo, Y. (2017). Periodicity analysis and a model structure for consumer behavior on hotel online search interest in the US. *International Journal of Contemporary Hospitality Management, 29*(5), 1486-1500. doi:10.1108/IJCHM-06-2015-0280

Lu, Z., & Liu, N. (2007). The guiding effect of information flow of Australian tourism website on tourist flow: process, intensity and mechanism. *Human Geography, 22*(5), 88-93.

Lu, Z., Zhao, Y., Wu, S., & Hang, B. (2007). The time distribution and guide analysis of visiting behavior of tourism website user. *Acta Geographica Sinica*, 621-630.

Lütkepohl , H. (2010). New Introduction to Multiple Time Series Analysis. (Springer, Ed.)

Lütkepohl, H. (2011). Vector Autoregressive Models. Economics Working Paper ECO 2011/30. European University Institute.

Ma, L., & Wu, F. (2005). Restructuring the Chinese City: Changing Society, Economy and Space. *Routledge Press, Taylor and Francis Group*, 34-51.

Ma, L., Sung, G., Huang, Y., & Zhou, R. (2011). A correlative analysis on the relationship between domestic tourists and network attention. *Economic Geography, 31*(4), 680-685.

Malhotra, N., Baalbaki, I., & Bechwati, N. (2010). *Marketing Research.* Retrieved from http://www.pearsonmiddleeastawe.com/pdfs/SAMPLE-marketing-research.pdf

Martín, J. M., Jiménez, J. d., & Molina, V. (2014). mpacts of Seasonality on Environmental Sustainability in the Tourism Sector Based on Destination Type: An Application to Spain'S Andalusia Region. *Tourism Economics, 20*(1), 123-142.

McCown, F. (2016). Harding University. Retrieved from https://www.harding.edu/fmccown/short-history-of-computing.pdf.

McKellips, F. (2017). Nowcasting the Unemployment Rate in Canada Using Google Trends Data. Retrieved from http://ifsd.ca/web/default/files/Presentations/Reports/17012%20-%20Nowcasting%20Unemployment%20Rate%20with%20Google%20Trends%20-%20Final.pdf.

McKercher, B. (2016). Towards a taxonomy of tourism products. *Tourism Management, 54*, 196-208. doi:https://doi.org/10.1016/j.tourman.2015.11.008

Miah, S., Vu, H., Gammack, J., & McGrath, M. (2017). A Big Data Analytics Method for Tourist Behaviour Analysis. *Information & Management, 54*(6). doi:https://doi.org/10.1016/j.im.2016.11.011

Montero, R. (2013). Test de Causalidad. *Documentos de Trabajo en Economía Aplicada. Universidad de Granada. España*.

Newey, W. K., & West, K. D. (1987). A Simple, Positive Semi-definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica, 55*(3), 703-708. doi:http://dx.doi.org/10.2307%2F1913610

Nilsson, N. (1998). Artificial Intelligence: A New Synthesis. Morgan Kaufmann Publishers, Inc. doi:https://doi.org/10.1016/B978-0-08-049945-1.50026-5

Nkoro, E., & Uko, A. (2016). Autoregressive Distributed Lag (ARDL) cointegration technique: application and interpretation. *Journal of Statistical and Econometric Methods*, 63-91.

Onder, I. (2017). Forecasting tourism demand with Google trends: Accuracy comparison of countries versus cities. *International Journal of Tourism Research*, 1-39. doi:10.1002/jtr.2137

Onder, I., & Gunter, U. (2015). Forecasting Tourism Demand with Google Trends: The Case of Vienna. *Tourism Analysis*. doi:10.3727/108354216X14559233984773

Palos-Sanchez, P., & Correia, M. (2018). The Collaborative Economy Based Analysis of Demand: Study of Airbnb Case in Spain and Portugal. *Journal of Theoretical and Applied Electronic Commerce Research, 13*(3), 85-98. doi:10.4067/S0718-18762018000300105

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems (Revised Second Printing). Elsevier science & technology. doi:https://doi.org/10.1016/B978-0-08-051489-5.50012-6

Peng, B., Song, H., & Crouch, G. I. (2014). A meta-analysis of international tourism. *Tourism Management*, 181-183. doi:http://dx.doi.org/10.1016/j.tourman.2014.04.005

Plackett, R. (1983). Karl Pearson and the Chi-Squared Test. *International Statistical Review, 51*(1), 59-72. Retrieved from https://www.jstor.org/stable/1402731

Plat, A. (2015). *Data Science and Ebola.* Inaugural Lecture, Universiteit Leidenon . Retrieved from https://www.researchgate.net/publication/274744049_Data_Science_and_Ebola

Pyo, D.-J. (2017). Can Big Data Help Predict Financial Market Dynamics?: Evidence from the Korean Stock Market. *journal East Asian Economic Review*, 147-165. doi:http://dx.doi.org/10.11644/KIEP.EAER.2017.21.2.327

Rödel, E. (2017). *Forecasting tourism demand in Amsterdam with Google Trends.* Master Business Administration.

Shimshoni, Y., Efron, N., & Matias, Y. (2009). *On the Predictability of Search Trends. Retrieved from https://www.researchgate.net/publication/238115677_On_the_Predictability_of_Search_Trends.* Google, Israel Labs.

Sims, C. (1980). Macroeconomics and reality. *Econometrica*, 1-48. doi:https://doi.org/10.2307/1912017

Song, H., & Li, G. (2008). Tourism demand modelling and forecasting- A review of Recent research. *Tourism Management, 29*(2). doi:10.1016/j.tourman.2007.07.016

Song, H., & Liu, H. (2017). Predicting Tourism Demand Using Big Data. 13-29. doi:https://doi.org/10.1007/978-3-319-44263-1_2

Sun, S., Wei, Y., Tsui, K.-L., & Wang, S. (2019). Forecasting tourist arrivals with machine learning and internet search index. *Tourism Management, 70*, 1-10. Retrieved from https://doi.org/10.1016/j.tourman.2018.07.010

Sung, S., Wei, Y., Tsui, K.-L., & Wang, S. (2019). Forecasting tourist arrivals with machine learning and internet search index. *Tourism Management, 70*, 1-10. doi:https://doi.org/10.1016/j.tourman.2018.07.010

Talia, D., Trunfio, P., & Marozzo, F. (2016). Data Analysis in the Cloud. Computer Science Reviews and Trends. 77-122. doi:https://doi.org/10.1016/B978-0-12-802881-0.00004-4

Tang, H., Qiu, Y., & Liu, J. (2018). Comparison of Periodic Behavior of Consumer Online Searches for Restaurants in the U.S. and China Based on Search Engine Data. *IEEE Access*. doi:10.1109/ACCESS.2018.2832196

Theil, H. (1978). *Introductions to Econometrics.* New Jersey: Prentice-Hall, Englewood Cliffs.

Tkacz, G. (2013). Predicting Recessions in Real-Time: Mining Google Trends and Electronic Payments Data for Clues. Revieved from https://www.cdhowe.org/sites/default/files/attachments/research_papers/mixed/Commentary_387_0.pdf. *Financial Services*.

Torraleja, F., Vázquez, A., & Franco, M. (2009). Flows into tourist areas: An econometric approach. *International Journal of Tourism research*, 1-15. doi:https://doi.org/10.1002/jtr.657

Trish, B. (2018). Big Data under Obama and Trump: The Data-Fueled U.S. Presidency. *Politics and Governance, 6*(4), 29-38. doi:DOI: 10.17645/pag.v6i4.1565

Tuhkuri, J. (2015). *Big Data: Do Google Searches Predict Unemployment?* Master's Thesis, University of Helsinki.

Tuhkuri, J. (2017). *Big Data: Google Searches Predict Unemployment in Finland.*

Verbeek, M. (2017). *A guide to modern Econometrics* (Fith Edition ed.). Wiley.

Wang, J., Ma, Y., Zhang, L., Gao, R., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems, 48, Part C*, 144-156. doi:https://doi.org/10.1016/j.jmsy.2018.01.003

Wong, C., & Mckercher, B. (2012). Day tour itineraries: Searching for the balance between commercial needs and experiential desires. *Tourism Management, 33*(6), 1360-1372. doi:https://doi.org/10.1016/j.tourman.2011.12.019

Wong, K., Song, H., & Chon, K. (2006). Bayesian models for tourism demand forecasting. *Tourism Management, 5*, 773-780.

Xiaoying Jiao, E., & Li Chen, J. (2018). Tourism forecasting: A review of methodological developments over the last decade. *Tourism Economics, XX*(X), 1-24. doi:DOI: 10.1177/1354816618812588

Yang, X., Pan, B., Evans, J., & Benfu, L. (2015). Forecasting Chinese Tourist volume with search engine data. *Tourism Management*. doi:10.1016/j.tourman.2014.07.019

Zeiger, R. (Director). (2009). *Google Flu Trends Overview* [Motion Picture]. Retrieved march 11th, 2019, from https://www.youtube.com/watch?v=6111nS66Dpk

Zeynalov, A. (2017). Forecasting Tourist Arrivals in Prague: Google Econometrics. *Munich Personal RePEc Archive*. Retrieved from https://mpra.ub.uni-muenchen.de/83268/

Zivot, E. (2000, June). The Power of Single Equation Tests for Cointegration When the Cointegrating Vector Is Prespecified. *Econometric Theory, 16*(3), 407-439. Retrieved from https://www.jstor.org/stable/3533230?seq=1#page_scan_tab_contents

Estimation of the crustal velocity field in the Balanegra fault from GPS position time series in 2006 - 2018

A.J. Gil[1,2,3], A. Sánchez-Alzola[4], J. Galindo-Zaldívar[5,6], M.J. Borque[1,2,3], M.C. Lacy[1,2,3], M. Avilés[1,2], P. Alfaro[7], A.C. López Garrido[6], C. Sanz de Galdeano[6], A. Herrera[2], F. Chacón[2], M. Madrigal[2], S. Blanca[2], J.C. Moreno[2], V. Tendero[5]

[1] Centro de Estudios Avanzados en Ciencias de la Tierra (CEACTierra), Univ. Jaén, Spain
[2] Grupo de Investigación RNM282-Microgeodesia Jaén, Univ. Jaén, Spain
[3] Departamento de Ingeniería Cartográfica, Geodésica y Fotogrametría, Univ. Jaén, Spain
[4] Departamento de Estadística e Investigación Operativa, Univ. Cádiz, Spain
[5] Departamento de Geodinámica, Univ. Granada, Spain
[6] Instituto Andaluz de Ciencias de la Tierra (CSIC, Univ. Granada), Spain
[7] Departamento de Ciencias de la Tierra y del Medio Ambiente, Univ. Alicante, Spain

Abstract. In 2006 a non-continuos GPS network was installed to monitor the Balanegra fault, one of the most active faults recognized in the Campo de Dalias area. This is a zone with relevant seismicity associated with the active tectonic deformations of the southern boundary of the Betic Cordillera. The goal of our research is to constrain the activity of this fault from high quality GPS measurements to obtain precise deformation rates. Here we show the first results computed from five GPS campaigns in the timespan 2006-2018. These results agree with the general movement of the area. Moreover, we have identified a residual velocity field with values ranging from 1.7 to 3.0 mm/yr. There is a clear geodetic evidence of recent and active deformation by folding and faulting in the eastern Betic Cordillera, which currently undergoes the NNW-SSE convergence of the Eurasian and African plates, while also affected by orthogonal extension. The integration of further geologic and seismic information with the GPS site rates will allow a better understanding of the kinematics of this interesting geologic structure.

Keywords: PPP, GPS position time series, crustal deformation, Balanegra fault

1 The Balanegra Fault and the GPS network

The study of low rate deformation tectonic structures is of great relevance because although they are related with high seismic risk it is very difficult to accurately determine the deformation rates [1]. The Balanegra fault is a geologic structures located in the southeastern region of the Betic Cordillera [2]. The village of Adra and its surroundings near the Balanegra fault have undergone long periods of seismic activity, particularly in the past two centuries. To observe the dynamic of this fault, five sites named 940, 960, 970, 980 and 990 were built in 2006 [3].

The markers were positioned in well-defined areas on both geologic blocks of the fault. Each GPS site consists of a benchmark anchored to solid rock. An aluminium tube of 0.5 m was screwed on with the GPS antenna located at the top during the measurements. Only site 980 is a classical pillar. Five GPS campaigns were carried out in 2006, 2011, 2016, 2017 and 2018. Simultaneous recording was done during at least 72 hours in each campaign.

The GPS equipment used in 2006 and 2011 campaigns was Leica Geosystems GX1230 receivers with LEIAX1202 antennas whereas in 2016, 2017 and 2018 campaigns LEICA Geosystems AR10 receivers and LEIAR10 antennas were used.

Figure 1 shows the geological map and the monitoring network and figure 2 the site 990 tracking the GPS constellation during the 2018 campaign.

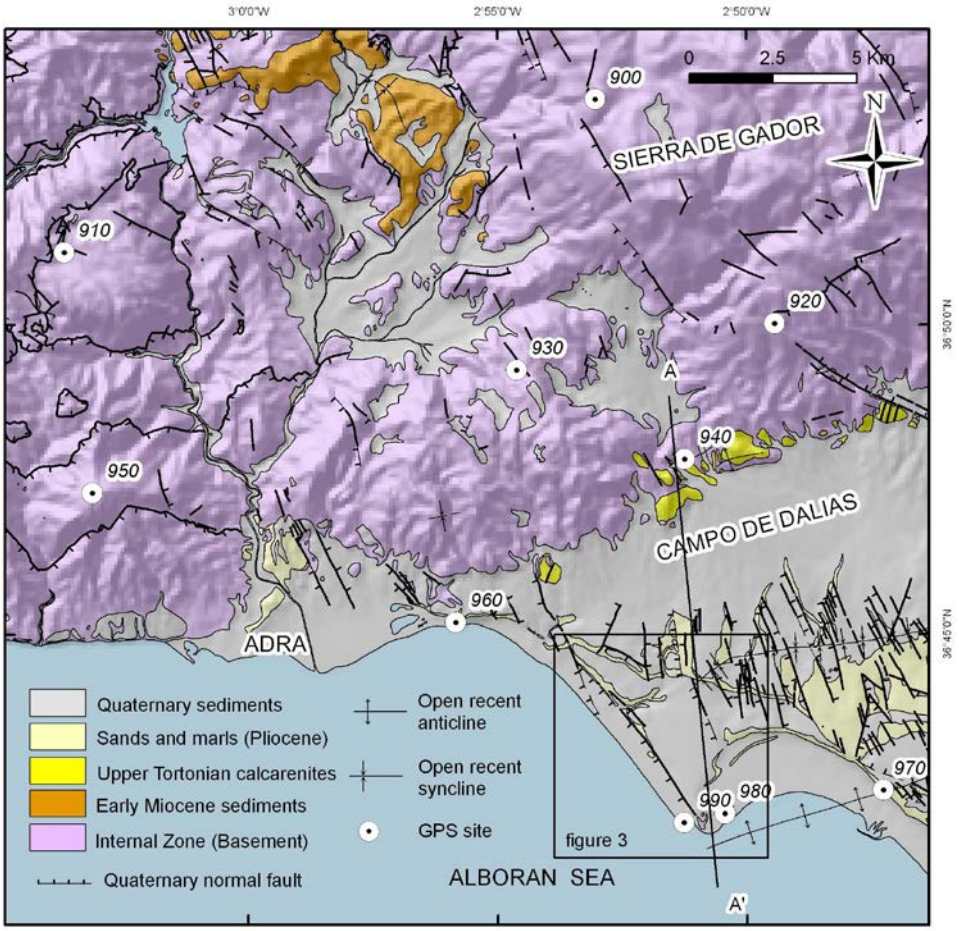


Fig. 1. Geological Map and GPS sites.

To process the GPS data we used Precise Point Positioning (PPP) methodology and the GIPSY-OASIS software, version 6.4, developed by the JPL (https://gipsy-oasis.jpl.nasa.gov/). The processing strategy uses zero-ambiguity resolution strategy [4]. A similar standard procedure for all five campaigns was used. We have considered JPL final ephemeris and pole products, FES2004 ocean tide loading model [5], hydrostatic and wet components of tropospheric delay, a 10º cut-off angle and calibrated data of antenna phase center.

Fig. 2. Site 990 during a GPS campaign.

2 GPS position time series

All the site coordinates were computed in the IGS14 reference frame, then we created the position time series of the network sites in North and East components. Here we also considered the changes of antennas between 2011 and 2016 campaigns, which produced a shift in the time series. Figure 3 shows the position time series in horizontal components (North and East) for all the sites of the network in centimetres. The model applied to the original time series, using weighted least squares, consists of an intercept, a site rate and an offset to account for antenna change. The error term is composed of white noise and temporally correlated random error. The colored noise is described by a random-walk process. We have assumed a typical magnitude for this process of 1.0 mm/$\sqrt{}$yr. The GPS-derived site rates and uncertainties are shown in table 1. A more effective representation of the velocity field was estimated through the residual velocities with respect to stable Eurasia. This residual velocity field gave us important information about the Balanegra fault.

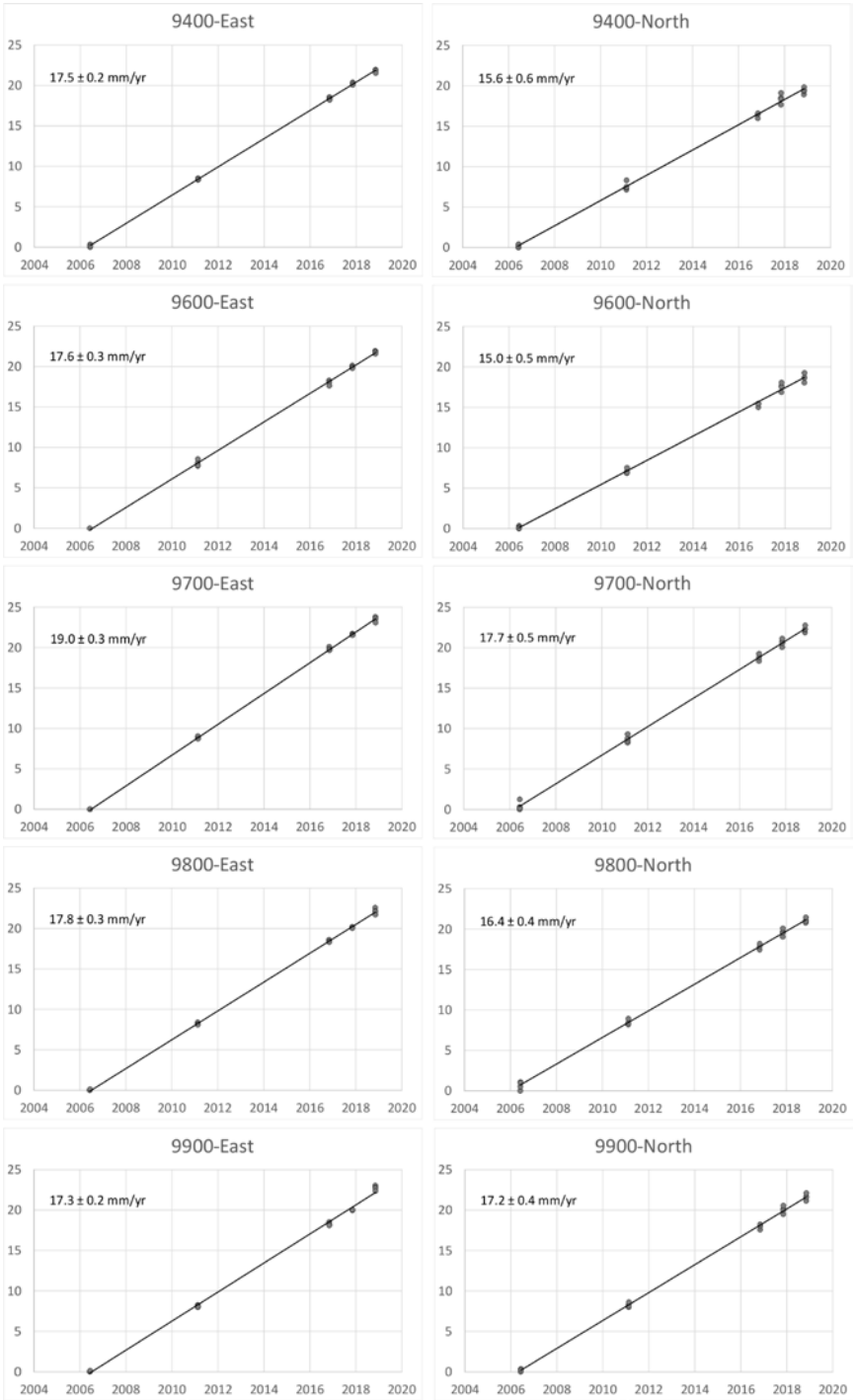


Fig. 3. Position time series in East and North components for all sites in cm.

3 First results and analysis

Table 1 lists the absolute velocities and 1σ uncertainties, as well as the residual velocities for all the sites. The figure 4 shows the absolute velocity field from GPS position time series in the IGS14 reference frame. The figure 5 presents the velocity field with respect to the Eurasian plate.

We have obtained absolute velocities in agreement with the general movement of the area: mean of 17.9 mm/yr East and 16.4 mm/yr North components. The differences with the EURASIA plate motion model allow us to compute a residual velocity field with values ranging from 1.7 mm/yr to 3 mm/yr, showing the deformation pattern of the area.

There is a clear geological evidence of recent and active deformation by folding and faulting in the eastern Betic Cordillera, which currently undergoes the NNW-SSE convergence of the Eurasian and African plates, while also affected by orthogonal extension.

Table. 1. Absolute and residual velocities for GPS sites.

Site ID	Velocity (mm yr^{-1})		Uncertainty (mm yr^{-1})		Res. Velocity (mm yr^{-1})	
	East	North	East	North	East	North
940	17.5	15.6	±0.3	±0.7	-2.7	-0.9
960	17.6	15.0	±0.4	±0.6	-2.5	-1.6
970	19.0	17.7	±0.4	±0.6	-1.2	1.2
980	17.8	16.4	±0.4	±0.5	-2.4	-0.1
990	17.3	17.2	±0.3	±0.5	-2.9	0.7

4 Conclusions

Our research reveals that our statistical methodology and GPS analysis technique have been able to measure the movements of a geologic structure from position time series at millimeter level. GPS rates show the complex interaction of active faults and folds in the Campo de Dalías region. The NNW-SSE Eurasian-Nubian convergence is the responsible of the development of ENE-WSW folds that produced a NW displacement of the southernmost sites. Moreover, the Campo de Dalías undergoes an ENE-WSW extension, in agreement with the recent most active faults. In this framework, the normal Balanegra fault has also a marked oblique dextral kinematics. Longer time series from new GPS campaigns will improve the estimations of the velocity field and will allow us to confirm these results.

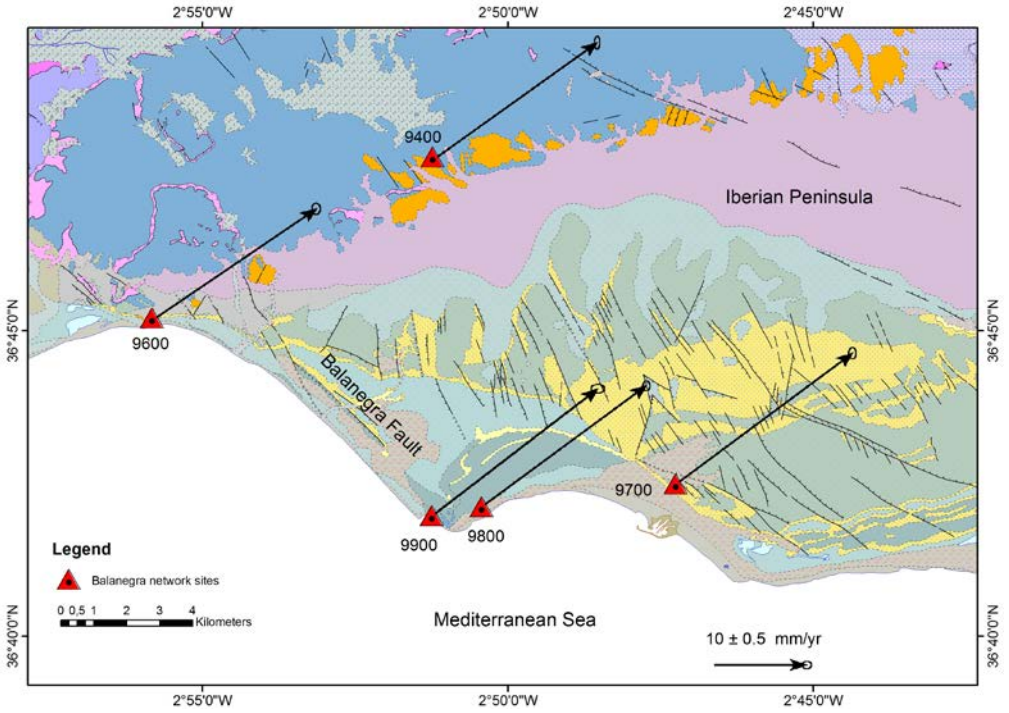


Fig. 4. Absolute velocity field from position time series in the IGS14 reference frame.

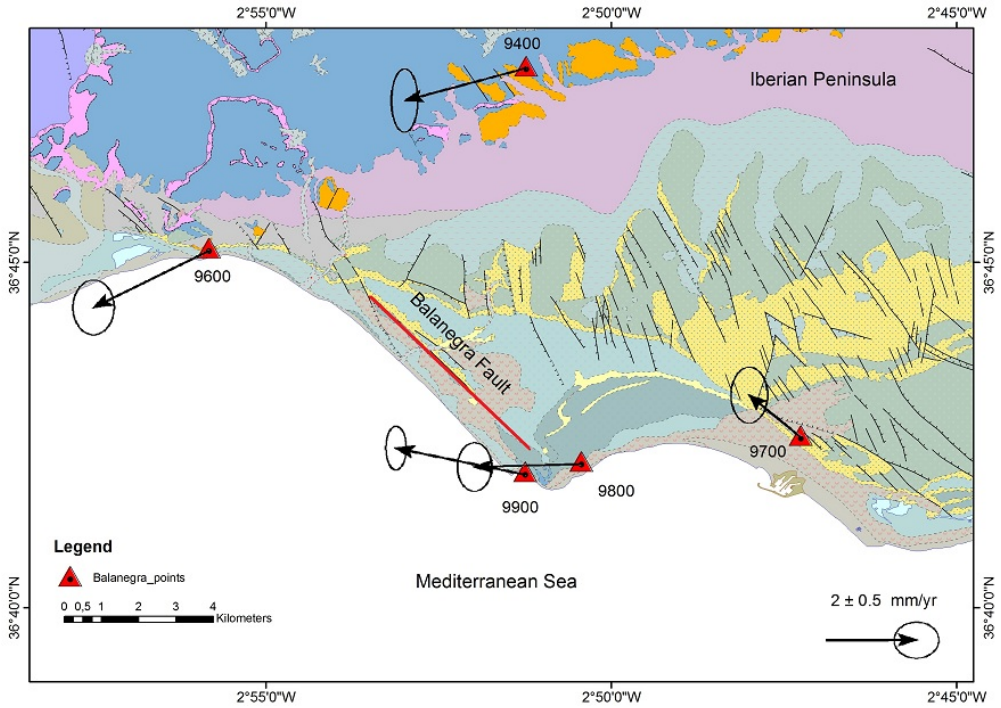


Fig. 5. Residual velocity field with respect to Eurasian plate.

Aknowledgements

PAIUJA 2019/2020 and CEACTierra supported this research. We thank all observers who collaborated in the field surveys.

References

1. Galindo-Zaldívar, J.; Borque, M.J.; Pedrera, A.; Marín-Lechado, C.; Gil, A.J.; López-Garrido, A.C. (2013). Deformation behaviour of the low-rate active Balanegra Fault Zone from high-precision levelling (Betic Cordillera, SE Spain). *J. Geodyn*., 71, 43- 51.

2. Marín-Lechado, C.; Galindo-Zaldívar, J.; Rodríguez-Fernández, L.R.; Serrano, I.; Pedrera, A. (2005). Active faults, seismicity and stresses in an internal boundary of a tectonic arc (Campo de Dalías and Níjar, southeastern Betic Cordilleras, Spain), *Tectonophysics*, 396, 81–96.

3. Marín-Lechado, C.; Galindo-Zaldívar, J.; Gil, A.J.; Borque, M.J., de Lacy, C; Pedrera, A.; López-Garrido, A.C.; Alfaro, P.; García-Tortosa, F.; Ramos, I.; Rodríguez-Caderot,G.; Rodríguez-Fernández, J.; Ruiz-Costán,A.; Sanz de Galdeano-Equiza, C. (2010). Levelling profiles and a GPS network to monitor the active folding and faulting deformation in the Campo de Dalias (Betic Cordillera, Southeastern Spain). *Sensors*, 10, 3504-3518.

4. Bertiger,W., Desai, S., Haines, B., Harvey, N., Moore, A., Owen, S., Weiss, J. (2010). Single receiver phase ambiguity resolution with GPS data. *J. Geod*. 84 (5), 327–337.

5. Lyard, F., Lefevre, F., Letellier, T., Francis, O. (2006). Modelling the global ocean tides: modern insights from FES2004. *Ocean Dyn*. 56, 394–415.

Electricity Load Forecasting - An Evaluation of Simple 1D-CNN Network Structures

Christian Lang[1,2], Florian Steinborn[1], Oliver Steffens[2], and Elmar W. Lang[1]

[1] Regensburg Universität, Regensburg, Germany,
christian3.lang@ur.de,
[2] OTH Regensburg, Regensburg, Germany

Abstract. This paper presents a convolutional neural network (CNN) which can be used for forecasting electricity load profiles 36 hours into the future. In contrast to well established CNN architectures, the input data is one-dimensional. A parameter scanning of network parameters is conducted in order to gain information about the influence of the kernel size, number of filters, and dense size. The results show that a good forecast quality can already be achieved with basic CNN architectures. The method works not only for smooth sum loads of many hundred consumers, but also for the load of apartment buildings.

Keywords: energy load forecasting, STLF, neural networks, CNN, convolutional networks

1 Introduction

There is no dispute in the scientific community that human-induced climate change is real. The effects of climate change are for example rising sea levels, an increasing CO_2 concentration in the atmosphere, and more regularly occurring extreme weather events, to name only few of them.[1, 2] To slow down and stop the global warming, it is crucial to reduce the generation of greenhouse gases, especially in energy production. One of the keys to accomplish a reduction is to establish more renewable energies in the energy market. By doing so, power plants that produce high levels of CO_2, like coal power plants, can in the long term be substituted by renewable energy sources. Another key to minimise the emission of greenhouse gases is to decrease the total energy consumption and to increase energy efficiency in consumption and production.

In the research project MAGGIE [3, 4], we try to address all of the above mentioned challenges. The goal of the research project is to energetically modernise existing historic apartment buildings and draft a concept for sector coupling in city districts. In the first step, one exemplary building will be modernised and evaluated. Afterwards, the whole city district will be modernised in a similar manner. In order to decrease the heat consumption of the building the thermal insulation is renewed and in the course of the research project new insulations are in development. In addition, a new heating system (see fig. 1) with an innovative control system is implemented. This heating system allows increases in

energy efficiency and can help integrate renewable energy sources into the power market. The core of the system is a combined heat and power plant (CHP), and a heat pump. All of them generate thermal energy, the heat pump from electricity and the CHP from fuel. The thermal energy is used to heat the water of a buffer storage, which is then, in combination with a heat exchanger, used as process and drinking water. In addition, the CHP generates electricity, as does a photovoltaic system (PV system) installed on the roof of the building, which can then be used to either power the heat pump or supply the habitants with electricity. A connection to the power grid receives surplus electricity and ensures there is always enough electricity available.

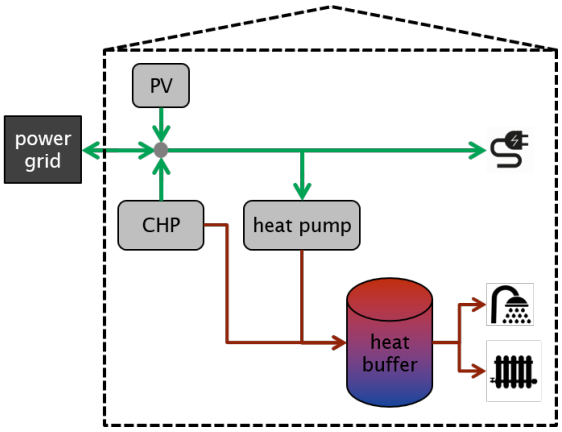


Fig. 1: Schematic of the new heating system. The red lines symbolise heat transport using water and the green lines symbolise electricity transport.

By utilising both forms of generated energy, the total energy efficiency of the system is nowadays higher than that of a conventional heating system, for example a gas-fired boiler, in combination with electricity from the power grid. [5]

All parts of the energy system are monitored continuously and can be controlled independently and remotely by the control system, which allows one to shift the production and consumption of heat and electricity in time and between the participants of the system by heating and using the water at the needed times. This allows for optimisation of the machine schedules depending on an optimisation target. Those targets can, for example, be self-sufficiency or cost-reduction.

After the modernisation of the entire city district, the energy systems of all houses in the district or even of several districts can act as a virtual power plant (VPP). This VPP can then work as a base load power plant or can help to stabilise the power grid and therefore assist to integrate renewable energies.

The main challenge of the system consists in knowing the electric and heat load of the building and its inhabitants. The loads are crucial for schedule optimisation as the feed-in into the power-grid and the consumption from the power grid have to be reported to the power grid operator the day before at midday in a 15 minute grid. Deviations in the heat load can be buffered with the heat buffer, deviations in the electric load, however, cannot be buffered in any way. Therefore, the focus of this paper is on forecasting electricity loads.

2 Smart Meter Data

In two directives [6, 7], the EU outlined their decision to establish SmartMeter devices in the energy sector throughout the entire European Union with the aim to enable customers to better monitor and manage their consumptional behaviour. A SmartMeter, in contrast with a conventional electric meter, records the energy consumption at least every 15 minutes or in even shorter periods. In this paper, the data of the CER Smart Metering project [8] is used. The dataset consists of individual SmartMeter data from over 5 000 Irish homes and businesses recorded for 18 months.

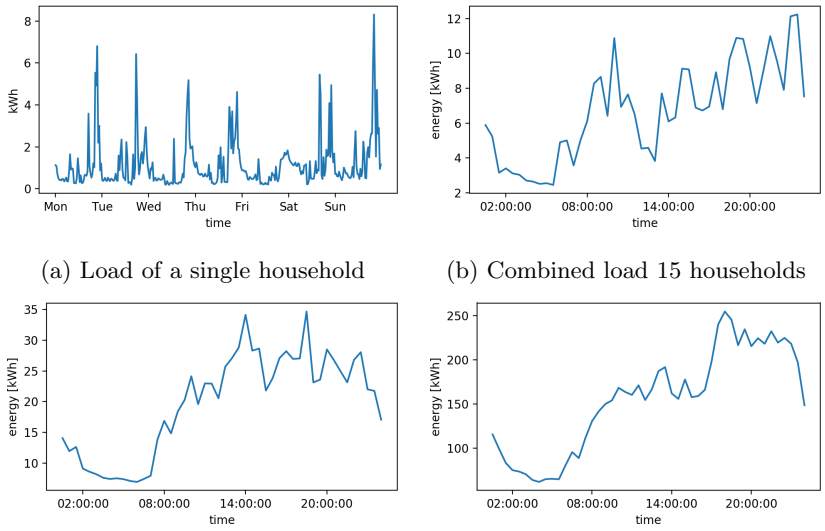
As the electric load of a single household is highly volatile and therefore impossible to predict, sum load time series of 15, 40, and 350 randomly picked households were created. Those time series correspond to a small apartment building, a big apartment building, and a whole city district. Figure 2 shows an exemplary day of the mentioned time series. In this work we focus on the load time series of 40 and 350 households.

3 Importance of Time Series Forecasting

The electric load forecast is crucial in order to fully utilise the possibilities of the implemented heating system and similar systems. Without a good forecast, a part of the heat buffer capacity has to be withheld in order to balance the deviation in the electric load by the CHP. The prediction horizon is $h = 144$ samples as 36 h have to be predicted in a 15 min grid.

There are already several publications about time series forecasting and short-term load forecasting (STLF). [9, 10] However, most of the methods predict either only one or very few time steps in the future, or are applied on load time series of whole cities/states which are, due to the properties of statistics, way smoother than the load time series of one building. Those smooth time series can be described properly with statistical methods when external factors (e.g. the weather) are taken into account. Therefore, their shape and features are also easy for neural networks to learn. None of the methods mentioned in recent publications, however, are designed to predict the electric load of only one building.

In the next chapter we propose the use of Convolutional Neural Networks (CNN) for time series prediction and report the first results of different network structures.



(a) Load of a single household

(b) Combined load 15 households

(c) Combined load 40 households

(d) Combined load 350 households

Fig. 2: (a) shows a load time series of a single household. (b)-(d) show each an exemplary day of the combined load time series. The different extracts display how volatile the load of a single household is and that the volatility decreases when households are combined.

4 Convolutional Neural Networks

Convolutional Networks, in the way they are used today, were first introduced by LeCun et al.[11] for zip code recognition. Since then, they were further developed and are now the standard for image and pattern recognition.

CNNs consist of convolutional layers, pooling layers, and fully-connected layers. In the convolutional layers, a set of feature maps, also called activation maps, are created. Each neuron in the feature map is only connected to a subset of neurons in its input layer. All neurons of the feature map share the same weights, thereby reducing the number of parameters significantly compared to a fully-connected neural network. In the most common CNN architectures, pooling layers alternate with convolutional layers. The pooling layer reduces the spatial dimension of the feature maps for the next computational steps in order to minimise the computational load and to avoid overfitting. At the end of the network, after an arbitrary number of the prior layers, fully-connected layers combine the resulting feature maps and return a classification measure. [11, 12]

5 Forecasting with CNNs

CNNs are traditionally used for image and pattern recognition by extracting features from two-dimensional data. In our research, we use a similar architecture for the forecasting of one-dimensional time series. The basic idea is that the convolutional layers extract features. These features are then combined by one or more fully-connected layers, and finally a forecast is created based on the classification of the last fully-connected layer (see fig 3). The pooling layers are omitted because an excessive amount of parameters is not a problem for one-dimensional data and the necessity of pooling layers is questioned in recent research [13].

A forecast can be created in two different ways, either directly or iteratively. A direct forecast means the network generates the desired forecast at once. Thus, the number of neurons in the output layer equal the prediction horizon h. When the forecast is generated iteratively, only one time step is predicted by the network. Then, the predicted point is appended to the input data and the first data point of the input is cut off, so that the new input has the required shape. Based on the new input, the next point is predicted. This procedure is repeated until h data points are predicted.

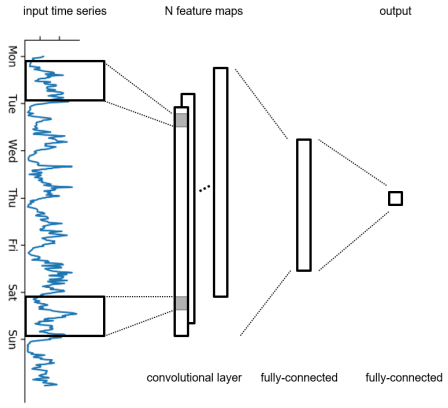


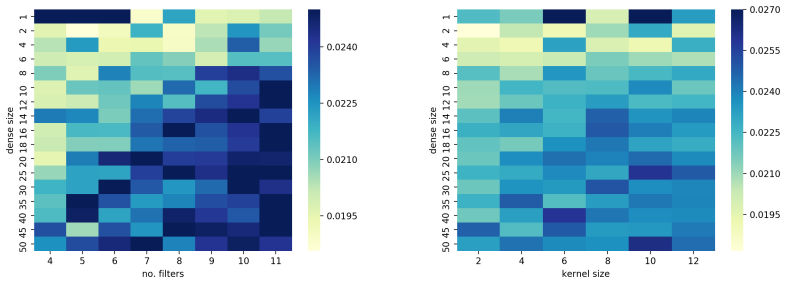
Fig. 3: Principle architecture of the used neural network.

6 Evalution of different network structures and training parameters

In order to get a better understanding of how the 1D-CNNs process data and how the network architecture influences the results, are the first tests conducted with very basic networks. They are built from one convolutional layer followed

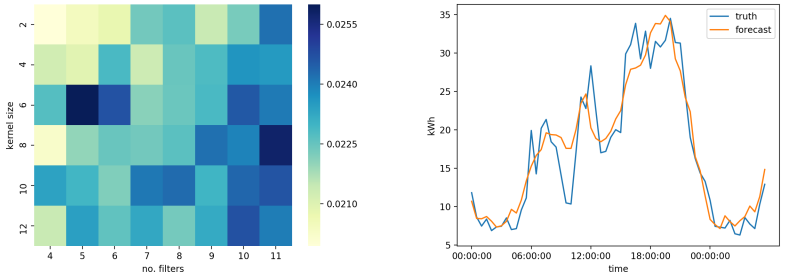
by one fully-connected layer which directly calculates the output. The evaluation of these networks yield the basic training parameters that are used throughout the rest of this work.

The best results were obtained with batch-size $b = 128$ and epochs $e = 40$. *Nadam* [14] is used as optimiser and the mean squared error (MSE) as loss-function. When using bigger batch-sizes, unwanted jumps in the training loss were observed regularly, and when using smaller batch-sizes, overfitting occured early during training.



(a) MSE dependent on the dense size and the number of filters.

(b) MSE dependent on the dense size and the kernel size



(c) MSE dependent on the kernel size and the number of filter.

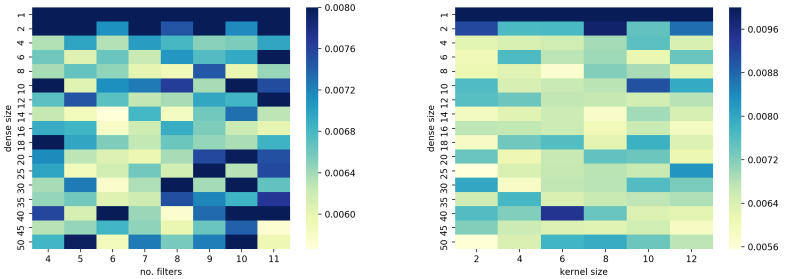
(d) An exemplary forecast of the 350 households load on the validation data.

Fig. 4: Heatmaps (a)-(c) show the MSE of the 40 household forecasts. The MSE values are a mean values across the third parameter. (d) shows a forecast using the trained CNN.

In the next trials, an additional fully-connected layer was added in between the convolutional layer and the fully-connected output layer (see network architecture in fig. 3). The additional fully-connected layer improved the forecast qualitiy independently from other network and training parameters. As this simple network already produces promising results, the convolutional layer is varied

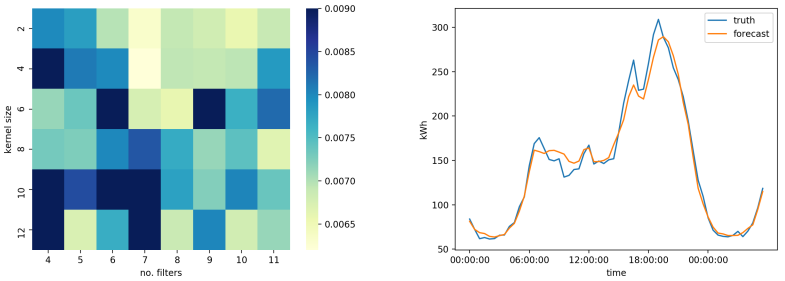
to further improve the forecasts.

The parameters that were varied are the kernel size and the number of filters. They describe how big the filters are, that sample over the time series, are and how many filters (each of them creates a feature map) are trained. The stride length is one. In addition, the influence of the first fully-connected layer is varied as well to identify how many features, from which the forecast is composed, exist.



(a) MSE dependent on the dense size and the number of filters.

(b) MSE dependent on the dense size and the kernel size



(c) MSE dependent on the kernel size and the number of filter.

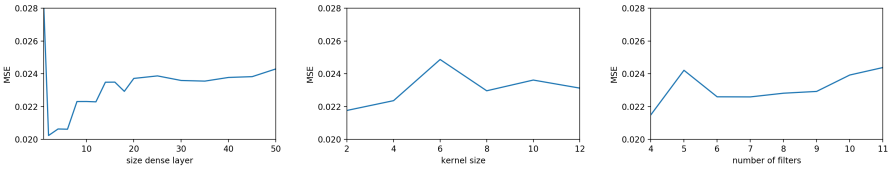
(d) An exemplary forecast of the 350 households load on the validation data.

Fig. 5: Heatmaps (a)-(c) show the MSE of the 350 household forecasts. The MSE values are a mean values across the third parameter. (d) shows a forecast using the trained CNN.

As is apparent from the heatmaps in figure 4, the four parameters have a crucial influence on the performance. On the heatmap plot (b), it can be seen that the best results can be achieved with a rather small fully-connected (also called dense) layer between the convolutional and the output layer. With an increasing dense size, the forecast results become unrelieable, probably overfitting occurs. In addition, the earlier conclusion that an additional fully-connected layer enhances

the forecast quality is confirmed by the significantly worse performance of the network when $dense\_size = 1$. This basically equals a network with only one fully-connected layer. The heatmap (a) indicates that a large fully-connected layer compensates a small number of filters and vice versa. However, when both parameters that are chosen are too big, the MSE increases. Due to the high amount of trainable parameters in the network that come with a large dense size, it is advisable to use a small fully-connected layer with a larger number of filters, in order to minimise the computational load. There is no obvious conclusion regarding the kernel size. It seems that a kernel size which is too big or too small has a negative influence on the forecast quality. That impression is supported by the average MSE across dense size and number of filters (see fig. 6 (a)-(c)).

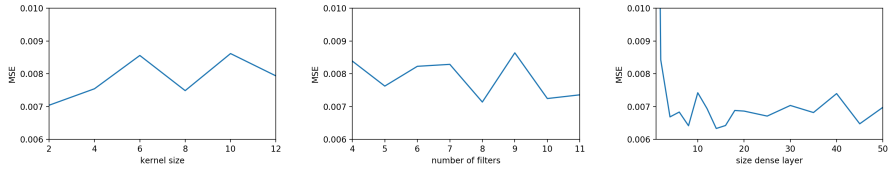
The heatmaps of MSE of the 350 household load are illustrated in figure 5. The influence of the network parameters differs from the 40 household load. In addition to the confirmation that the additional fully-connected layer increases the forecast quality, it also becomes apparent that only with more than two neurons in the fully-connected layer good forecasts are possible. Furthermore, it seems that the number of filters and the kernel size only have a minor influence on the MSE. On heatmap (c), however, it appears that the most accurate forecasts are the ones with smaller kernel sizes. Figure 6 (d)-(f) supports this assumption.



(a) MSE relative to dense size; 40 households.

(b) MSE relative to kernel size; 40 households.

(c) MSE realtive to number of filters; 40 households.



(d) MSE relative to dense size; 350 households.

(e) MSE relative to kernel size; 350 households.

(f) MSE relative to number of filters; 350 households.

Fig. 6: The MSE averaged across the other two parameters for the 40 household load in (a)-(c) and for the 350 household load in (d)-(f).

7 Conclusion

The network parameters dense size, number of filters, and kernel size were varied in a wide range. It can be conclude that the right set of parameters depends on the type of time series that is to be predicted.

The 350 household load time series can be forecasted properly with a CNN (see fig. 5 (d)). When the size of the fully-connected layer chosen is larger than two, the network is quite robust against changes in the number of filters and kernel size.

A relieable forecast of the load time series of 40 households is possible with a rather simple CNN when the parameters are chosen correctly (see fig. 4 (d)). It appears the time series can be described properly with 4 to 6 features as the best results were obtained with $dense\_size = 4...6$. Furthermore, it became apparent that with too many training parameters the forecast quality decreases, probably due to overfitting.

The forecasters for both time series can already outperform the standard load profile, even though the network architecture is quite simple and no external factors have been taken into account yet. For volatile load profiles, simplicity in the network architecture seems to be the key for good forecasting results.

8 Acknowledgments

We thank the Federal Ministry for Economic Affairs and Energy (BMWi) for funding the project MAGGIE.

References

1. V. Masson-Delmotte, P. Zhai, H. O. Pörtner, D. Roberts, J. Skea, P.R. Shukla, A. Pirani, W. Moufouma-Okia, C. Pan, R. Pidcock, S. Connors, J. B. R. Matthews, Y. Chen, X. Zhou, M. I. Gomis, E. Lonnoy, T. Maycock, M. Tignor, T. Waterfield (eds.): IPCC, 2018: Global warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty
2. Field, C.B., V. Barros, T.F. Stocker, D. Qin, D.J. Dokken, K.L. Ebi, M.D. Mastrandrea, K.J. Mach, G.-K. Plattner, S.K. Allen, M. Tignor, and P.M. Midgley (eds.): IPPC, 2012: Managing the Risks of Extreme Events and Disasters to Advance Climate Change Adaptation
3. Bundesministerium für Wirtschaft und Energie, Energiewende bauen, Solares Bauen: MAGGIE. https://projektinfos.energiewendebauen.de/projekt/energetisch-modernisieren-mit-solaraktiven-baustoffen-und-hybridem-heizsystem/
4. Projektträger Jüllich, EnArgus, Solares Bauen: MAGGIE.https://www.enargus.de/pub/bscw.cgi/?op=enargus.eps2&q=%2201180590/1%22&v=10&id=539378
5. M. Sterner, I. Stadler: Energiespeicher - Bedarf, Technologien, Integration. 2014 Springer Berlin Heidelberg.

6. Directive 2006/32/EC of the European Parliament and of the Council of 5 April 2006 on energy end-use efficiency and energy services and repealing Council Directive 93/76/EEC. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32006L0032
7. Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 concerning common rules for the internal market in electricity and repealing Directive 2003/54/EC. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32009L0072
8. Commission for Energy Regulation (CER). (2012). CER Smart Metering Project - Electricity Customer Behaviour Trial, 2009-2010 [dataset]. 1st Edition. Irish Social Science Data Archive. SN: 0012-00. www.ucd.ie/issda/CER-electricity
9. A. K. Srivastava, A. S. Pandey, D. Singh: Short-term load forecasting methods: A review. 2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), Sultanpur, 2016, pp. 130-138. doi: 10.1109/ICETEESES.2016.7581373
10. B. Hayes, J. Gruber,M. Prodanovic: Short-Term Load Forecasting at the local level using smart meter data. 2015 IEEE Eindhoven PowerTech, Eindhoven, 2015, pp. 1-6. doi: 10.1109/PTC.2015.7232358
11. Y LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition. Neural Computation, 1(4):541-551, Winter 1989.
12. N. Aloysius, M. Geetha: A review on deep convolutional neural networks. 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2017, pp. 0588-0592. doi: 10.1109/ICCSP.2017.8286426
13. J.T. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller: Striving for Simplicity: The All Convolutional Net. ICLR (workshop track), 2015.
14. T. Dozat: Incorporating Nesterov Momentum into Adam. 2016 ICLR 2016 workshop submission

A study of variable importance in multiclass classification problems based on the Volume Under the Surface measure

Ismael Ahrazem Dfuf, José Manuel Mira McWilliams and
Mª Camino González Fernández
Universidad Politécnica de Madrid (Spain)

Abstract

When dealing with more than two classes, classification problems become more complex, and hence the variable importance analysis. In this article, we present a comparative study of variable importance analysis in the multiclass case. The applied methodology uses the Conditional Inference Tree algorithm (CIT) (as it perfoms well in pronounced imbalanced data sets) along with the permutation importance technique. The comparison consists of evaluating the methodology using different performance measures: The Volume Under the Surface (VUS) and its computation alternatives, the G-Mean measure and the generalization of the Are Under the Curve (AUC). Several simulated scenarios characterized by their linearity and multiclass imbalance levels are built and tested.

A machine learning-based approach to forecasting alcoholic relapses

Nikola Katardjiev, Steve McKeever, and Andreas Hamfelt

Department of Informatics and Media, Uppsala University,
Kyrkogårdsgatan 10, 753 13 Uppsala, Sweden
{nikola.katardjiev,steve.mckeever,andreas.hamfelt}@uu.im.se
https://www.im.uu.se

Abstract. This research aims to explore alcoholic relapses by modelling four types of machine learning algorithms on clinical trial data of patients in an alcohol addiction treatment plan provided by an Uppsala-based company called Kontigo Care, with the goal of predicting downwards trends in the data that could indicate a relapse. The learning algorithms were support vector machine, random forest, decision trees, and a K-nearest neighbour regressor. Results indicated that using a random forest model, it is possible to forecast future data entries in the particular data set by beating the benchmark of a baseline predictor, suggesting that alcoholism could indeed be predicted, and thus future work leveraging larger data sets and deep learning methods could improve relapse predictions even further.

Keywords: Alcohol abuse, relapse, machine learning, time series forecasting.

1 Introduction

Irresponsible alcohol consumption is a wide-spread issue plaguing most modern countries; alcoholism is the third-leading cause of death in Russia [1], and it contributes to over 100,000 deaths in the United States alone [2]. Indeed, such is its effect that several countries, such as Sweden and Finland, have taken action to downsize and restrict the consumption of alcohol by the local population through restricting the sale to government-owned stores, Systembolaget and Alko, respectively [3].

However, individuals who try to restrict their alcohol consumption voluntarily often fail to do so [4], putting them at risk for a relapse following any temporary restriction they may have made. For those in recovery programmes and healthcare plans, a relapse following a substantial investment on the part of the caretaker can only be considered a failure and a sunk investment; it is estimated that relapse rates are usually found at around 60%, going as high as 90% in some cases [5]. Current research in addictive sciences is advising stronger patient monitoring and long-term treatment plans [6], which are argued to aid in

reducing relapse rates [7]. Solutions for monitoring these types of plans, however, are oftentimes rooted in more traditional techniques [8].

We seek to assess the efficacy of a machine learning approach to this domain. While healthcare at large has seen successfully applications of machine learning [9], the domain of addiction care has not been investigated thoroughly [10]. Given that machine learning has been applied to forecast other types of behaviours and trend changes, it follows that a careful application of those routines and appropriate transformations (given appropriate input data that corresponds to patient behaviour) should make it possible to forecast trend changes in alcohol abuse behaviour.

It is not unreasonable to then assume that there is a substantial knowledge gap in the development of heuristics for machine learning applications in a specialised field such as that of alcohol abuse treatment. This research aims to provide a proof-of-concept by modeling several common machine learning models against monitoring data of patients suffering from alcohol addiction. Due to the exploratory nature of our work, the data selection is taken from a single source, an Uppsala-based eHealth company called Kontigo Care AB [11].

The paper is structured as follows. In the next section we discuss past work in artificial intelligence being used in addiction monitoring and patient treatment. We then show our proposed implementations and explain the nature of the data set used in this research, before discussing the test results. Finally, we conclude and suggest various avenues for future work.

2 Related work

Only a handful of papers appear to have tackled the issue, such as the case of [8], which developed a classification-based approach to identifying relapses during ongoing treatment plans. They compared two models, a decision tree classifier and a Bayesian network, on 73 in-treatment patients, mapping outcomes as either successful or relapsing, with both models yielding a success rate higher than 50%. However, the research only modelled whether or not the patients would relapse following a completed treatment, whereas our research attempts to model relapses in ongoing treatment plans.

Modern research concerning time series forecasting can primarily be found in the field of machine learning; a paper by [12] details the development of an LSTM network capable of not only forecasting future time steps, but also performing anomaly detection on that time series. In other words, it can forecast a series and anticipate when aberrant behaviour will occur x time steps ahead.

Similarly, [13] used a recurrent ARIMA-like neural network to forecast several types of time series, ranging from Turkish stock prices to Australian beer consumption, generating a non-linear auto-regressive function as its output. This type of hybrid network has been developed in previous examples as well; [14] tested a case using a feed-forward neural network and an ARIMA model in tandem to forecast stock markets with results surpassing either independently.

There is evidence to suggest that hybrid ensemble models such as these can have improved performance on time series forecasting.

In the area of healthcare, machine learning can indeed be used to make predictions on patient data. Previous experiments in the domain have been used to predict myocardial infarctions [9], and machine learning was used to track the progression of Alzheimer's disease [15].

These cases are some strong samples of how machine learning is currently used to model predictions for time series forecasting, and we propose testing similar approaches in the domain of relapse forecasting.

3 Training and testing data

In 2018, a Sweden-based eHealth company called Kontigo Care published the results of a longitudal set of clinical trials done on 30 patients suffering from alcohol addiction [11]. The study was conducted over the course of an entire year, with the objective of monitoring, several times per day, alcohol levels in these patients, and this data forms the basis of our research. The clinical trial was undertaken in conjunction with Kontigo Care's Previct Alcohol product, a breathalyzer tool that patients can use to self-report their own behaviours, up to four times per day, each usage resulting in a unique measurement.

The breathalyzer results are calculated into a construct called Addiction Monitoring Index (AMI), an exponentially smoothed value scaling from 0 to 100 which acts as a metric for how well a patient is performing. An average AMI of 0 means that a patient has entered a full alcoholic relapse, while an AMI of 100 suggests that the patient has not missed any measurements, and each breathalyzer test has shown no traces of alcohol. The original research, from which the data stems, suggested that AMI is a valid method "for monitoring the recovery process and for early identification of lapse/relapse patterns" [11]. An additional study undertaken by the same company was conducted for a digital biomarker called Maximum Time Between Tests (MTBT), which is simply the difference in time between any one test and the previous. 54 patients were self-reporting using the same breathalyzer tool as in the previous clinical trial. The results showed that there was a correlation between MTBT and phosphatidyl ethanol (PEth) in the blood. The authors report the following: "The time-based digital biomarker maximum time between tests described here has the potential to become a generally useful metric for all scheduled measurement-based eHealth systems to monitor test behaviour and compliance, factors important for dosing of eHealth systems and for early prediction and interventions of lapse/relapse." [16]

MTBT is an indicator of a particular quirk of the dataset; a missed measurement (i.e., a null value) has some meaning with regards to how AMI is calculated. If a patient is expected to perform four tests, and one of them is skipped or otherwise not executed, it is nigh equivalent to having failed one breathalyzer test. In a sense, a null value does, in this dataset, still contain some information. While there is no guarantee that a patient has been drinking during the missed measurement, it is still a useful indicator that they could have done so.

Table 1: Example data of a fictive patient. AMI, on the far right, is the label, while all other columns sans "Entry" - are features.

Entry	**BAC**	**MTBT**	**PEth**	**AMI**
13	0.04	9.669	0.025	97.3
14	0.002	12.956	0.03	99.5
15	0.001	19.643	0.025	97.0
16	0.08	12.102	0.05	95.4
17	0.02	21.745	0.25	98.2
18	0.01	11.243	0.01	99.3
19	0.005	20.418	0.01	100.0

While MTBT can be a useful indicator of alcohol intake at a given moment, the aforementioned AMI is composed of several other key components. Phosphatidyl ethanol levels in the blood (PEth), blood alcohol content in permille reported in a measurement (BAC), and AMI in the previous measurements are all used in the calculation of the next measurement's AMI. For the purposes of this experiment, MTBT, PEth, BAC, and the previous AMI can be considered dataset features, while the next AMI measurement is the label.

Table 2: Descriptive statistics of relevant columns in the data set

	BAC	**MTBT**	**PEth**	**AMI**
Count	13959	13959	13959	13959
Mean	0.035918	0.0042	0.03	62.29
Std	330.69	0.0773	3.230	32.30
Min	0.00	0.000	0.000	0.012
25%	0.02	13.000	0.000	39.34
50%	0.06	15.000	0.000	72.51
75%	0.01	33.000	0.000	90.28
Max	4.822	3648	4.100	100.0

See Table 1 for an example of how the data can be structured. Note that the data is fictional, and does not represent a real patient in the clinical trials. The data set used for this research consists of the original research data for the clinical trials of 30 patients. The descriptive statistics of MTBT, AMI, PEth, and BAC are illustrated in Table 2.

Figure 1 shows the AMI curve of a single patient across approximately 200 days treatment. AMI is an exponentially smoothed value, meaning as it approaches its upper and lower bounds, it becomes increasingly difficult to 'stay' in those values; a missed measurement at a high value with cause a big drop, for example. In this research, AMI can be considered an output value ranging from 0 to 100, with a maximum change between any two time steps of 21%. Figure 1 provides an example of how AMI is smoothed towards the upper bounds.

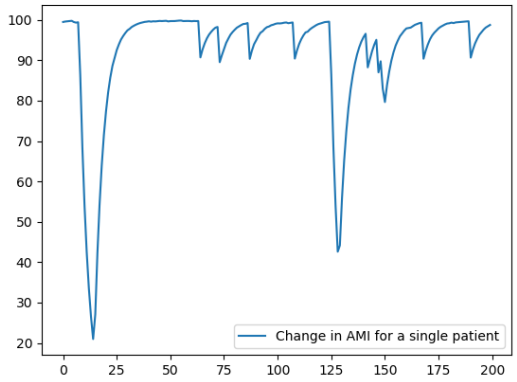


Fig. 1: Exponential smoothing in AMI of a single patient

The fact that the data is exponentially smoothed is particularly relevant, as maintaining a high AMI becomes progressively harder the closer one is to 100; in order to maintain that level, all breathalyzer tests must pass and none may be missed. Similarly, in order to stay at 0, a patient must be constantly failing or skipping all their tests. Hence, if the behaviour of patients was entirely random, the distribution of AMI would be expected to be primarily in the middle values, with 0 and 100 being particularly rare. A histogram of AMI values, Figure 2, shows that this is not the case:

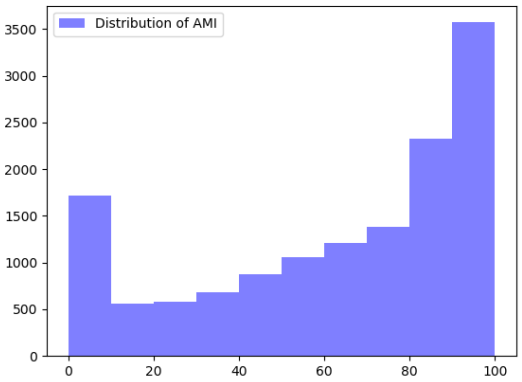


Fig. 2: A histogram showing the spread of AMI across the patients

Indeed, the histogram reveals that 0 and 100 are the most common values in the data set. This is a strong indicator that once patients fall into a lapse/relapse pattern, they tend to stay there for long periods of time. Similarly, a patient who is performing very well is likely to continue performing well. Hence, it becomes increasingly important to attempt to forecast increased relapse risks, as previously well-performing patients could enter a lapse/relapse pattern and subsequently stay in that pattern for a substantial time.

The model fitting will not be done on the data set only as is; a relapse is not determined only by whether or not a person has been consuming alcohol in the last measurement, but rather by a longer time sequence. It stands to reason, that past measurements can serve as indicators of a relapse occurring. As an abstract example, whether or not a day is a weekend could act as an indicator of cyclical behaviour; patients who maintain working lives or other weekly responsibilities may be consuming alcohol only on weekends, and by only considering the previous measurement in the data, such an occurrence would be missed. As expressed in To accommodate this, the data set is time shifted an arbitrary amount of steps in the past, like so: $y'_t = prediction(y_{t-3}, y_{t-4}, ..., y_{t-n})$.
Note that y'_t is the predicted AMI 3 days in advance. The data is time lagged to minimise the impact of the limited change in AMI from time step to time step; as the data is exponentially smoothed, any prediction of y'_t given y_{t-1} is flawed, as whatever number is provided, it can only change by at most a few percentage points, so any prediction within that range is decent. By lagging the series with two additional days, the amount of possible outcomes increases drastically.

For our research, in order to account for patient history, the usage of time lag and a time window is employed; the experiment is rendered once with a time lag of 30 and a window of 5 (in other words, providing inputs for days 1-30 to predict the output of day 35). The window needs to be sufficiently large such that exponential smoothing does not become a relevant factor in predictions; given that the smoothing factor limits how much the data can change day-by day, so by increasing the window we reduce the amount of variance that can be explained by the smoothing. Based on this, if the window was set to only 1, well over 90% of the variance could be explained by exponential smoothing alone.

By extending the time lag to a state where the entire range of 0-100 is included, it becomes significantly more difficult to make accurate predictions by repeating the previous measurement's AMI with some random noise added.

The models will be fitted and trained on this transformed data set. We increase the range to overlap the past month of measurements; this causes significant overlap between entries; much of the data between entries y_t and y_{t-1} will simply be repeated data. We argue that this redundancy of data is necessary to capture the behaviour of the patients in a representative fashion.

4 Implementation and approach

This research serves as a proof-of-concept that alcohol abuse patterns are to some degree predictable several days in advance, thereby making them preventable.

The objective is to fit a data set that in some way models alcohol usage over time onto machine learning models, and to show that it is possible to outperform baseline predictors. We compare four common machine learning algorithms – decision trees, random forests, support vector machines, and K-nearest neighbours – against a baseline predictor and compare highest degree of explained variance.

4.1 Model development, training process, and data collection

Each model was developed and tested independently, but the process of data collection (having developed and ensured the stability of each respective algorithm) followed the same three-step approach:

1. Train the algorithm using the split training data.
2. Test the resulting model with respect to the unseen test data.
3. Perform optimisations and validations to examine algorithm performance.

In this case, data collection is performed through the form of the *root mean squared error* (RMSE) and the *explained variance* (r^2 of the model predictions against real observations). Data collection is also undertaken by examining predicted trend changes in each model to examine their capabilities as forecasting tools. As the data set is rather small (13,959 entries, as shown later on), we use an 80/20 split of training/test data, using K-fold cross validation to ignore the use of validation data.

In order to provide baseline comparisons, which add relevancy to the models rating, we include a baseline predictor that simply returns the last known value in the time series; we also test a predictor that outputs the mean difference in AMI. This is a simple yet effective strategy that has seen usage in time series forecasting [17]. The mean difference is calculated only from the same training sets that the machine learning models are exposed to; this is to ensure that the testing data remains completely unseen. We define the baseline predictor as such: $y'_t = y_{t-5} + \textit{mean difference}\ (\textit{training data})$.
The mean difference is calculated exclusively from the training data, so as not to give it an unfair advantage on the testing set. It is assumed that a model that can outperform the baseline predictor should contain some power in predicting alcoholic relapses in the modeled data.

4.2 Algorithms

We use a single *decision tree* as our first tested model. They have a high degree of transparency, and if the data can be modeled and predicted in a simple enough manner, it may be sufficient to use a single tree for our purposes. Conversely, if the problem cannot be modeled simply enough, we expect the decision tree to overfit onto the training set. The second predictor is a *random forest* [18]. Studies [19] appear to indicate that they show more promise in solving regression problems with higher levels of accuracy. We argue that our inclusion of a random forest is predicated on the case that decision trees do not have sufficient

explanative power to cover a problem on this nature. Finally, we use a *support vector machine* (SVM) fitted with a *radial basis function* (RBF) kernel to model the data in the same manner as the previous two models, as well as a K-nearest neighbour regressor. This is to help show if models perform differently on the data set, helping suggest that they are not forecasting the same thing.

The models were trained on 80% of the samples, selected semi-randomly; for any given patient, all entries of that patient needed to be in either the training set or the testing set. Splitting data from the same patients into both data sets would give the models an unfair advantage on predicting on patients it's already been trained on, which would cause the test set to overlap with the training set and make it less reusable for new patients. If sample Y_t is found in the training set, it does not make sense to add Y_{t-1} to the test set, as a significant amount of data is repeated, only in different columns.

5 Results

Once trained on the appropriate data, the models were tasked with making predictions on both the training and testing sets, in order to show not only how well they performed compared to each other, but also to see to what extent overfitting may have played a part in their performances, as shown in Table 3.

Table 3: Summary of algorithm performances. We show root mean square error of train and test data, as well as the explained variance for each model.

Model	**Train RMSE**	**Train Variance**	**Test RMSE**	**Test Variance**
Decision Tree	0.077	1.000	18.091	0.646
Random Forest	4.908	0.977	12.124	0.841
Support Vector regressor	23.344	0.489	28.183	0.142
K-nearest Neighbour regressor	12.780	0.847	14.237	0.781
Baseline Predictor	16.529	0.744	15.062	0.755

The baseline predictor is a relatively accurate indication of what extent exponential smoothing (as shown in Figure 1) can be used to explain the variance in the data set. Looking at the results, the baseline predictor can explain 75.5% of the variance in the data. It is not unreasonable, then, to suggest that exponential smoothing can explain most of that variance. Hence, we argue that a model that can explain more of the variance must be forecasting better than the baseline predictor.

We find that a majority of the models perform extremely well on the training set; in three out of four cases, the models are able to explain over 80% of the variance. Of course, this is on the training set, so we expect an extremely high correlation here. Indeed, a single decision tree can explain 100% of the variance

in the training set. An impressive number, but once exposed to previously unseen data in the testing set, the decision tree can only explain 64.6% of the variance: a clear warning sign of overfitting.

In general, all models saw increased RMSE values and decreased r^2-values as predictions were moved onto the test sets, as one would expect, given that they have been exposed to training sets previously. It is notable that the support vector regressor performed extremely poorly on the test set, explaining only 14.2% of the variance in the data set. As illustrated earlier, simply using exponential smoothing can explain significantly more of the variance.

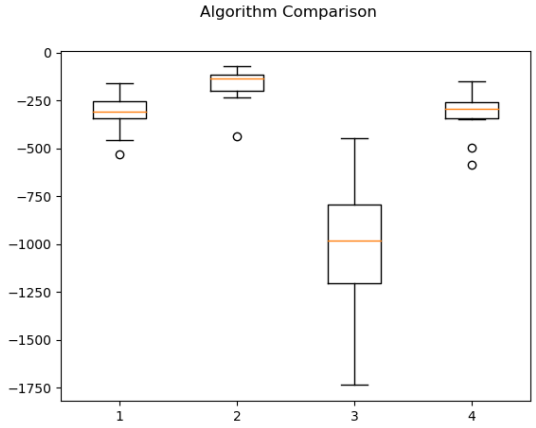


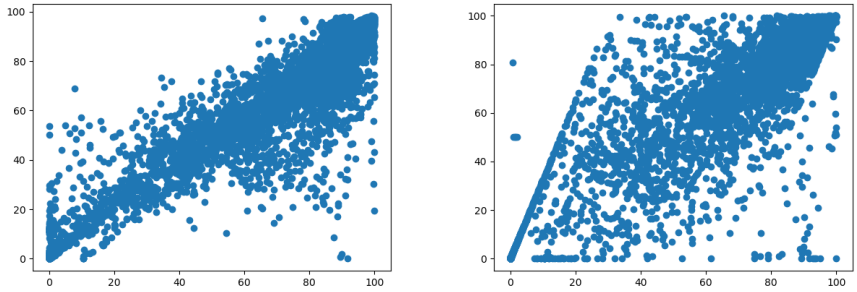
Fig. 3: A box-whisker plot comparing the four different machine learning models on negative square mean error. From left to right: decision tree, random forest, support vector regressor, and K-nearest neighbour.

In Figure 3 we find a much larger spread in predictions in one model than the other on the testing set; the support vector machine had a range of prediction errors from 22.5 AMI points to 42.4; this contrasts with the decision tree, that, despite overfitting severely onto the training data, produced its largest erroneous prediction at 22.3 AMI points off, while its smallest error was in the single digit.

Comparing Figure 3 and Table 3, the random algorithm performed the best, dropping only a relatively small amount in predictive power when moving onto the test set, being able to explain 84.1% of the variance in the data, and with the lowest RMSE recorded on the test set. Similarly, the box-whisker plot shows how reliable its predictions are, with only a single outlier, and very low spread in the errors of its predictions. However, it should be noted that the random forest is only a somewhat better predictor than the K-nearest neighbour regressor, which also outperforms the baseline predictor, with both lower RMSE and more variance explained (14.237 and 0.781 compared to 15.062 and 0.755, respectively).

6 Analysis

While the K-nearest neighbour model does boast better statistics than the baseline predictor, the difference is rather miniscule, and could be tallied up to being purely coincidental. Decision trees are very prone to overfitting to the training data for a problem of this nature, thus they can easily be dismissed. We suspect the support vector regressor performed so poorly due to the difficulty in optimising them. As the results clearly indicated, the random forest model outperformed our baseline predictor, and to better illustrate how its predictions compared to the baseline, Figure 4a and 4b plot predicted and observed AMI-values for the two models, respectively.



(a) For a random forest algorithm (b) For a baseline predictor

Fig. 4: A scatterplot showing the spread of predicted AMI on the X-axis against observed AMI on the Y-axis.

The closer the data points are arranged in a straight line along the axis $y = x$, the more accurate the model predictions are. In Figure 4a, we can see that the data is approximately arranged in a line, though there appears to be a case of excessive warning in the high end of the spectrum; that is, the model predicts low values, indicating relapses where there were none in actuality, so the model is predicting a *false positive*. The same but in reverse happens on the low end of the spectrum. This is particularly disturbing, as it is indicative of the model suggesting that a patient will do well, when, actually, a relapse has occurred; a *false negative* has occurred.

However, as Figure 4b shows, the baseline predictor's forecasted AMI-values have a significantly larger spread; it does not warn in case of relapses nearly enough, and misses a significant amount. This is clearly visible to the naked eye; the aforementioned target, $y = x$, is noticeably less present. We also see a skewed 'grid'-like feature in the data set; this is because the majority of values are oftentimes just the same value shifted some days forward or backwards. As

it was shown earlier that the baseline predictor was more likely to perform both false positives and false negatives, it can be argued that the random forest was predicting more accurately.

7 Conclusion

Our findings suggest that machine learning models can make relatively meaningful predictions on alcoholic relapses five days in advance, as they can beat baseline predictors, both in terms of RMSE and explained variance. Crucially, one model was able to outperform a mean difference predictor. There is a gap in the domain-specific applications of machine learning in this field, and this paper serves as a proof-of-concept that shows that simple, commonplace models can indeed be used to forecast relapses in alcohol abuse. Hence, we argue that further research in this area is highly advised.

There are some caveats that must be highlighted. The study lacks a certain level of generalizability, as the methodology for this paper was built on a single company's own construct for measuring sobriety. This implies that the approach taken is not necessarily generalizable to other data sets dealing with the same domain. In order to examine the subject in a more general sense, one would have to incorporate aspects touched upon by [8], requiring transformations of output variables. Hence, this research can only serve as a proof-of-concept that relapse forecasting is indeed possible, given sufficient data.

There were other factors neglected during this research. Due to availability concerns, only data pertaining to the measurements completed in the clinical trials performed by Kontigo Care [11] were used, but there are other potential indicators of substance abuse. As mentioned in the data set discussion, weekdays, whether or not a salary day has passed, seasons, or vacation periods could be used to make more accurate predictions.

The reason why more modern techniques, such as LSTM networks [12], were not tried is because of the limited data set size. Neural networks and deep learning demand extremely large quantities of data, well beyond the scope of the initial clinical study. However, as this paper shows, relapse forecasting is a possibility, and given the costs associated with treatment plans and eventual relapses [2], further research is beneficial. With larger data sets, deep learning could be used to make extremely accurate long- and short-term predictions alike.

References

1. Baltagi, B.H., Geishecker, I.: Rational Alcohol Addiction: Evidence from the Russian Longitudinal Monitoring Survey. Health Economics **15**(9) (2006) 893 – 914
2. Miller, N.S., Gold, M.S.: Comorbid Cigarette and Alcohol Addiction. Journal of Addictive Diseases **17**(1) (1998) 55 – 66 PMID: 9549603.
3. Örnberg, J.C., Ólafsdóttir, H.: How to sell alcohol? nordic alcohol monopolies in a changing epoch. Nordic studies on alcohol and Drugs **25**(2) (2008) 129–153
4. Moos, R.H., Moos, B.S.: Rates and Predictors of Relapse after Natural and Treated Remission from Alcohol Use Disorders. Addiction **101**(2) (2006) 212–222

5. Stout, R.L., Rubin, A., Zwick, W., Zywiak, W., Bellino, L.: Optimizing the cost-effectiveness of alcohol treatment: A rationale for extended case monitoring. Addictive Behaviors **24**(1) (1999) 17–35
6. Thylstrup, B.: Numbers and Narratives. Relations between Patient Satisfaction, Retention, Outcome and Program Factors in Outpatient Substance Abuse Treatment. Nordic Studies on Alcohol and Drugs **28**(5-6) (2011) 471 – 486
7. McLellan, A.T., McKay, J.R., Forman, R., Cacciola, J., Kemp, J.: Reconsidering the Evaluation of Addiction Treatment: from Retrospective Follow-up to Concurrent Recovery Monitoring. Addiction **100**(4) (2005) 447 – 458
8. Connor, J.P., Symons, M., Feeney, G.F.X., Young, R.M., Wiles, J.: The Application of Machine Learning Techniques as an Adjunct to Clinical Decision Making in Alcohol Dependence Treatment. Substance Use and Misuse **42**(14) (2007) 2193–2206
9. Weiss, J.C., Natarajan, S., Peissig, P.L., McCarty, C.A., Page, D.: Machine learning for personalized medicine: Predicting primary myocardial infarction from electronic health records. AI Magazine **33**(4) (2012) 33
10. Cruz, J.A., Wishart, D.S.: Applications of Machine Learning in Cancer Prediction and Prognosis. Cancer Informatics **2** (2006) 59 – 78
11. Hämäläinen, M.D., Zetterström, A., Winkvist, M., Söderquist, M., Karlberg, E., Öhagen, P., Andersson, K., Nyberg, F.: Real-time monitoring using a breathalyzer-based ehealth system can identify lapse/relapse patterns in alcohol use disorder patients. Alcohol and Alcoholism **53**(4) (2018) 368–375
12. Lee, T.J., Gottschlich, J., Tatbul, N., Metcalf, E., Zdonik, S.: Greenhouse: A Zero-Positive Machine Learning System for Time-Series Anomaly Detection. (01 2018)
13. Akdeniz, E., Egrioglu, E., Bas, E., Yolcu, U.: An ARMA Type Pi-Sigma Artificial Neural Network for Nonlinear Time Series Forecasting. Journal of Artificial Intelligence and Soft Computing Research **8**(2) (2018) 121 – 132
14. Zhang, G.P.: Time Series Forecasting using a Hybrid ARIMA and Neural Network Model. Neurocomputing **50** (2003) 159 – 175
15. Xu, J., Deng, C., Gao, X., Shen, D., Huang, H.: Predicting alzheimers disease cognitive assessment via robust low-rank structured sparse model. **2017** (2017) 3880
16. Zetterström, A., Hämäläinen, M.D., Karlberg, E., Winkvist, M., Söderquist, M., Öhagen, P., Andersson, K., Nyberg, F.: Maximum Time Between Tests: A Digital Biomarker to Detect Therapy Compliance and Assess Schedule Quality in Measurement-Based eHealth Systems for Alcohol Use Disorder. Alcohol and Alcoholism **54**(1) (12 2018) 70–72
17. Kilian, L., Taylor, M.P.: Why is it so difficult to beat the random walk forecast of exchange rates? Journal of International Economics **60**(1) (2003) 85–107
18. Liaw, A., Wiener, M.: Classification and Regression by RandomForest. R News **2/3** (12 2002) 18 – 22
19. Segal, M.R.: Machine Learning Benchmarks and Random Forest Regression. UCSF: Center for Bioinformatics and Molecular Biostatistics (2004)

Improved Extreme Rainfall Events Forecasting Using Neural Networks and Water Vapor Measures

Matteo Sangiorgio[1], Stefano Barindelli[2], Riccardo Biondi[3], Enrico Solazzo[2], Eugenio Realini[4], Giovanna Venuti[2], and Giorgio Guariso[1]

[1] Department Electronics, Information, and Bioengineering, Politecnico di Milano, Milan, Italy
[2] Department of Civil and Environmental Engineering, Politecnico di Milano, Milan, Italy
[3] Department of Geosciences, Università degli Studi di Padova, Padova, Italy
[4] Geomatics Research & Development srl (GReD), Lomazzo (CO), Italy
matteo.sangiorgio, stefano.barindelli@polimi.it

Abstract. In the last few years, many studies claimed that machine learning tools would soon overperform the classical conceptual models in extreme rainfall events forecasting. In order to better investigate this statement, we implement advanced deep learning predictors, such as the deep neural nets, for the forecasting of the occurrence of extreme rainfalls. These predictors are proved to overperform more simple models such as the logistic regression, which are traditionally used as a benchmark for these tasks. Also, we evaluate the value of the information provided by the Zenith Tropospheric Delay. We show that adding this variable to the traditional meteorological data leads to an improvement of the model accuracy in the order of 3-4 %. We consider an area composed by the catchments of four rivers (Lambro, Seveso, Groane, and Olona) in the Lombardy region, northern Italy, just upstream from the metropolitan area of Milan, as a case study. Data of convective extreme rainfall events from 2010 up to 2017 (more than 600 extreme events) have been used to identify and test the predictors.

Keywords: Nowcasting, Extreme Rain Events, Deep Neural Networks, Global Navigation Satellite System, Zenith Tropospheric Delay.

1 Introduction

Many researchers in the field of meteorology claim that machine learning techniques will soon overperform the traditional physically based models in weather forecasting.

Also, black box models seem to be well suited for real-time application, since they are faster due to the lower computational effort required with respect to the traditional meteorological nowcasting methodologies, which are based on physically based models.

In particular, extreme events are very difficult to predict with classical Numerical Weather Prediction (NWP) models because they usually affect very small and local-

ized areas and the convection is triggered by peculiar and local conditions, requiring both high-resolution NWP and high temporal and spatial resolution observations.

In this work, we deal with the problem of forecasting the occurrence of extreme local rainfall events 30 minutes ahead.

The considered area, located in Lombardy region, Northern Italy, is composed by the hydrological basin of four torrential rivers (Lambro, Seveso, Groane, and Olona). This is a high-risk territory due to the high frequency of severe and short thunderstorms, which usually trigger flash floods. The situation is even more critical due to the presence of the metropolitan area of Milan, where the flows coming from the four considered rivers are drained, causing severe damage. In 2014, for instance, floods produced damages evaluated in several million euros in the Milan municipality.

In this work, we adopted advanced machine learning tools, the Deep Neural Networks (DNNs hereafter), which receive as input some meteorological variables sampled inside and around the study area and return as output the prediction about the occurrence of an extreme event.

In addition to the classical meteorological variables (temperature, pressure, humidity, wind speed), we also included the Zenith Tropospheric Delay (ZTD), which seems to be promising since it is a proxy of water vapor in the atmosphere, a fundamental variable in rain events genesis [1] [2] [3] [4].

This represents a novel element of this research since it is one of the first attempts to use the ZTD in a black box model for prediction of severe storms [5] [6]. We quantify the impact of ZTD repeating the task twice: the first time without considering ZTD, the second including it within the model inputs.

Developing a black box model for this environmental problem could become an innovative nowcasting product exploitable also by Civil Protection Agencies to face emergencies.

This work is part of the Lombardy based Advanced Meteorological Predictions and Observations (LAMPO) project (http://www.geolab.polimi.it/projects/lampo-project/).

2 Methods

2.1 Extreme Event Definition

The objective of this work is to identify machine learning models able to forecast the occurrence of extreme rainfall events 30 minutes ahead.

We consider a rainfall event as extreme if it persists for more than 25 minutes within the study area and if the radar reflectivity factor is greater than 50 dBZ.

2.2 Machine Learning Models

Since the model's output is a Boolean variable (occurrence of the extreme event), the task we are dealing with is a binary classification task.

As it is well-known, while developing machine learning tools, it is important to start with some simple models which will be considered as a benchmark for more

complex (and hopefully more performing) ones. In this case, we adopted a logistic regression (see Fig. 1) as a baseline model, using its Python implementation provided by Scikit-learn library [7].

The logistic regression is a linear classifier which splits the feature space (which in this case is a high-dimensional one) with a linear manifold and classifies each sample according to its position relative to a linear decision boundary.

Given the complexity of almost all the real-world applications, it is unlikely that the decision boundary is actually a linear one. For this reason, we introduced a more advanced machine learning model which can efficiently deal with problems where classes are not linearly separable: a DNN [8] (see Fig. 1).

The deep neural network here considered has a traditional fully connected structure [9] and has been implemented in Keras [10] with TensorFlow backend.

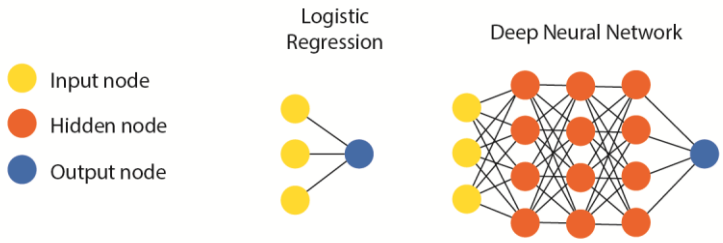


Fig. 1. Representation of the considered model's architectures.

To find the best combination of hyper-parameter (learning rate, batch size, regularization rate, activation functions shape, number of hidden layers, number of neurons for each layer, class weights) values, we implemented a traditional grid search approach.

The dataset used to identify the classifiers has been split into training (70 % of the samples), validation (15 %) and test (15 %) sets, as it is common practice in the neural network's identification procedure.

Since we are dealing with a classification task, we considered the binary cross-entropy as loss function and the overall classification accuracy as validation metrics.

Early stopping and L2 norm weight regularization have been used to avoid overfitting on training data. The performances, in terms of overall accuracy and confusion matrix, are then evaluated on the test set.

2.3 Meteorological Variables

Several classical meteorological variables are measured every 10 minutes: temperature, air pressure, wind speed, and relative humidity. In addition, another variable has been considered: the Global Navigation Satellite System (GNSS) derived ZTD estimated from the observations of the permanent geodetic station of Como. ZTD represents the zenithal delay in the transmission of the GNSS signal from the satellite to the ground receiver caused by the troposphere [11]. It is the sum of a delay caused by the troposphere gases in hydrostatic equilibrium, called Zenith Hydrostatic Delay

(ZHD) and a delay caused by the presence of water vapor called Zenith Wet Delay (ZWD). Since the temporal variations of the first term are very small, the ZTD could be considered a proxy of the presence of water vapor in the atmosphere [12], which is a fundamental variable in rain events genesis.

Each sample in the dataset is thus formed by an input vector, whose elements are the meteorological variables, and by an output value, a boolean variable which represents the occurrence (or not) of the rainfall extreme event.

The dataset considered in this work covers the period from 2010 to 2017 and contains 656 extreme events (together with thousands of cases where the extreme events did not occur).

3 Results

The baseline situation (i.e., using logistic regression with traditional meteorological variables only) guarantees an overall classification accuracy of 72.5 % corresponding confusion matrix is reported in Fig. 2.

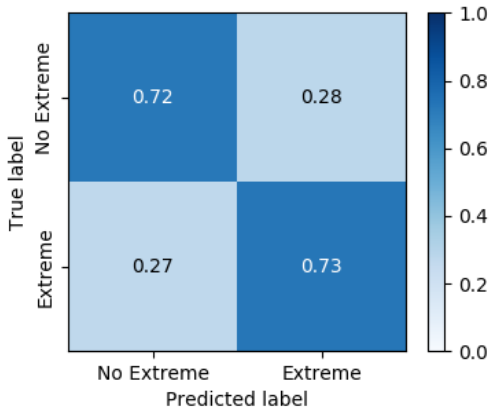


Fig. 2. Confusion matrix obtained with the logistic regression considering traditional meteorological variables only.

As already stated in the previous section, given the complexity and the nonlinear nature of the processes which occur in the atmosphere, it is very unlikely that a simple model such as the logistic regression would turn out to be the best approach to deal with the considered problem.

This idea is confirmed by the performances obtained with a more complex model: a DNN with three hidden layers, each one composed by ten neurons: the overall accuracy grows up to 79.0 % (see Fig. 3 for the confusion matrix).

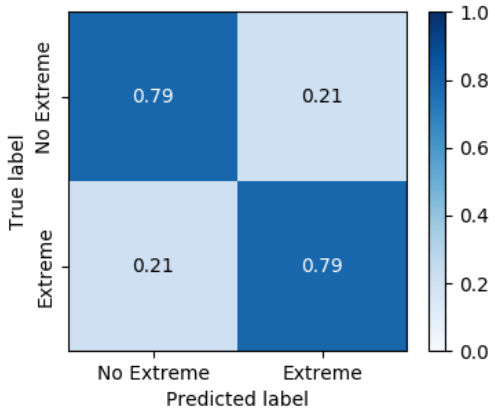


Fig. 3. Confusion matrix obtained with the DNN considering traditional meteorological variables only.

To evaluate the importance of including ZTD estimates, we repeated the identification of the two models with the new set of input variables.

Fig. 4 and 5 show the confusion matrices computed with the logistic regression and the DNN, respectively. Looking at the comparison between the models, the results exhibit almost the same trend when the ZTD is included or not in the inputs: adopting complex models like the DNNs, the overall accuracy in the forecasting of extreme events increases of 6.5 % and 8.5 % for the cases without and with the ZTD, respectively (see Table 1).

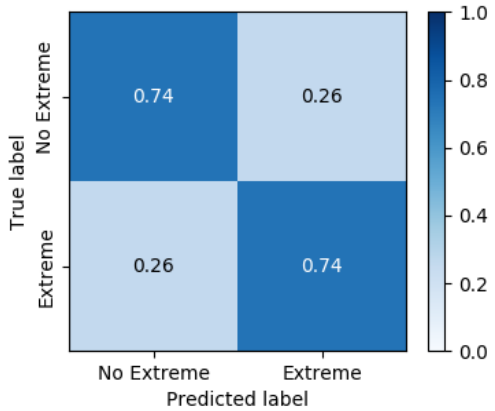


Fig. 4. Confusion matrix obtained with the logistic regression, including the ZTD in the input variable set.

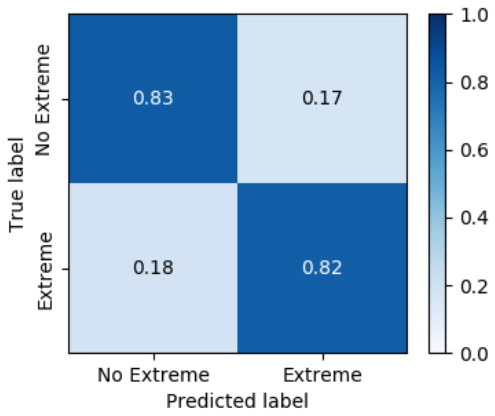


Fig. 5. Confusion matrix obtained with the DNN, including the ZTD in the input variable set.

The performances computed in terms of overall accuracy, which are reported in Table 1, allow quantifying the value of the information provided by the ZTD measured at Como. In fact, considering the logistic regression, including the ZTD within the input set increases the accuracy from 72.5 % to 74.0 % (+1.5 %). The advantage is even more evident when adopting a DNN: the overall accuracy grows from 79.0 % to 82.5 %.

Table 1. Overall accuracy of the models identified in the study.

Model	Overall accuracy
Logistic regression without ZTD	72.5 %
DNN without ZTD	79.0 %
Logistic regression with ZTD	74.0 %
DNN with ZTD	82.5 %

4 Conclusion

In this paper, we showed how machine learning techniques can be effectively used to forecast extreme rainfall events. In particular, the results demonstrate that complex nonlinear models, such as the DNNs, overperform the logistic regression, which has been used as a benchmark. For the considered case study, this advantage can be quantified in the range of 5-10 %.

In addition, we confirm the results recently obtained in [5] and [6]: including the ZTD in the input set leads to an increase of the model accuracy, especially when adopting a DNN, of the order of 3-4 %.

This fact seems interesting because the ZTD station, located in Como, is on the border of our study area. We would expect even better performances in case the station where ZTD is measured was localized closer to the center of the study area or if there were some stations inside and/or outside the considered boundary.

References

1. Barindelli, S., Realini, E., Venuti, G., Fermi, A., Gatti, A.: Detection of water vapor time variations associated with heavy rain in northern Italy by geodetic and low-cost GNSS recievers. Earth Planets Space 70, 28 (2018).
2. De Haan, S.: Assimilation of GNSS ZTD and radar radial velocity for the benefit of very-short-range regional weather forecasts. Q. J. R. Meteorol. Soc. 139, 2097-2107 (2013).
3. Dousa, J., Vaclavovic, P.: Real-time zenith tropospheric delays in support of numerical weather prediction applications. Adv. Space Res. 53, 1347-1358 (2014).
4. Benevides, P., Catalão, J., Miranda, P.M.A.: On the inclusion of GPS precipitable water vapour in the Nowcasting of rainfall. Nat. Hazards Earth Syst. Sci. 15, 2605-2616 (2015).
5. Benevides, P., Catalão, J., Nico, G., Miranda, P.: Evaluation of rainfall forecasts combining GNSS precipitable water vapor with ground and remote sensing meteorological variables in a neural network approach. In: Remote Sensing of Clouds and the Atmosphere XXIII. International Society for Optics and Photonics, p. 1078607 (2018).
6. Benevides, P., Catalao, J., Nico, G.: Neural Network Approach to Forecast Hourly Intense Rainfall Using GNSS Precipitable Water Vapor and Meteorological Sensors. Remote Sensing 11(8), 966 (2019).
7. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al: Scikit-learn: Machine Learning in Python. J Mach Learn Res (2012).
8. Bengio, Y.: Learning Deep Architectures for AI. Found Trends® Mach Learn. 2, 1–127 (2009).
9. Goodfellow, I., Bengio, Y., Courville, A.: Convolutional Networks. In: Thomas Dietterich, editor. Deep Learning. Cambridge, Massachusetts; London, England: MIT Press. 321–359 (2016).
10. Chollet, F.: Keras Documentation. (web: keras.io) (2015).
11. Kleijer, F.: Troposphere modeling and filtering for precise GPS leveling (2004).
12. Bevis, M., Businger, S., Herring, T.A., Rocken, C., Anthes, R.A., Ware, R.H.: GPS meteorology: Remote sensing of atmospheric water vapor using the Global Positioning System. Journal of Geophysical Research: Atmospheres, 97(D14), 15787-15801 (1992).

Statistical Approach to Predict Meteorological Material for Real-time GOCI Data Processing

Hyun Yang

Korea Institute of Ocean Science and Technology, 385, Haeyang-ro, Busan, Korea
yanghyun@kiost.ac.kr

Abstract. The Geostationary Ocean Color Imager (GOCI) can be utilized to analyze subtle changes on oceanic environments because it observes ocean colors around the Northeast Asia hourly, for 8 times a day. To realize this, the Korea Ocean Satellite Center (KOSC) which is the main operating agency of GOCI has a role to receive, process, and distribute its data within an hour. In this situation, we need several meteorological materials (e.g., ozone, wind, relative humidity, pressure, etc.) to successfully process the GOCI atmospheric corrections. Meteorological materials from National Aeronautics and Space Administration (NASA) Ocean Biology Processing Group (OBPG) are used when the GOCI atmospheric corrections are processed. Unfortunately, however, these materials cannot be used for the real-time GOCI data processing because they cannot be provided in real-time. In this paper, therefore, we propose a statistical approach for predicting the meteorological material and analyzed its accuracy.

Keywords: Statistical Approach, GOCI, Meteorological Material.

1 Introduction

The Geostationary Ocean Color Imager (GOCI) is the world's first ocean color sensor operated in geostationary orbit in order to observe ocean colors around Northeast Asia [1]. GOCI acquires ocean color scenes hourly, for 8 times a day to observe subtle changes on maritime environments [2].

GOCI Data Processing System (GDPS) employs reanalysis meteorological materials (e.g., ozone, wind, relative humidity, pressure, etc.) supported from National Aeronautics and Space Administration (NASA) Ocean Biology Processing Group (OBPG) to improve the accuracy of the atmospheric correction algorithm [3]. Unfortunately, however, these reanalysis materials are supported after 1-3 days later than the given date, although GOCI data must be received, processed, and distributed within an hour. In this paper, therefore, we proposed a statistical approach using the meteorological materials during past 5 years for supporting the estimated meteorological materials when GOCI data are processed in real time.

To validate the accuracy of the proposed approach, we compared the estimated meteorological materials to the reanalysis ones for a target area which located in the middle of East/Japan Sea, shown in **Fig. 1**. Also, we compared the atmospheric cor-

rection results derived from the estimated meteorological materials to those derived from the reanalysis meteorological materials.

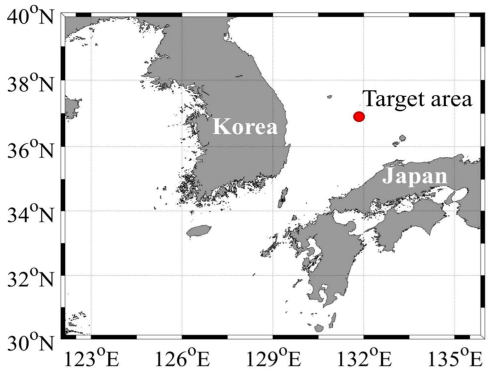


Fig. 1. Target area for the comparative analysis

2 Data

In this study, we used 5 kinds of reanalysis meteorological materials: ozone, zonal wind, meridional wind, relative humidity, and pressure during 5 years from 2013 to 2017. We obtained these data from the NASA OBPG website [4]. Also, we obtained the GOCI data from Korea Ocean Satellite Center (KOSC) to compare the atmospheric correction results derived from the estimated meteorological materials to those derived from the reanalysis meteorological materials, in terms of 2018.

3 Method and Results

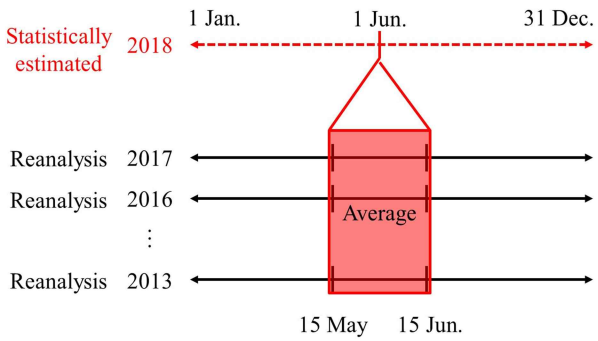


Fig. 2. A schematic diagram for the proposed method.

A schematic diagram for the proposed approach is shown in **Fig. 2**. This methodology calculates the average of the 30-days reanalysis meteorological materials for the past

5 years for estimating the meteorological materials on a given date. For example, an average of reanalysis meteorological materials from 15 May to 15 Jun. for each year from 2013 to 2017 is calculated for estimating the meteorological materials on 1 Jun. 2018.

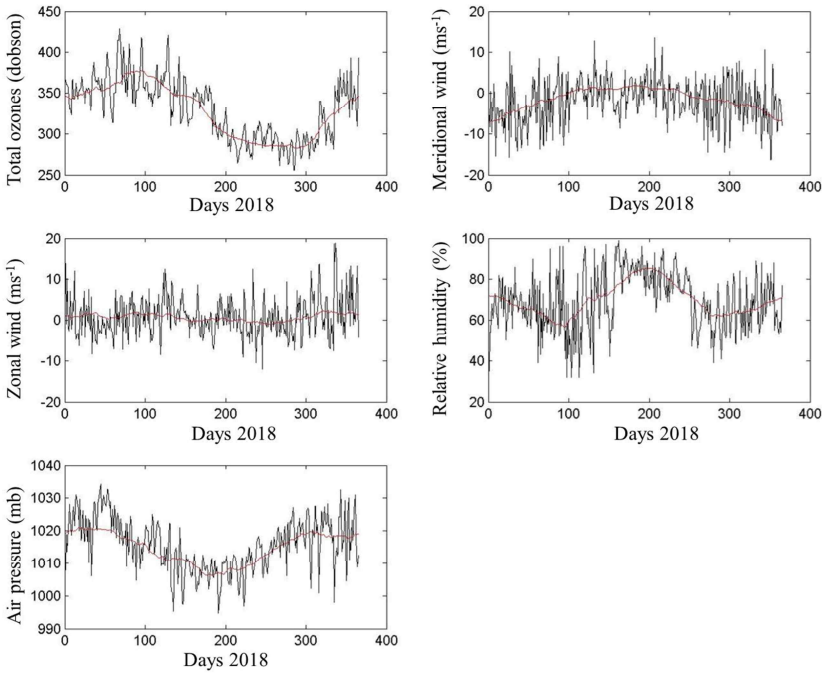


Fig. 3. Time series of the estimated (red) and the reanalysis (black) meteorological materials

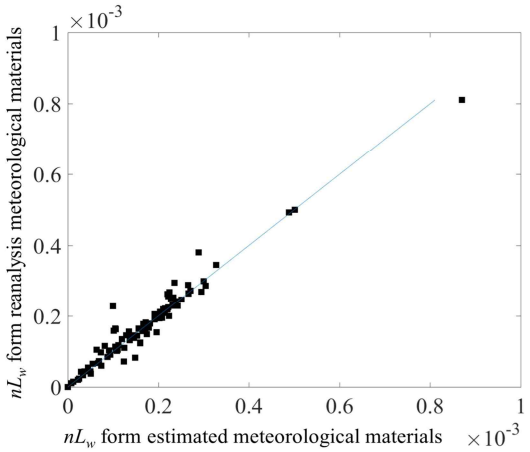


Fig. 4. A comparison of nL_w values from estimated and reanalysis meteorological materials

Fig. 3 shows the results for the time series analysis of the estimated and the reanalysis meteorological materials. Red lines indicate the estimated meteorological materials and black lines indicate the reanalysis meteorological materials. We can find out that the time series of the estimated and reanalysis materials tend to be similar each other.

Fig. 4 shows a scatter diagram for comparing the normalized water-leaving radiance (nL_w) values from the estimated and the reanalysis meteorological materials. nL_w is a product derived from GOCI atmospheric correction algorithm, It is shown that the nL_w values from the estimated and the reanalysis meteorological materials are distributed similarly.

4 Conclusion

In this study, we proposed a statistic methodology for estimating the meteorological materials on the given date using the materials of past 5 years in order to support the GOCI atmospheric correction processing in real time. In several experimental results, it was validated that the proposed approach is feasible because seasonal tendencies are derived from the estimated meteorological materials. The atmospheric correction results derived from the estimated meteorological materials were also available. For the future work, we will compare the estimated materials to the in situ data to verify its accuracy.

Acknowledgements

This work has supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2018R1D1A1B07049261). This work has also supported by the 'Technology development for Practical Applications of Multi-Satellite data to maritime issues' and 'Development of the integrated data processing system for GOCI-II' funded by the Ministry of Ocean and Fisheries, Korea.

References

1. Ryu, J. H., Han, H. J., Cho, S., Park, Y. J. and Ahn, Y. H.: Overview of Geostationary Ocean Color Imager (GOCI) and GOCI Data Processing System (GDPS). Ocean Science Journal. 47(3), 223-233 (2012).
2. Yang, H., Choi, J. K., Park, Y. J., Han, H. J., and Ryu, J. H.: Application of the Geostationary Ocean Color Imager (GOCI) to estimates of ocean surface currents, Journal of Geophysical Research: Oceans, 119(6), 3988-4000 (2014).
3. Yang, H., Yoon, S., Han, H. J., Heo, J. M., and Park, Y. J.: Data Processing System for the Geostationary Ocean Color Imager (GOCI), KIISE Transactions on Computing Practices, 23(1), 74-79 (2017).
4. NASA OBPG Homepage, (https:// oceancolor.gsfc.nasa.gov), last accessed 2019/06/21.

New Technique for Risk Measurement: Beyond Conventional Methods

Maryam Zamani[1], Ali Namaki[2], Gholamreza Jafari[3], and Holger Kantz[1]

[1] Max Planck Institute for the Physics of Complex Systems, Dresden, Germany, zamani@pks.mpg.de,
[2] University of Tehran, Tehran, Iran
[3] Shahid Beheshti University, Tehran, Iran

Abstract. Risk management is a crucial factor to consider before any investment decision. Risk is measured based on the deviation of the prices from our expectation, most common way to measure risk is using standard-deviation. In this abstract, we introduce a new method for risk measurement using Level-Crossing analysis. In this method, we calculate the waiting time and the frequency of different events in the markets. The results are compared with the corresponding fBm data with the same Hurst exponent as the time series of stocks. For instance US stock markets Dow Jones Industrial Average (*DJIA*) and *S&P* have Hurst-exponent around $H = 0.5$, Shanghai Stock Exchange (*SSE*) and Tehran Stocks (*TEPIX*) have Hurst Exponent around $H = 0.6$. Therefore, according to the Hurst exponents of the markets, the frequency of the different values in fBm data with the same Hurst exponent is an appropriate standard reference point for our expectation of that market . Results show, although US stock markets are more active and have less average waiting time in compare with other two markets, they have a high compatibility with fBm data, specially in extreme events demonstrate the low level of risk in these markets. At the other hand Shanghai and Tehran markets are less active, having higher average waiting time and deviate from its corresponding fBm data, show the high level of risk in these markets.

1 Introduction

We apply the level crossing analysis [1] for measuring expected risk from different financial markets. In the level crossing analysis, we are interested in determining the average frequency ν_α^+ of observing a definite value (an event) in time series and as the consequence measuring the expected waiting time $(\nu_\alpha^+)^{-1}$ for its occurrence. Consider a time series with stochastic value $x(t)$, One can find the averaged number N_α^+ of crossing the given value $x(t) - \bar{x} = \alpha$ (with positive slope) in a time series, $N_\alpha^+ = \nu_\alpha^+ T$, T is the total time of the series. It is shown that ν_α^+ can be derived from joint probability distribution of $y = x(t) - \bar{x}$ and its derivation $\frac{dy}{dt}$ (figure1). A quantity $N_{tot}^+ = \int_{-\infty}^{\infty} \nu_\alpha^+ d\alpha$ could be introduced as the total number of crossing the time series with positive slope. In comparison of two time series with the same value of T, the one with the higher value of N_{tot}^+ has more fluctuations. This quantity could be generalized for different values of moments as $N_{tot}^+(q) = \int_{-\infty}^{\infty} |\alpha - \bar{\alpha}|^q \nu_\alpha^+ d\alpha$, which for the case that $q >> 1$ the extreme values of time series (the ones in the tail of distribution) gets more importance and for $q < 1$, the fluctuations close to the average (in the middle of

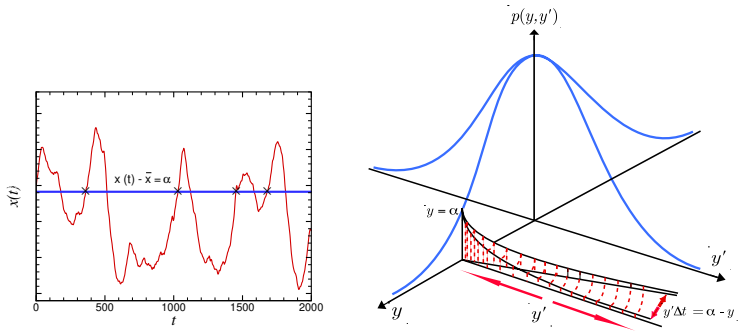


Fig. 1. (Left) crossing of an arbitrary level $x(t) - \bar{x} = \alpha$. (Right) Joint probability distribution function of the time series and the corresponding derivative in respect to time, shaded area shows the probability of a crossing with positive slope.

distribution) are more important. If N_{tot}^{+} is smaller/bigger than its shuffled values, the time series is correlated/uncorrelated. We applied the level crossing method for the time series from different stock markets with Hurst exponent (H=0.5 and H=0.6), and compared the results with the corresponding fBm series with the same Hurst exponent [2].

2 Results and Discussion

Four different time series of stock markets, *DJIA*, *S&P*, *SSE* and $TEPIX$ are analyzed using level crossing method. Expected waiting time and frequency for different levels of these series were calculated and compared with the corresponding fBm time series (figure 2). *DJIA* and *S&P* show the same behavior and have less deviation from the corresponding fBm series, demonstrate these markets behave as our expectation, therefore the level of risks in these markets are low. *SSE* and $TEPIX$, for smaller values of investment show more or less same behavior with less deviation from fBm series, but by increasing the value of investment, the deviation from fBm data increases. Results show Tehran market has higher level of risk than Shanghai and both of them are riskier than US markets. For a definite value of α, the average waiting time for US markets are always less than *SSE* and $TEPIX$ demonstrate the high level of activity in US stocks. In figure. 3, the total number of crosses or frequencies in all values of αs in different moments are plotted. *DJIA* and *S&P* have a good match with fBm series $H = 0.5$. *SSE* and $TEPIX$ deviate from $H = 0.6$, show the high level of risk in these markets, specially in extreme events.

References

1. Shahbazi, F, Sobhanian, S, Rahimi Tabar, M.R, Khorram, S, Frootan, G.R and Zahed, H: Level Crossing Analysis of Growing surfaces. J. Phys. A: Math. Gen. 36, 2517–2524 (2003).

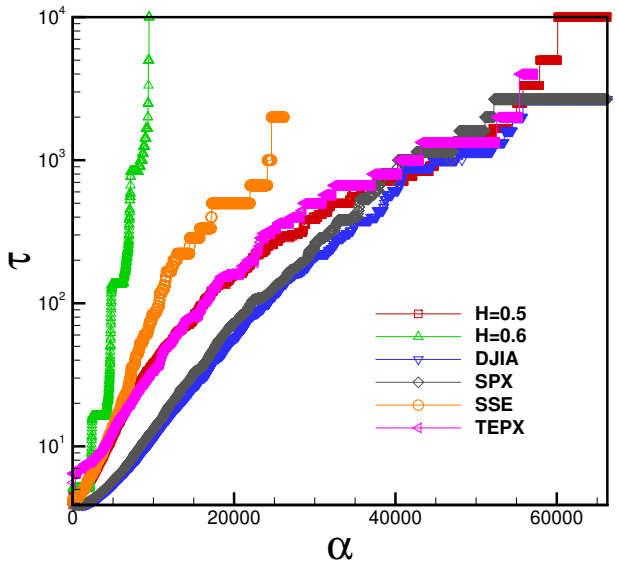


Fig. 2. Comparison of waiting times (τ) in different levels for four different Stock Markets.

2. Chen, L, Bassler, K.E, McCauley, J.L. and Gunaratne G.H. : Anomalous scaling of stochastic processes and the Moses effect. Phys. Rev. E. 95, 042141 (2017).

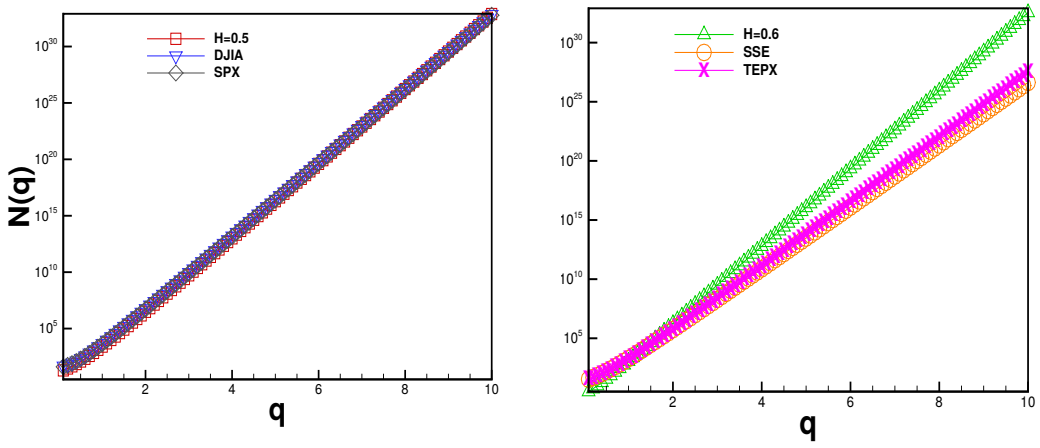


Fig. 3. Total number of crossing with positive slope in different values of moments q for four stocks and the comparison with the corresponding fBm series.

Power transformer monitoring based on a non-linear autoregressive neural network model with exogenous inputs

Javier Ramírez[1], Francisco. J. Martinez-Murcia[1], Fermín Segovia[1], Susana Carrillo[2], Javier Leiva[2], Jacob Rodriguez-Rivero[2], and J. M. Górriz[1]

[1] Dept. Signal Theory, Telematics and Communications, University of Granada, Granada, SPAIN,
javierrp@ugr.es,
[2] Endesa Distribución, Madrid, SPAIN

Abstract. Next generation of low- and medium-voltage distribution networks will demand to be better planned, operated and supervised as transportation networks have been managed for decades. The adaptation of these networks will require to incorporate much more intelligence, sensorization, broadband communications, optimal control and intelligent reporting among other emerging technologies. This paper shows a non-linear autoregressive neural network with exogenous inputs (NARX) for time series forecasting in these scenarios. The network model provides a description of the system by means of a non-linear function of lagged inputs, outputs and prediction errors that can be interpreted as a recurrent dynamic network, with feedback connections enclosing several layers of the network. It consists of a multilayer perceptron (MLP) in the hidden layer that takes as input a window of past independent (exogenous) inputs and past outputs followed by an output layer that finally forecast the target time series. The proposed NARX network was trained and evaluated in open-loop and closed-loop modes for prediction of the temperature of the transformer based on other correlated exogenous inputs that were recorded at the power transformation centers. The results show that the NARX networks can accurately predict and monitor the operation of power transformers.

Keywords: power transformers, time-series forecasting, non-linear discrete-time modelling, NARX networks, power distribution networks, fault diagnosis, power transformer monitoring

1 Introduction

In the next years, with a growing presence of electric vehicles and a massive penetration of renewable sources, with low levels of voltage for self-consumption, it will be essential that medium- and low-voltage distribution networks be planned, operated and supervised as transportation networks have been managed for decades, from the distributor to be a simple agent of distribution assets to be operator of the network. In order to accomplish it, the systems has to incorporate

much more intelligence than before, which involves a whole spectrum of digital technologies: sensorization, smart meters, broadband communications, local electronic device controllers, IoT (Internet of Things), SCADAs (Supervision, Control and Acquisition of Data), energy management centers, advanced data processing software (data analytics), optimal control, intelligent reporting, etc.

Non-linear system modeling techniques have significantly evolved during the last decades. Among them, the non-linear autoregressive moving average with exogenous inputs (NARMAX) [1–3] model represents a wide class of discrete-time non-linear systems. The NARMAX model provides a description of the system by means of a non-linear function of lagged inputs, outputs and prediction errors. Since the definition of the NARMAX model is independent of the non-linear functional, multi-layered neural networks offer a powerful alternative in this context for modelling complex non-linear systems and time series forecasting. Among these networks, the non-linear autoregressive neural network with exogenous inputs (NARX) can be interpreted as a recurrent dynamic network, with feedback connections enclosing several layers of the network and has found application in many real scenarios for time series forecasting [4–9].

A large collection of predictive techniques and methods to diagnose the health of power transformers are available in the literature [10–13]. These techniques are classified as off-line or on-line methods depending if the monitoring process requires to disconnect the transformer or not. Expert knowledge and experienced engineers are needed to correctly interpret the results of the monitoring process. This paper shows a power transformer monitoring approach based on a non-linear autoregressive discrete-time model with exogenous inputs and neural networks.

2 The NARMAX model for power transformer monitoring

The NARMAX model [1–3] is a discrete-time non-linear system model in which the next value of the dependent output signal $y(t)$ is defined to be a function of past values of the output signal and previous values of independent (exogenous) input signals

$$y(t) = f(y(t-1), y(t-2), ..., y(t-n_y), \mathbf{u}(t-1), \mathbf{u}(t-2), ..., \mathbf{u}(t-n_u)) \quad (1)$$

where the $\mathbf{u}$ vector represents the set of exogenous signals that are used to predict the value of the output signal $y(t)$.

Figure 1 shows the architecture of open-loop and closed-loop non-linear autoregressive neural networks with exogenous inputs. It consists of a multilayer perceptron (MLP) in the hidden layer that takes as input a window of past independent (exogenous) inputs $\{\mathbf{u}(t-1), \mathbf{u}(t-2), ..., \mathbf{u}(t-n_u)\}$ and past outputs $\{y(t-1), y(t-2), ..., y(t-n_y)\}$ and calculates the current output $y(t)$, followed by an output layer. These networks are usually trained in open-loop mode using

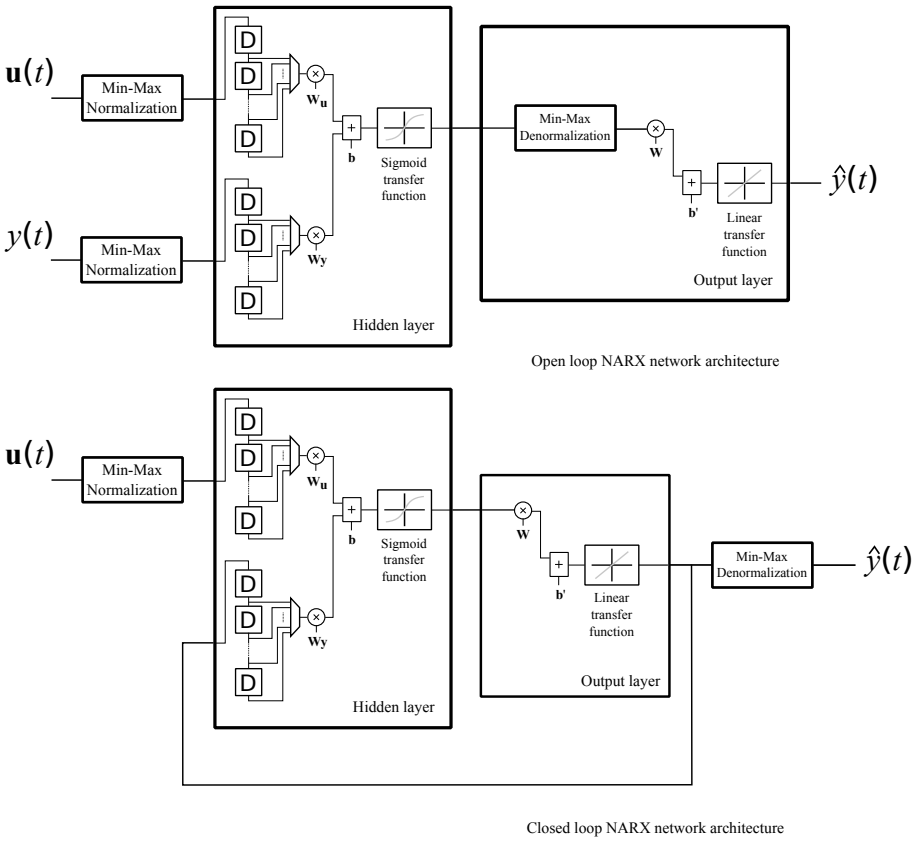


Fig. 1. Architecture of open-loop and closed-loop non-linear autoregressive neural networks with exogenous inputs.

past values of the exogenous signals and the output to predict the next sample. Once the network is trained, the time series forecasting problem can be accomplished without using past values of the output signal and based only on the exogenous inputs. This step requires to introduce a feedback from the output layer to the input of the hidden layer so that the past values of $y(t)$ are replaced by their predictions.

3 Datasets

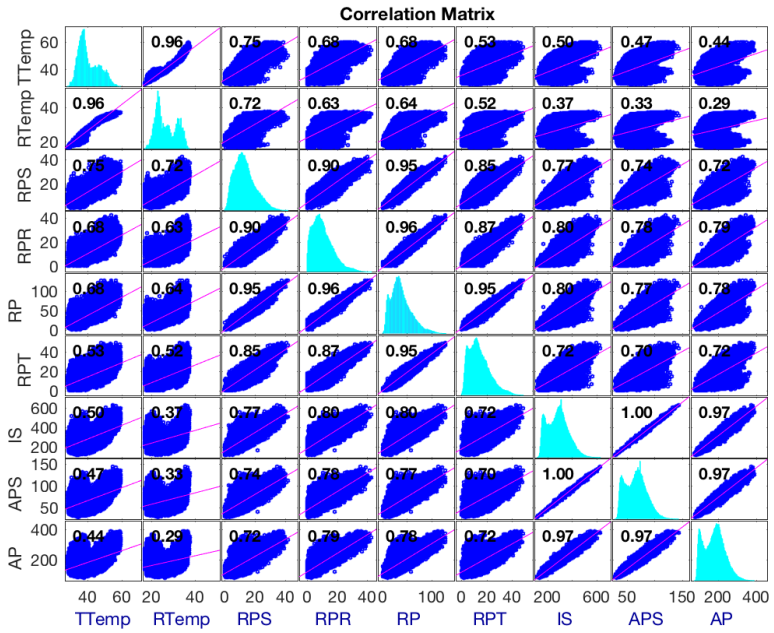
Data used in this study was provided by Endesa company through the smart grid of Smartcity Malaga Living Lab. For each of the 17 power transformers, a total of 20 variables were recorded at a sample rate of 12 samples/hour for the whole 2018 year. These signals include the 'Phase Imbalance (PI)', 'Active Energy Exported (AEE)', 'Active Energy Imported (AEI)', 'Capacitive Reactive Energy Exported (CREE)', 'Capacitive Reactive Energy Imported (CREI)', 'Inductive Reactive Energy Exported (IREE)', 'Inductive Reactive Energy Imported (IREI)', 'Intensity R (IR)', 'Intensity S (IS)', 'Intensity T (IT)', 'Active Power R (APR)', 'Active Power S (APS)', 'Active Power T (APT)', 'Active Power (AP)', 'Reactive Power R (RPR)', 'Reactive Power S (RPS)', 'Reactive Power T (RPT)', 'Reactive Power (RP)', 'Room Temperature (RTemp)', 'Transformer Temperature (TTemp)', 'Tension R (TR)', 'Tension S (TS)', and 'Tension T (TT)', and were used to monitor the operation of the power transformer, fault diagnosis and rapid intervention. The temperature of the power transformer is then the main target of the time series prediction problem. Time series forecasting assumes the use of an accurate model of the power transformer that is able to predict future values of the signals based on previously observed values of them and/or other exogenous time series.

4 Correlation analysis

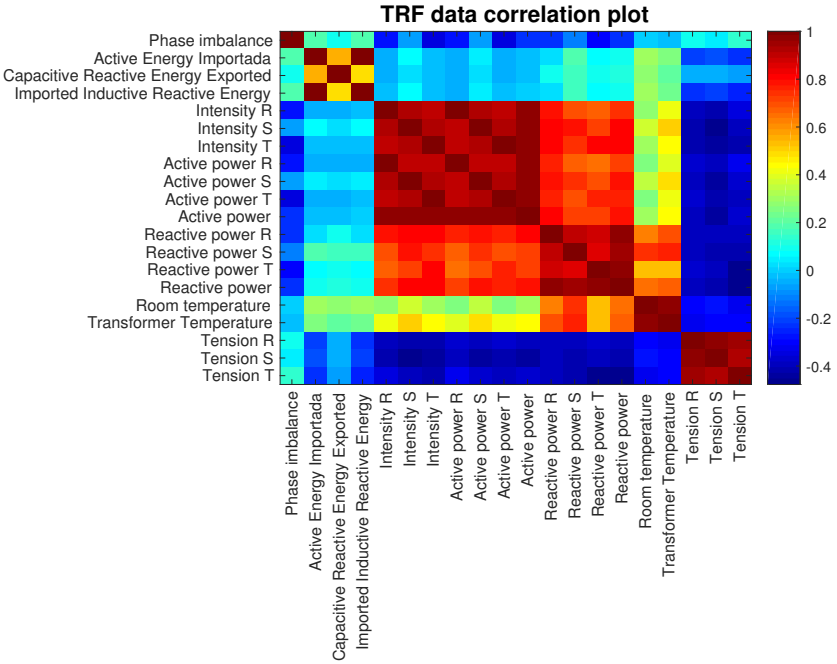
A correlation analysis was conducted in order to determine correlations among pairs of variables for each of the power transformers in the dataset. This information will be used to identify the exogenous inputs enabling forecasting the temperature of the power transformers. Figure 2.a provides a matrix of plots showing the correlations among pairs of variables that are highly correlated with the TTemp target. Note that, histograms of the variables are shown along the diagonal of the matrix plot while scatter plots of variable pairs appear in the off diagonal. The slopes of the least-squares linear regression problem are equal to the displayed correlation coefficients. Finally, the correlation coefficients are summarized in figure 2.b for a given power transformer in the dataset.

5 Experimental results

Before carrying out the training process of the NARX network, the transformed signals that were digitized were resampled with a uniform sampling rate of 12



(a)



(b)

Fig. 2. Correlation analysis. a) Correlation plots between multiple time series, b) Correlation matrix between transformer data time series.

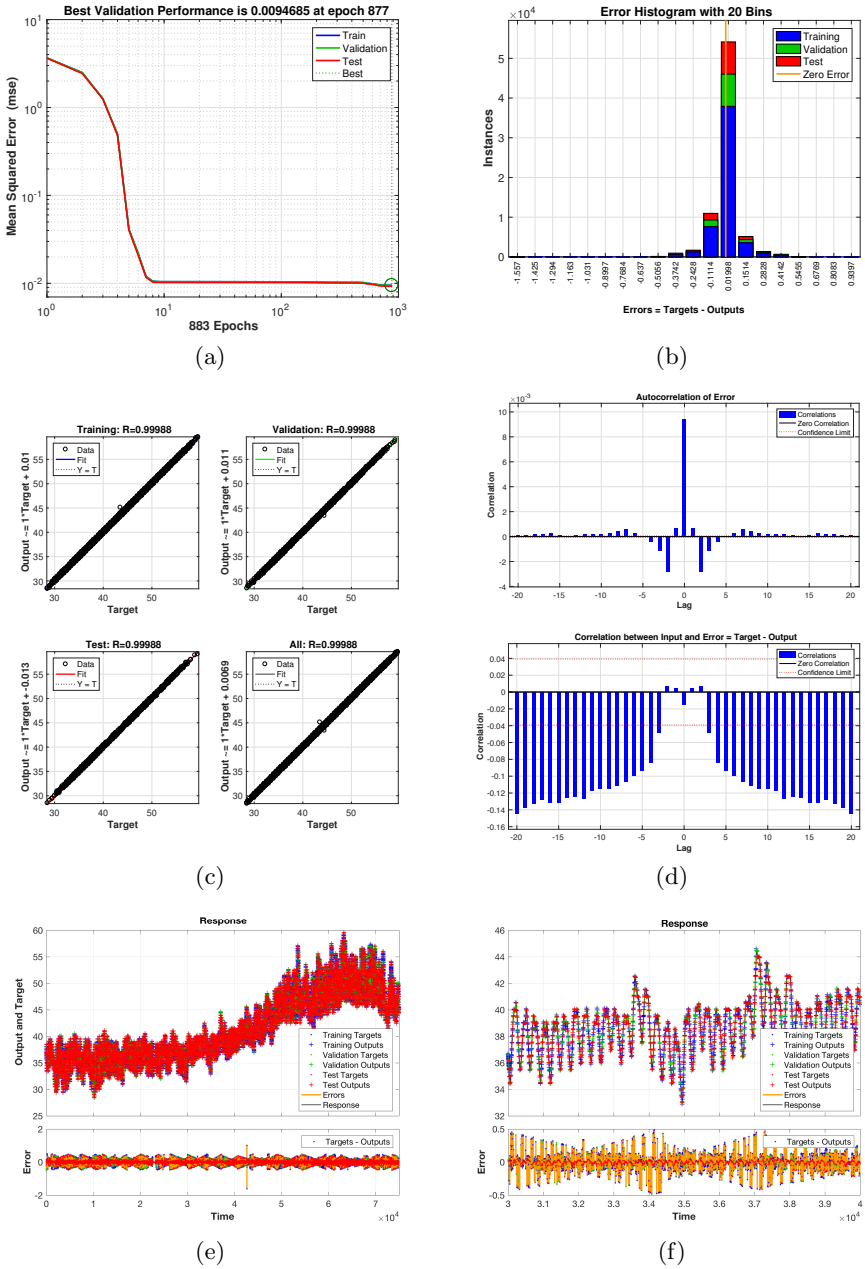


Fig. 3. NARX training: a) MSE as a function of the number of epochs, b) Histogram of the time series forecast error, c) Output-target linear regression, d) Autocorrelation of the time series prediction error and correlation among the input and the prediction error, e) Response on unseen data (output, target and error) of the NARX network, and f) Detail (zoom) of the response on unseen data of the NARX network.

samples/hour. This process is necessary since the signals stored in the cloud are not registered with a uniform period in all the cases and the NARX networks are based on an uniform sample rate discrete-time model. The corresponding resampling process was based on a simple linear interpolation between consecutive samples.

Once the resampling procedure was carried out, an analysis of the variables recorded in the transformer that could have influenced the variation of the transformer temperature was carried out. In this way, an experiment was conducted in which the TTemp time series was predicted based on other transformer variables such as RTemp, AP, APR, APS and APT, through a two-delay elements NARX network consisting of a 10-neuron single MLP-based hidden layer. In a first phase, 75.000 data samples were used for cross validation (70% for training, 15% for validation and 15% for testing). Figure 3 shows that:

a) The convergence of the network in terms of the mean squared error (MSE) as a function of the number of epochs. It is verified that the MSE converges to 0.01 for about 10 epochs,
b) The histogram of the prediction error. It is centered at zero mean and has a reduced variance,
c) The linear output-objective regression results. A good fit between both variables is shown with a regression coefficient close to unity,
d) The autocorrelation of the error and the correlation between the input and the prediction error, and
e) The response of the model for the prediction of the temperature of the transformer (TTemp).

Once the open-loop NARX network was trained, the closed-loop network model was built in order to make a prediction of the the temperature of the transformer (TTemp) without using previous values of the target. Thus, the closed-loop NARX network only uses the values of the exogenous variables RTemp, AP, APR, APS and APT to predict the temperature of the transformer. To evaluate this second model, 25.000 samples not used previously for training were used. Figure 4 shows the output of the NARX network in closed loop and the target. It can be concluded that the NARX network model used for monitoring the temperature of the transformer yields a high prediction accuracy.

6 Conclusion

NARX networks were evaluated for time series forecasting in low/medium voltage power transformation centers. The proposed NARX network models predicted the temperature of the transformer as a function of past values of outputs and exogenous inputs. The system was then described by a non-linear function of lagged inputs, outputs and prediction errors that can be interpreted as a recurrent dynamic network, with feedback connections enclosing several layers of the network. A dataset consisting of a total of 20 variables that were recorded at a sample rate of 12 samples/hour for each of the 17 power transformation

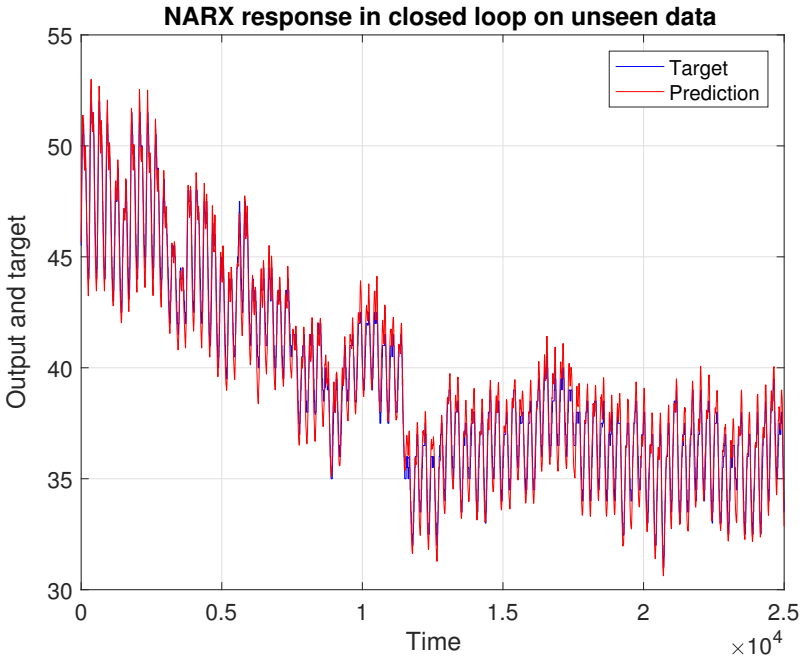


Fig. 4. Closed-loop NARX response (output and target) on unseen data.

centers during the whole 2018 year. A correlation analysis was conducted in order to identify those variables that has more impact in the target. The NARX networks was trained and evaluated by cross validation showing a high accuracy when operated in open-loop and closed-loop modes.

Acknowledgment

The authors would like to acknowledge the support of CDTI (Centro para el Desarrollo Tecnológico Industrial, Ministerio de Ciencia, Innovación y Universidades and FEDER) under the PASTORA (Preventive analysis of intelligent networks in real time and integration of renewal resources) project (Ref.: ITC-20181102).

References

1. Chen, S., Billings, S.A., Grant, P.M.: Non-linear system identification using neural networks. International Journal of Control **51**(6) (1990) 1191–1214
2. Tsungnan Lin, Horne, B.G., Tino, P., Giles, C.L.: Learning long-term dependencies in narx recurrent neural networks. IEEE Transactions on Neural Networks **7**(6) (Nov 1996) 1329–1338

3. Siegelmann, H.T., Horne, B.G., Giles, C.L.: Computational capabilities of recurrent narx neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **27**(2) (April 1997) 208–215
4. Guzman, S.M., Paz, J.O., Tagert, M.L.M.: The use of narx neural networks to forecast daily groundwater levels. Water Resources Management **31**(5) (Mar 2017) 1591–1603
5. Marcjasz, G., Uniejewski, B., Weron, R.: On the importance of the long-term seasonal component in day-ahead electricity price forecasting with narx neural networks. International Journal of Forecasting (2018)
6. Cadenas, E., Rivera, W., Campos-Amezcua, R., Heard, C.: Wind speed prediction using a univariate arima model and a multivariate narx model. Energies **9**(2) (2016)
7. Vaz, A., Elsinga, B., van Sark, W., Brito, M.: An artificial neural network to assess the impact of neighbouring photovoltaic systems in power forecasting in utrecht, the netherlands. Renewable Energy **85** (2016) 631 – 641
8. Basso, M., Giarre, L., Groppi, S., Zappa, G.: Narx models of an industrial power plant gas turbine. IEEE Transactions on Control Systems Technology **13**(4) (July 2005) 599–604
9. Villacci, D., Bontempi, G., Vaccaro, A., Birattari, M.: The role of learning methods in the dynamic assessment of power components loading capability. IEEE Transactions on Industrial Electronics **52**(1) (Feb 2005) 280–290
10. Booth, C., McDonald, J.: The use of artificial neural networks for condition monitoring of electrical power transformers. Neurocomputing **23**(1) (1998) 97 – 109
11. de Faria, H., Costa, J.G.S., Olivas, J.L.M.: A review of monitoring methods for predictive maintenance of electric power transformers based on dissolved gas analysis. Renewable and Sustainable Energy Reviews **46** (2015) 201 – 209
12. AJ, C., Salam, M., Rahman, Q., Wen, F., Ang, S., Voon, W.: Causes of transformer failures and diagnostic methods a review. Renewable and Sustainable Energy Reviews **82** (2018) 1442 – 1456
13. Catterson, V.M., McArthur, S.D.J., Moss, G.: Online conditional anomaly detection in multivariate data for transformer monitoring. IEEE Transactions on Power Delivery **25**(4) (Oct 2010) 2556–2564

Partial Least Squares for the Characterization of Meditation and Attention States

Jorge García-Torres[1], Juan M. Gorriz[1,*], Javier Ramírez[1], F J. Martínez-Murcia[1,*]

Dept. of Signal Theory, Networking and Communications. University of Granada, Spain

Abstract

Electroencephalographic activity (EEG) has allowed the exploration and development of a new non-muscular communication channel under the name of brain-computer interfaces (BCI), which eases the study of neurological disorders and brain functions. This paper deploys machine learning algorithms based on partial least squares feature extraction to characterize two easily recognizable cognitive states: attention and meditation. The classification problem is addressed by using the power spectrum density of the raw EEG values provided by a low-cost BCI system as the input data of our experimental approach. Accuracies around 80% over a dataset of 30 subjects are obtained by combining the two first PLS components with a RBF kernel, significantly improving the results provided by other approaches.

Keywords: EEG, BCI, Partial Least Squares, Classification, SVM, Characterization, Cognitive states.

1. Introduction

Electroencephalography is a non-invasive technique of functional exploration of the central nervous system by which electrical brain activity can be recorded (Niedermeyer and da Silva (2005)). It has emerged as an area of great clinical interest and it is an indispensable tool in neurology for the study, analysis and diagnosis of various brain functions and diseases, abnormalities or biological difficulties originating in the brain (Friedman et al. (2009)). Despite the

*Corresponding author

Email addresses: jorgegtf@hotmail.com (Jorge García-Torres), gorriz@ugr.es (Juan M. Gorriz), javierrp@ugr.es (Javier Ramírez), fjesusmartinez@ugr.es (F J. Martínez-Murcia)

[1]Dept. of Signal Theory, Networking and Communications. University of Granada, Spain

fact EEG recordings have a low spatial resolution when compared with other electrophysic measurements of the brain function, it offers an easy acquisition, becoming the most used technique in the Brain-Computer Interfaces (BCI) development (Leuthardt et al. (2004); Lotte et al. (2007)). Therefore, if brain activity can be measured and processed using these affordable devices, it is possible to recognize the stimuli that generate it and even reproduce it. In recent years, there have been an emergence of a consumer market for inexpensive, mobile EEG devices for BCI (Ponce et al. (2014)). Thus, research has resulted in several mobile BCI systems for detecting emotional states and event-related potentials. Generally, these devices have the purpose of recognizing the different user's mental gestures as a symbol of a finite set of discrete symbols, a problem that can be identified as a pattern recognition issue (Lotte et al. (2007)). The principal difficulty is the variable and non-stationary nature of the neural signals since each symbol is expressed differently between individuals, and can be affected by others factors such as mood and stress (Vidaurre and Blankertz (2010)).

In this study, an experimental approach will be implemented to classify two cognitive states: attention and meditation. Several optimization and statistical methods employed will be briefly presented. Then, an experimental approach will be developed to establish a comparison between different feature extraction procedures. Finally, the results will be shown and analyzed, indicating the technique that provides a better performance for the classification problem.

2. Materials and Methods

In general, pattern recognition implies three principal stages: preprocessing, feature extraction and classification (Lau and Wu (2003)). All of them were considered in our experimental setup, as shown in figure 1. Data preprocessing includes a reduction algorithm of EMG artefacts and power line noise cancelation performed during the recording of the EEG by the own device used for the dataset. As well, a Multitaper power spectrum estimation (Xu et al. (1999)) together with a standardization stage based on the z-scores were performed over the raw values of EEG. In the following sections we focus on the definition of partial least squares as a feature extraction method, and the basis of the SVM paradigm for classification. Finally, the results are compared with other feature extraction methods: principal component analysis or PCA (Jolie (1986)) and a technique based on a logarithmic quantization proposed in Merril (2016).

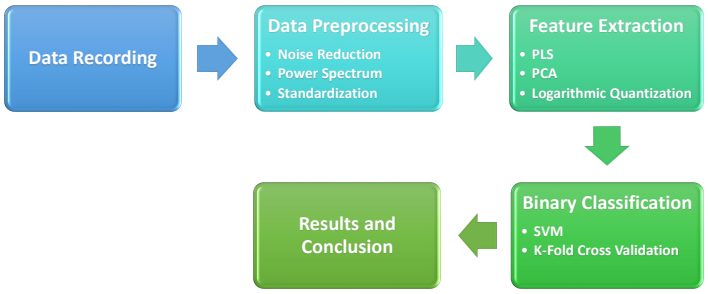


Figure 1: Flow diagram of the procedure used in the proposed classification algorithm.

2.1. Database

The dataset was shared by the MIDS class at the UC Berkeley School of Information (Chuang et al. (2015)). Data have been acquired through the NeuroSky Mindwave device (Neurosky MindWave (2011)) from a group of students to which audiovisual stimuli has been shown. Particularly, two slightly different stimuli were presented to two different groups. For both stimuli, a group of 15 people saw the stimuli at the same time, while EEG data was being collected with a sampling frequency of 512 Hz. Each stimulus, a video with an approximate duration of 5 minutes and 20 seconds, consists of performing a series of exercises, such as blinking, relaxation, math calculations, listening music, watching an advertisement, thinking about certain objects and looking at colours, with the objective of generating in the subjects different mental states. The device includes an algorithm called NeuroSky eSense which provides metrics for estimating the grade of attention or meditation each subject shows. Together with this, the dataset includes readings of all the subjects during the stimulus presentation, as well as readings from before the start and after the end of the stimulus. The server receives one data packet every second from each Mindwave Mobile device, and stores the data in one row entry.

2.2. Feature Extraction

Feature extraction is a procedure applied to original data for reducing dimensionality. This generates feature vectors that simplify and favor the classification stage. It must be done in such a way that the least possible amount of information is eliminated and the new data can be replaced by the original one.

A. *Partial Least Squares.* Partial least squares (Wold et al. (1984)) comprises regression and classification tasks as well as dimension reduction and modeling tools. It consists of building new predictors variables, similar to PCA, but also considering the class labels (Khedher et al. (2015)). In short, this method aims to combine both information about the variances of the predictors and the response as well as the correlations between them. Taking the observed variables $X \subset \mathbb{R}^N$ and their labels $Y \subset \mathbb{R}^M$ for n samples, PLS descomposes the $n \times N$ zero-mean variables $\mathbf{X}$ and the $n \times M$ zero-mean variables $\mathbf{Y}$ into the regression models form (Bastien et al. (2005)):

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E} \tag{1}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \mathbf{F} \tag{2}$$

where $\mathbf{T}$ and $\mathbf{U}$ are $n \times p$ matrices of the p extracted score or latent vectors (components), the $N \times p$ matrix $\mathbf{P}$ and the $M \times p$ matrix $\mathbf{Q}$ represent matrices of loadings with number of columns being the number of PLS components and $n \times N$ matrix $\mathbf{E}$ and the $n \times M$ matrix $\mathbf{F}$ are the matrices of the residuals.

B. *Logarithmic quantization.* Method proposed to extract better feature vectors in which a logarithmic quantization technique (Merrill et al. (2015)) is applied over the raw values of EEG. Once data is standardized, the power spectrum with the standard periodogram is calculated for each row of raw values. A set of three periodograms is averaged and logarithmically spaced to generate $100-$values feature vectors, offering a way to quantize the information contained in the signal and to summarise the statistical properties of the power spectrum for each cognitive state.

2.3. Classification

Support Vector Machine is an algorithm commonly used for medical applications (Ramírez et al. (2010); Segovia et al. (2013); Khedher et al. (2015)). It is a supervised learning method that takes a set of labeled data to generate a class prediction model of non-labeled data, using a hyperplane to draw discriminatory boundaries between classes and maximizing the distance between the nearest training points (Burges (1998)). The aim is to find the function $f : \mathbb{R}^p \mapsto \{\pm 1\}$ that allows one to assign a binary value to the new sample. This function is generated from the training data X, which contains N, $p \times 1$ vectors (or x_i samples) and its corresponding class y_i. The decision hyperplane in the feature space is defined by a linear discriminant function (Lau and Wu (2003)):

$$(x_1, y_1), (x_2, y_2), ..., (x_N, y_N) \in \mathbb{R}^p \times \{\pm 1\}, \quad g(x) = w^T x + b \tag{3}$$

where $g(x) = 0$ is the optimal hyperplane, w is the coefficient vector that is orthogonal to the decision hyperplane and b is the threshold. The SVM maximizes the distance between classes (margin), thus avoiding overfitting the data (Vapnik (1998)). For non-linearly separable classes, non-linear functions are used to transform the original low-dimensionality space to a high-dimension feature space (Vapnik (1998)), transfering the nonlinear separable data to linear separable data in the feature space. Since the new space may become computationally infeasible when the dimension of the input space is large, a kernel trick is applied. The inner product involved in the solution of the optimization problem can be replaced with a kernel function K: a function that quantifies the similarity of two observations $g(x) = K(w, x) + b$. The computational advantage is that it is only neccessary to compute $K(w, x)$ for all distinct observations, without explicity working in the enlarged feature space. The most commonly kernel function are: Linear, Gaussian or Radial Basis Function and Polynomial.

3. Experimental Framework

To carry out the binary classification stage, the same two exercises were selected from the audiovisual stimuli: relaxation (relax) and mathematical calculations (math), associated with a cognitive state of meditation and attention respectively. Our binary classification process considers SVMs with different kernels. To obtain reproducible results, the feature vectors were, as aforementioned, standardized using the z-scores. For a proper model validation, a k-fold cross-validation (Kohavi et al. (1995)) was employed. The results were averaged to obtain an estimate of the general performance of the classification stage. To reduce the variability of the results obtained in random selected folds, a Monte Carlo simulation was employed, averaging the results after repeating the process a sufficient number of times, i.e. $N_F = 50$ iterations. All experiments were run in an AMD FX-7500 Radeon R7 at 2100 MHz with 8 GB RAM.

3.1. Evaluation

To analyze the performance of the classifier in depth (Vapnik (1998)), several parameters are obtained from the generated confusion matrix. The receiver operating characteristic (ROC) curve is a statistical metric used to check the quality of a classifier (Hanley and McNeil (1982)). It shows the true positive rate versus the false positive rate for different thresholds of the classifier output,

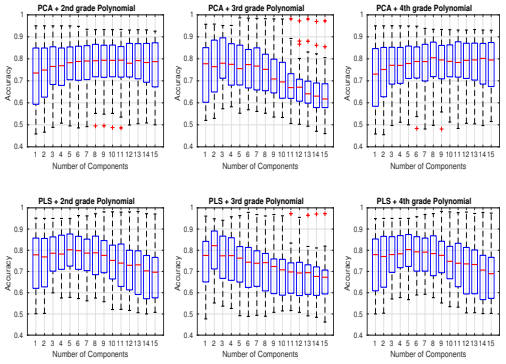


Figure 2: Box plots of the accuracy for a 2nd, 3rd and 4th grade polynomial kernel either with PLS and PCA.

revealing the one that maximizes the classifier accuracy and evaluating how the classifier works in the regions of high sensitivity and high specificity. The area under the curve (AUC) is a typical measurement assessed of the ROC curve.

3.2. Results

The selection of the optimal number of PLS and PCA components must be performed experimentally. First of all, the grade of the polynomial kernel is analyzed to check which one provides the best performace. Figure 2 represents the accuracy in terms of distribution along different number of components for 2^{nd}, 3^{rd} and 4^{th} polynomial grade, showing that the best result is obtained with polynomial kernel of grade three. Then, the results provided by all kernel methods mentioned are compared. Figure 3 shows the accuracy distribution along the number of components with different kernel functions. The analysis is carried out selecting the best accuracy achieved with the lowest number of components over all the experiments performed. Therefore, as figure 3 reflects, the number of components chosen either for PCA and PLS is two. Some examples of the data distribution with two components are shown in figure 4.

To evaluate our experimental approach, the results of the classifiers with different feature extraction techniques are compared. Thus, the average of the ROC curves family corresponding to all the subjects using vertical averaging (Fawcett (2006)) is analyzed. As figure 5 represents, it seems that PCA provides very similar results in terms of average accuracy that the ones provided by the logarithmic quantization technique employed by Nick Merril (Merril (2016)). In contrast, PLS achieves to improve briefly the performance of the classifier.

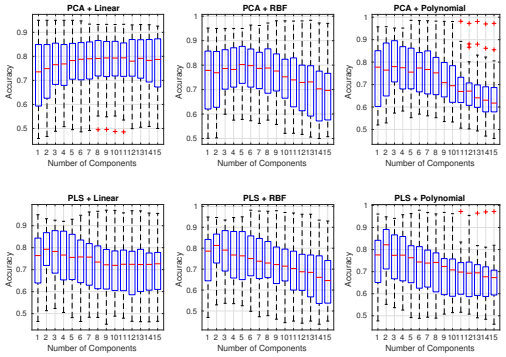


Figure 3: Box plots of the accuracy for different kernel functions.

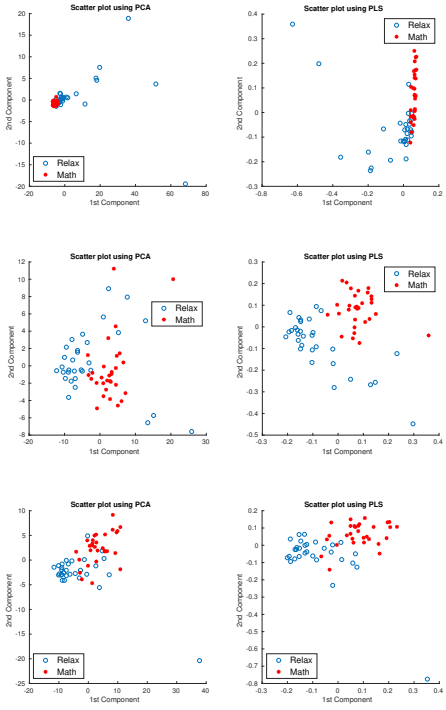


Figure 4: Bidimensional representation of the first two components for the subjects number 1, 5 and 25 respectively, distinguishing the classes under study. Since both exercises have a duration of 30 seconds, there is a total of 60 samples per subject.

Table 1 shows the efectiveness of each classifer reflecting that the best performance is provided by PLS together with a RBF kernel. This performance can be

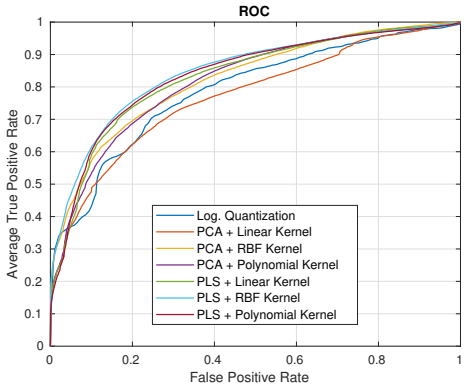


Figure 5: Reciever operating characteristic curves of the classifier using different feature extraction techniques and kernel functions.

Method	Accuracy	Sensitivity	Specificity	AUC
Log. Quantization	0.7569 (0.1587)	0.7479 (0.1624)	0.7716 (0.1793)	0.7939 (0.1761)
PCA + Linear	0.7394 (0.1430)	0.7119 (0.2083)	0.7668 (0.2293)	0.7713 (0.1840)
PCA + RBF	0.7586 (0.1291)	0.7214 (0.1726)	0.7958 (0.1762)	0.8241 (0.1321)
PCA + Polynomial	0.7681 (0.1259)	0.7372 (0.1772)	0.7992 (0.1739)	0.8139 (0.1237)
PLS + Linear	0.7793 (0.1162)	0.7701 (0.1219)	0.7880 (0.1661)	0.8316 (0.1193)
PLS + RBF	0.7932 (0.1110)	0.7796 (0.1096)	0.8065 (0.1341)	0.8476 (0.1167)
PLS + Polynomial	0.7926 (0.1157)	0.7760 (0.1314)	0.8092 (0.1321)	0.8372 (0.1164)

Table 1: Performance measurements average of all the 30 subjects. The values in brackets represent the standard deviation.

observed with greater detail for every subject in table 2.

Results can be analyzed from a different perspective considering the metrics of attention and meditation provided by the NeuroSky MindWave device. Taking the current cognitive state of the subject at each time as the one with greater value, a classification with the labels provided by Neurosky eSense and the ground truth can be done (see figure 6), showing a low performance.

4. Discussion

We have devised a feature extraction method that simplifies and favors the binary classification stage of two cognitive states, attention and meditation. Previously, as a preprocessing step, the Multitaper power spectrum estimation

Subject	Accuracy	Sensitivity	Specificity	Positive Likelihood	Negative Likelihood
1	0.8837	0.8767	0.8907	8.2973	0.1385
2	0.8534	0.8720	0.8355	5.3434	0.1532
3	0.9414	0.8756	0.9969	-	0.1249
4	0.8468	0.7123	0.9813	-	0.2934
5	0.8367	0.8239	0.8500	5.7194	0.2075
6	0.6227	0.5680	0.6773	1.8066	0.6419
7	0.7492	0.7342	0.7647	3.1921	0.3480
8	0.8773	0.8487	0.9060	11.1587	0.677
9	0.8053	0.7913	0.8193	4.5962	0.2551
10	0.7710	0.7027	0.8393	4.7837	0.3552
11	0.7925	0.7219	0.8653	6.1535	0.3223
12	0.8363	0.7780	0.8947	7.7383	0.2481
13	0.9410	0.9000	0.9820	-	0.1019
14	0.7120	0.7453	0.6787	2.3511	0.3759
15	0.7272	0.6462	0.8083	3.4728	0.4385
16	0.7490	0.8860	0.6120	2.2960	0.1872
17	0.8377	0.8155	0.8607	6.4873	0.2149
18	0.9153	0.8847	0.9460	-	0.1222
19	0.9188	0.8839	0.9515	21.1732	0.1221
20	0.5477	0.6227	0.4727	1.2027	0.8183
21	0.7127	0.6733	0.7520	2.7642	0.4355
22	0.8250	0.8740	0.7760	3.9869	0.1631
23	0.5863	0.5720	0.6007	1.4451	0.7148
24	0.9343	0.9440	0.9247	13.6700	0.0606
25	0.7457	0.7633	0.7280	2.8457	0.3259
26	0.5803	0.5677	0.5933	1.4111	0.7320
27	0.8727	0.8927	0.8527	6.4426	0.1261
28	0.7037	0.7527	0.6547	2.2181	0.3798
29	0.9205	0.8890	0.9520	21.5213	0.1167
30	0.7502	0.7703	0.7293	2.8714	0.3148

Table 2: Classifier performance with two PLS components and using a RBF kernel function.

is obtained. This procedure may generate large feature vector, so a data dimensionality reduction is required. Particularly, the selected reduction methods are PLS and PCA. In order to compare the implementation of different kernel functions in the SVM, in figure 2 the accuracy distribution of various polynomial kernels with different grade are represented, indicating that the one with better performance is the third grade polynomial. Then, as figure 3 shows, the performance of different classifiers are represented. In general terms, the accuracy tends to decrease as the number of PCA or PLS components increases, so just two components are chosen to carry out the analysis. Results are contrasted with the ones provided by Nick Merrill (Merril (2016)) employing a logarithmic quantization method proposed in Merrill et al. (2015). Following these techniques and according to the results in table 1, PLS outperforms PCA as a feature extraction technique. Regarding the method proposed in Merril (2016), PCA provides a similar performance whereas PLS achieves to improve this average accuracy an almost 4%, being a more significant improvement.

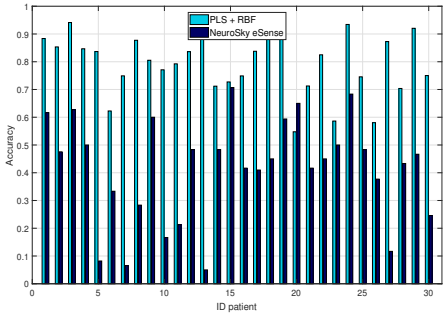


Figure 6: Comparation between the accuracy obtained per subject using the experimental approach proposed in this study and the NeuroSky eSense algorithm.

By its definition, PCA aims to find the subspace that maximizes the variance between the predictors without taking into account the label of each observation. PLS gives some importance to how each predictor is related to each response for establishing the components, acting in some way like a supervised dimension reduction technique. This is particularly interesting in cases in which the number of highly collinear independent predictors is higher than the number of observations. Note that the use of the power spectrum values seems to produce significant variations in the classifier performance among the subjects. This is probably due to a different behavior of each subject when performing the exercises (the subject might not be calm or focused enough during the exercises) or even the use of biased power spectrum estimations.

5. Conclusion

An experimental approach for classifying cognitive states based on EEG recordings was proposed in this paper. The algorithm combines the information given by EEG signals and a feature extraction procedure to detect meditation or attention states. As a baseline approach, the method proposed by Merrill et al. (2015) based on a logarithmic quantization technique was implemented for comparison purposes. On the other hand, PLS and PCA are used to reduce data dimension and to surmount the small-sample size problem. All these methods were proved to solve the binary classification problem based on SVM with different kernel functions. It has been shown that PLS with a RBF kernel reaches greater accuracy and performance, outperforming the remaining feature extraction procedures employed.

Acknowledgements

This work was partly supported by the MINECO/FEDER under the RTI2018-098913-B100 project.

References

Bastien, P., Vinzi, V. E., and Tenenhaus, M. (2005). Pls generalised linear regression. *Computational Statistics & Data Analysis*, 48(1):17 – 46. Partial Least Squares.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Chuang, J., Merril, N., and Thomas, M. (2015). Synchronized brainwave recordings from a group presented with a common audio-visual stimulus.

Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recogn. Lett.*, 27(8):861–874.

Friedman, D., Claassen, J., and Hirsch, L. J. (2009). Continuous electroencephalogram monitoring in the intensive care unit. *Anesthesia & Analgesia*, 109(2):506–523.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Jolie, I. (1986). Principal component analysis. springer-verlag. *New York*.

Khedher, L., Ramírez, J., Górriz, J., Brahim, A., and Segovia, F. (2015). Early diagnosis of alzheimer's disease based on partial least squares, principal component analysis and support vector machine using segmented mri images. *Neurocomputing*, 151(Part 1):139 – 150.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada.

Lau, K. and Wu, Q. (2003). Online training of support vector classifier. *Pattern Recognition*, 36(8):1913–1920.

Leuthardt, E. C., Schalk, G., Wolpaw, J. R., Ojemann, J. G., and Moran, D. W. (2004). A brain–computer interface using electrocorticographic signals in humans. *Journal of neural engineering*, 1(2):63.

Lotte, F., Congedo, M., Lécuyer, A., Lamarche, F., and Arnaldi, B. (2007). A review of classification algorithms for eeg-based brain–computer interfaces. *Journal of neural engineering*, 4(2):R1.

Merril, N. (2016). Classifying relaxation versus doing math.

Merrill, N., Maillart, T., Johnson, B., and Chuang, J. C.-I. (2015). Improving physiological signal classification using logarithmic quantization and a progressive calibration technique. In *PhyCS*, pages 44–51.

Neurosky MindWave (2011). *MindWave User Guide*.

Niedermeyer, E. and da Silva, F. L. (2005). *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins.

Ponce, P., Molina, A., C. Balderas, D., and Grammatikou, D. (2014). Brain computer interfaces for cerebral palsy. *Cerebral Palsy - Challenges for the Future*.

Ramírez, J., Górriz, J., Segovia, F., Chaves, R., Salas-González, D., López, M., Álvarez, I., and Padilla, P. (2010). Computer aided diagnosis system for the alzheimer's disease based on partial least squares and random forest spect image classification. *Neuroscience Letters*, 472(2):99 – 103.

Segovia, F., Górriz, J. M., Ramírez, J., Salas-Gonzalez, D., and Álvarez, I. (2013). Early diagnosis of alzheimer's disease based on partial least squares and support vector machine. *Expert Syst. Appl.*, 40(2):677–683.

Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

Vidaurre, C. and Blankertz, B. (2010). Towards a cure for bci illiteracy. *Brain topography*, 23(2):194–198.

Wold, S., Ruhe, A., Wold, H., and W. J. Dunn, I. (1984). The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses. *SIAM Journal on Scientific and Statistical Computing*, 5(3):735–743.

Xu, Y., Haykin, S., and Racine, R. J. (1999). Multiple window time-frequency distribution and coherence of eeg using slepian sequences and hermite functions. *IEEE Transactions on Biomedical Engineering*, 46(7):861–866.

A time-varying Markov-switching regimes in a financial stress transmission. Evidence from Non-Eurozone Visegrad Group Countries

Magdalena Ulrichs

Department of Econometrics, Faculty of Economics and Sociology, University of Lodz, Poland
magdalena.ulrichs@uni.lodz.pl

Abstract. The main purpose of the article is to identify structural changes in the reactions of macroeconomic variables due to occurring financial stress episodes. In our empirical analysis, we use Marcov-switching vector autoregression models with time-varying transition probabilities. The main attention of our analysis is on the monetary policy and banking sector and its reactions on the financial stress shocks during two regimes: normal times and financial stress episodes. The obtained empirical results indicate that, during periods corresponding to the high levels of financial stress index, reactions of variables included in the system are stronger and more persistent than during periods of lower stress level.

Keywords: TV-MS-VAR models, financial shock transmission, regimes, financial stress index

1 Introduction

The global financial crisis has changed the relationships between the financial market and the macroeconomy. These kinds of changes can be empirically reflected in the structural changes of the parameters of econometric models. In the article, we follow the definition of the multiple partial structural changes model [4], [5], where the structural changes are reflected in the shifts in the subset of the model's parameters.

The empirical analysis is based on Non-Eurozone Visegrad Group Countries: the Czech Republic, Hungary, and Poland. These countries, during the analysed period, have experienced significant structural changes. Especially important were the conversions connected with entering the European Union, increasing level of trade openness, changes in the monetary and exchange rate policy, and lately the influence of the financial crisis. System transformation and structural changes were and are the main reasons why, in countries of transition, the market mechanisms of adjustments have not yet been fully established.

Hence the attempt to verify to what extent the shocks generated in the financial market are transferred to the real economy and whether they determine the further behaviour of financial markets, seems to be extremely important.

Examining the stability of financial and economic processes in case of an economy is not fully developed is a particularly important problem.

The importance of the impact of the instability of the global financial markets on the economy was widely underlined. What is crucial is that the literature focused not only on the influence of financial variables (e.g. stock-market volatility, exchange rates) on the economy, but also on the influence of variables that approximate the uncertainty and instability of financial markets ([6], [24], [8], [28], [34], [30], [1]).

This paper attempts to extend the existing empirical literature in several aspects. Firstly, we use the framework of Markov-switching vector autoregressive models with time-varying endogenous transition probabilities (TV-MS-VAR), which enables us to investigate the non-linear impact of financial stress on the macroeconomy and, moreover, to make transition probabilities of shifts between regimes conditional on an observable variable. The latter variable can be interpreted as a leading indicator of financial stress episodes. Secondly, we analyse, in a multivariate way, the impact of financial stress, which is empirically approximated within our model by published by European Central Bank the Country Level of Financial Stress Index. Thirdly, we compare empirical results obtained for the Non-Eurozone Visegrad Group Countries, what gives us the opportunity to analyse the influence of financial stress shocks on the banking sector, monetary policy and real sector for the small-open economies, which over the past few decades have experienced significant structural changes. In the case of the presence of structural changes in the studied relationships, a flexible estimation framework that accounts for the possibility of time variation is most appropriate. Fourthly, since in the case of analysed economies the financial system is considered to be bank-oriented, special attention is placed on the banking sector. In the model, we use a credit-to-GDP ratio as a proxy of credit cycle, which is considered to be strongly correlated with subsequent banking crises ([3], [11], [29]).

There are some empirical studies that are concerned with the influence of financial stress on the European economy, but they are usually concentrated on the Euro area aggregates. For example, [17] and [24] used, as a measure of financial stress, a Composite Index of Systematic Stress for the Euro area. Hartman et al. [17], using the MS-VAR models, have shown that the response of output to financial stress is much stronger when allowing for regime-switching. In turn, Kremer [24] have investigated the macroeconomic effects of financial stress based on the VAR model for the Euro area. Duprey [12], using univariate time-varying Markov-switching framework, investigated significant leading indicators affecting the probabilities of entering and exiting high financial stress regimes. The United States economy is analysed e.g. in [18]. They used a richly parametrized Markov-switching VAR model with constant transition probabilities, in which as a measure of financial stress the authors used the Financial Stress Index for the USA, constructed by the Federal Reserve Board.

As in [18], we argue that financial factors tend to be episodic in nature. The decision-making mechanisms during "normal times" are different than during

times when the financial system is not operating normally (these periods we will call "stress events"). The assumption that there are different regimes ruling the interdependency of the financial sector and the macroeconomy gives us the opportunity to answer - based on the nonlinear, multivariate framework - some questions related to whether or not the economy behaves differently across regimes. In particular, does the propagation of the financial stress shocks differ during stress and normal periods? Does the monetary policy react differently in cases of financial stress episodes? Is the bank system exposed to financial stress episodes? In order to answer these questions we estimate the TV-MS-VAR respectively for each analysed economy.

Markov-switching models are convenient tools for estimating models under the condition of not assuming *a priori* moments of structural change. They allow for inference on the probabilities of being in a specific regime, as well as on the probabilities of transition between states. The transition, in this case, can be modelled as a hidden Markov chain.

The structure of the paper is as follows. In section 2, we briefly describe the methodology based on the Markov-switching VAR models with endogenous transition probabilities. Basic features of the Financial Stress Index are reported in section 3. Section 4 presents empirical results, and discussion on how monetary policy, real economy and the banking sector react to changes in the Financial Stress Index. Finally, we present conclusions in Section 5.

2 Time-varying Markov-switching models

Standard vector autoregression models (VAR) enable the linear relationship between the vector $\mathbf{y}_t$ and their delays to be taken into account. In cases of approximating the processes by the linear models, it is assumed that responses to endogenous variable shocks are proportional to the shock values, that the reactions are symmetric regardless of whether the shock was positive or negative and that they do not depend on the moment of the impulse. In economic research, these assumptions are often unrealistic. Therefore, it is considered a model in which the endogenous variable $\mathbf{y}_t$ depends nonlinearly on its lags [20].

Markov-switching vector autoregressive models (MS-VAR) are a tool, that allows for taking into account the dependence of the magnitude of the reaction on the moment of the impulse. They constitute a convenient tool for estimating models under the conditions of not assuming *a priori* the moments of structural change. They also allow for estimating the probability of being in a specific regime, as well as, the probability of transition between states. The transition probabilities, in this case, can be modelled as hidden Markov chains.

Hamilton [16] used models with hidden Markov-chains for the business cycle analysis. He developed the Expectation-Maximization (EM) algorithm to the estimation of model parameters. Krozlig [25] generalized the univariate models for the multivariate analysis and described the EM algorithm for different data-generating processes. The identification of regime-dependent impulse response functions was developed in particular by [14], [26], and [27].

In standard Markov-switching models, time series dynamics are governed by a finite-dimensional parameter vector, which switches depending upon which of the states is realised, with state transition governed by a finite-order Markov process with constant transition probabilities. It is assumed that the data-generating process depends on the unobservable state variable s_t. The process $s_t \in \{1, ..., M\}$, where M is the number of states, is called Markov chain if the transition probability depends only on the previous state. s_t is assumed to be an ergodic Markov chain process with transition probability matrix $\mathbf{P} = [p_{ij}]$.

If the data-generating process is described by vector autoregression with hidden Markov chains (MS-VAR), then the parameters of this process depend on the observable vector of endogenous variables $\mathbf{y}_t$ and unobservable state variable s_t. The MS-VAR is described as in the form (1):

$$\mathbf{y}_t = \mathbf{A}_{0,s_t} + \mathbf{A}_{1,s_t}\mathbf{y}_{t-1} + ... + \mathbf{A}_{P,s_t}\mathbf{y}_{t-P} + \mathbf{u}_t, \tag{1}$$

where $\mathbf{y}_t$ is a K-dimensional vector of observable variables, $\mathbf{A}_{0,s_t}$ is a $K \times 1$ intercept vector, $\mathbf{A}_{p,s_t}$ is $K \times K$ slope coefficients, $\mathbf{u}_t \,|s_t \sim NID\,(\mathbf{0}, \Sigma\,(s_t))$ is a white noise with the covariance matrix depending on the state of Markov chain s_t, transition probability equals $p_{ij} = Pr(s_t = j \mid s_{t-1} = i)$.

As Diebold et al. [9] have pointed out, MS-VAR models incorporate a potentially severely binding constraint, particularly the constancy of state transition probabilities.

Agnello et al. [2] have observed that transition probabilities may depend on some observable indicators (e.g. leading indicators of the business cycle phases). Kim et al. [21] have noted that the values of macroeconomic shocks in the VAR model are often related to the state of the economy, e.g. the phase of the economic cycle, though this state is not usually observable. Agents can, however, make decisions based on the observation of some variables being an approximation of this state.

Therefore, it could be beneficial to relax the assumption on the time homogeneity of hidden Markov chains. If the transition probabilities are influenced by some observable variables, the data-generating process may be described by time-varying transition probability Markov-switching models - TV-MS [25, 32].

In cases of time-varying transition probabilities of Markov processes, it is assumed that these are conditional on some leading indicators that are considered to be good predictors of the fluctuations. So, if the decisions are based on the observation of the variable, which approximate the state of the system, then the transition probability depends on the vector of predetermined variables $\mathbf{z}_t$ and is given by:

$$Pr\,(s_t = j\,|s_{t-1} = i, \mathbf{z}_t\,) = p_{ij}(\mathbf{z}_t). \tag{2}$$

The time-varying transition probability Markov-switching VAR, i.e. TV-MS-VAR models, are in the form of (1) where transition probabilities p_{ij} depend on the vector $\mathbf{z}_t$ of observable variables and described as in eq. (2). The time-varying transition probabilities evolve as logistic function of vector $\mathbf{z}_t$, which contains economic variables that affect the state probabilities [25].

The estimation of MS-VAR models is usually done by maximum likelihood or Bayesian methods [35]. In the case of a maximum likelihood estimator, the estimator values must be found numerically. Krolzig [25] discusses the EM algorithms that simplify this optimization task for a range of important special cases of model specifications. Diebold et al. [9] showed that the EM algorithm is a stable and robust procedure for maximizing the incomplete-data log likelihood via iterative maximization of the expected complete-data log likelihood - conditional upon the observed data.

3 Financial stress and economic dynamics

The course of financial crisis emphasized the need to measure financial stress. In the empirical investigations, different approximations of unobservable stress in the financial markets are used. These indices are commonly known as stress or uncertainty indices. A broad review of financial stress indices was made by e.g. Kliesen [23]. The main purpose of stress indices is to measure instability and the resulting uncertainty about the behaviour of financial markets. It is usually assumed that they should reflect situations where financial stress materializes simultaneously in various market segments. The discussed indicators are most often approximated either by means of a volatility measure or by using a synthetic index of unobservable conditions on financial markets. In financial markets, implied share-returns volatility is one of the canonical measures for uncertainty [6].

In this paper, we refer to the financial stress literature and use as a measure of financial stress the Country-Level Index of Financial Stress (CLIFS).

Duprey et al. [13] have defined financial stress as simultaneous financial market turmoil across a wide range of assets (equity markets, government bonds and foreign exchange), reflected by the uncertainty in market prices, sharp corrections in market prices and the degree of commonality across asset classes. The CLIFS corresponds to that definition and allows for the identification of systematic financial stress episodes, i.e. periods of high financial stress that are associated with a substantial and prolonged decline in real economic activity. The essential feature of a country-specific index of financial stress is that it captures co-movements in main financial market segments.

Fig. 1. The level of CLIFS for the Czech Republic, Hungary and Poland. Source: ECB Statistical Data Warehouse.

The levels of CLIFS for the Czech Republic, Hungary and Poland are depicted in Fig. 1. During the analysed period, we can observe that the highest values of the index in each county occurred in the first quarter of 2009, which can be linked to the influence of global financial crisis. Moreover, in the Czech Republic, there are also high values of the index at the beginning of the sample (1999 and 2002), for Hungary we can observe one more dominant level at the end of 2011, while in the case of Poland there were two more dominant levels: 2001 and 2011. These periods correspond to the periods of highest volatility in the segments of the financial market.

4 The financial stress transmission - empirical results

The empirical analysis is concerned with the comparison of financial stress shocks transmission in the Czech Republic, Hungary and Poland. The chosen economies are interesting examples of developed countries, that were influenced by significant structural changes since the transformation period. All of them are small-open economies exposed to the influence of global shocks (e.g. financial shocks). We only analyse Non-Eurozone Visegrad Group Countries and do not compare the results with Monetary Union countries. In the analysed countries, the level of development experienced by their financial systems and monetary policy strategies are comparable.

In the analysed economies, among all the sectors of the financial system, the banking sector has the strongest influence on the economy. In all of them, the development of financial intermediation exhibits a relatively low level compared when with the ratio's average value in the Euro area countries.

Taking under consideration the relation of financial system assets to GDP, the financial system in the Czech Republic is the most developed among analysed countries (in 2016, it accounts for almost 160% of GDP). However, in analysed countries, the financial system is not excessively developed in relation to the real economy when compared to the Euro area countries.

In all countries of the region, the dominant source of financing the real sector is banking. However, among analysed countries, Hungary's and Poland's financial systems can be regarded as two of the least bank-oriented, while in the Czech republic the ratio of banking sector assets to GDP is much higher.

In CEE countries, the level of banking sector development remained low compared to Euro are countries. Domestic banks focused on providing traditional banking services, mainly on deposit-taking from and landing to non-financial clients. The banking sector assets in Hungary and Poland have exceeded 90% of GDP since 2013, which is a relatively low ratio among European Union countries. The largest item among bank's assets, were loans (in particular housing loans to households), which in the analysed period accounted for more than 60% of total bank assets. Among European Union countries, Poland has the highest level of loans ratio to bank assets.

Two specific financial system features in the analysed countries are the relatively low stock market capitalisation and the low value of outstanding private

sector debt securities, including corporate and bank bonds. However, compared with other countries in the region, the Polish stock market plays a significant role. The ratio of capitalisation of domestic companies noted on the Warsaw Stock Exchange to GDP was around 30% in the analysed period, whereas in the Czech Republic and Hungary it did not exceed 20%. In the Euro area, in 2016, the capitalisation of domestic companies was around 66% of GDP.

The main conclusion is that, in Non-Eurozone Visegrad Group Countries, the banking sector is the most significant source of real sector financing, while stock market financing is not highly developed. The structure of the financial market means that these economies may not be subject so highly to financial distortions coming from the capital market, though banking sector resistance to financial stress should be carefully examined.

4.1 The model specification

The models parameters were estimated by use of the quarterly data for the period 1997-2018. The vector of the model's endogenous variables is represented by:

$$\mathbf{y}_t = [CREDIT_t, GDP_t, R_t, CLIFS_t] \tag{3}$$

and includes the following stationary and seasonally-adjusted time series: $CLIFS$ - Country Level of Financial Stress, R - changes in the nominal 3-month money market interest rate (p.p.), $CREDIT$ - the credit-to-GDP ratio measured as changes in the ratio of total loans and advances from the banking sector to the non-financial sector in domestic and foreign currency to the GDP (p.p.), GDP - dynamics of real gross domestic product (%).

Market agents make decisions based on observable changes in the equity markets (stock exchanges indices), the foreign exchange market and also bond markets. Therefore, we can assume that the transition probabilities may depend on the leading indicator variable, which is in this case represented by lagged value of $CLIFS$ index: $\mathbf{z}_t = [CLIFS_{t-1}]$.

The credit-to-GDP gap is one of the most commonly used single indicators for judging the increasing level of financial vulnerabilities, which may result in an occurring of financial crisis. As e.g. [11] pointed out that the credit-to-GDP gap is, despite its many drawbacks, one of the most informative indicators of future changes to the financial market. The focus on credit is justified by empirical analysis that strong credit growth has typically proceeded crises [22].

Furthermore, in reference to the literature, the monetary policy is represented in the model by changes in the money market rate, which are used to approximate key policy rates ([6], [7]). The economic activity is measured by the dynamics of real gross domestic product (as in [22] and [29]).

It should be stressed that the ML estimator, in cases of a highly parametrised MS-VAR model, may be unreliable or impossible to use in practice, especially regarding small samples [20]. Therefore, we consider a partial structural change model ([4], [5]) and assume that only parameters capturing the influence of $CLIFS$ on other system variables, autoregressive parameters and constants are

regime-dependent. A partial change model is useful for allowing potential savings in the number of degrees of freedom.

4.2 Empirical results

During the analysed period two states (regimes) of the economy were identified[1]. In Fig. 2 the smoothed state probabilities and time-varying transition probabilities for each country are demonstrated.

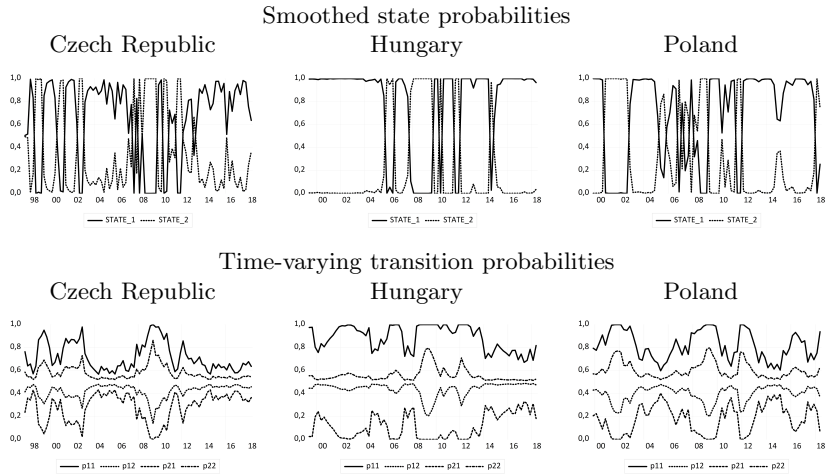


Fig. 2. Smoothed state probabilities

The first state corresponds to the "normal" periods and the second to the stress episodes. During stress episodes, the stress level and variance of shocks are higher. The high probabilities of being in the second state correspond to the high levels of the $CLIFS$ index in each country.

For each regime, as in [14], regime-dependent impulse response functions were estimated under the assumption that, within each regime, responses are derived exactly as if the system was a VAR model with $\mathbf{A}_{1,s_t}$ matrix of autoregressive parameters and the structural identification of shocks does not depend on the regime.

The general model contains regime-dependent impulse response functions corresponding to the expected reactions of endogenous variables in period $t+h$ to the one standard deviation macroeconomic shock in the period t and they are conditional on the regime s.

[1] For estimations, we used the EM algorithms implemented in the Matlab environment by [33] and [10].

The structural shocks are identified by applying the recursive Cholesky decomposition with the conventional ordering of real macro variables before financial and monetary policy variables with the following order of variables $CREDIT_t$, GDP_t, R_t, $CLIFS_t$. This Cholesky ordering implies that only the Financial Stress Index reacts simultaneously to macroeconomic shocks. Other variables respond to the financial stress shock in subsequent periods[2].

In Fig. 3 the responses to one standard deviation financial stress shock are presented. The first-state regime-depended impulse response functions indicate that, during "normal" times, monetary policy does not react strongly to the financial stress shock but, during the time when the financial market is characterized by significant tensions, the monetary policy reaction is highly expansive. The interest rate reaction during the second state is even higher than the statistically significant reaction in the linear VAR model. The reaction of macroeconomic variables indicates a strong decrease in economic activity and the credit-to-GDP ratio. The reactions of all variables during normal times indicate that, when there is no high pressure on the financial market, then the real economy, monetary policy and the banking sector do not react strongly - only during financial stress episodes are the reactions notable. The results are comparable for all analysed economies.

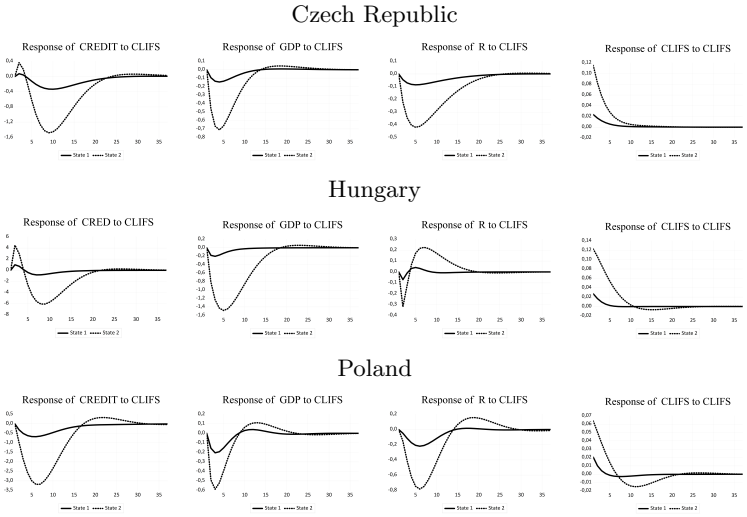


Fig. 3. Regime-dependent impulse response functions - Response to Cholesky One S.D. CLIFS Innovations

[2] Our identification strategy is similar to that in [17], [24], [19].

Conclusions

The aim of this paper is to provide empirical investigation concerning the bilateral relationships between financial market stress and the banking sector, as well as monetary policy and real economy. The focus of our study was placed on the analysis of developing small-open economies, without fully established financial adjustment mechanisms but which can be subject to structural changes. We bridge the gap between existing literature which is focused mainly on an analysis of developed economies (the USA, Euro area). We also take into account non-linear dependencies between financial market shocks and the macroeconomy variables, as well as allowing for the transition probabilities between the regimes being conditional on leading indicator variables changing over time.

The obtained estimates allow for the following conclusions to be formulated: during the analysed period two regimes were identified; the first state corresponded to the normal time and the second to financial stress episodes. Regime-dependent impulse response functions indicate that, during financial stress episodes, financial stress shock has a higher and more persistent impact on monetary policy, economic activity and the stability of the banking system. While during "normal" times, reactions of variables included in the model are rather negligible. Moreover, monetary policy should react adjustably to the value of financial stress shocks and, even in cases of high inflation pressure, the increase of interest rates during stress episodes should be less than it could be during normal times to avoid increases in the Financial Stress Index.

Our research shows that during a global and very rapid influence of financial market distortions on macroeconomy, especially in cases of small-open economies, advice based on solid empirical studies can be useful for policymakers. Our study is, nevertheless, not free from conclusion limitations. First, the model used by us did not include all important factors that could potentially influence the analysed relationships, it was estimated on a relatively short time series. However, the results were subject to numerous robustness checks. First of all, different approximations of bank sector self-funding ability were used: apart from the credits-to-GDP ratio also the total-deposits-to-total-assets ratio, a deposits-to-loans ratio was used. Overall results did not change significantly. Furthermore, different Cholesky decomposition ordering and other structural decomposition restrictions were also examined. Since the simultaneous correlations between all variables were not statistically significant, the results are resistant to the decomposition scheme, e.g. placing the real and monetary variables before the Financial Stress Index (like in [6], [7], [8], [30]), did not change the overall results. It was also found that including other important variables, e.g. inflation rate or unemployment rate, probably due to a relatively short time series and richly parametrised Markov-switching VAR models, did not improve the results.

References

1. Adam, T., Benecká, S., and Matěj, J. (2018). Financial stress and its non-linear impact on CEE exchange rates. *Journal of Financial Stability*, 36:346–360.

2. Agnello, L., Dufrénot, G., and Sousa, R. M. (2013). Using time-varying transition probabilities in Markov switching processes to adjust US fiscal policy for asset prices. *Economic Modelling*, 34:25–36.
3. Aikman, D., Haldane, A. G., and Nelson, B. (2014). Curbing the credit cycle. *The Economic Journal*, 125:1072–1109.
4. Andrews, D. D. K. and Fair, R. C. (1988). Inference in Nonlinear Econometric Models with Structural Change. *Review of Economic Studies*, 55(4):615–639.
5. Bai, J. and Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, 18(1):1–22.
6. Bloom, N. (2009). The Impact of Uncertainty Shocks. *Econometrica*, 77(3):623–685.
7. Bonciani, D. and van Roye, B. (2016). Uncertainty shocks, banking frictions and economic activity. *Journal of Economic Dynamics and Control*, 73:200–219.
8. Caldara, D., Fuentes-Albero, C., Gilchrist, S., and Zakrajšek, E. (2016). The macroeconomic impact of financial and uncertainty shocks. *European Economic Review*, 88:185–207.
9. Diebold, F. X., Lee, J.-H., and Weinbach, G. C. (1994). Regime switching with time-varying transition probabilities. *Nonstationary Time Series Analysis and Cointegration*, pages 283–302.
10. Ding, Z. (2012). An Implementation of Markov Regime Switching Model with Time Varying Transition Probabilities in Matlab. Technical report.
11. Drehmann, M. and Tsatsaronis, K. (2014). The credit-to-GDP gap and countercyclical capital buffers: questions and answers. *BIS Quarterly Review*, March:55–73.
12. Duprey, T. and Klaus, B. (2017). How to predict financial stress? An assessment of Markov switching models. *ECB Working Paper Series*, (2057).
13. Duprey, T., Klaus, B., and Peltonen, T. (2015). Dating systemic financial stress episodes in the EU countries. *ECB Working Paper Series*, 1873.
14. Ehrmann, M., Ellison, M., and Valla, N. (2003). Regime-dependent impulse response functions in a Markov-switching vector autoregression model. *Economics Letters*, 78:295–299.
15. Frederic S. Mishkin (2007). *The economics of money, banking, and financial markets*. Pearson Education.
16. Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.
17. Hartmann, P., Hubrich, K., Kremer, M., and Tetlow, R. J. (2014). Melting Down: Systemic Financial Instability and the Macroeconomy. *European Central Bank (ECB)*, pages 1–42.
18. Hubrich, K. and Tetlow, R. J. (2015). Financial stress and economic dynamics: The transmission of crises. *Journal of Monetary Economics*, 70:100–115.
19. Kanngiesser, D., Martin, R., Maurin, L., and Moccero, D. (2017). Estimating the impact of shocks to bank capital in the euro area. *ECB Working Paper Series*, 2077.
20. Kilian, L. and Lütkepohl, H. (2017). *Structural Vector Autoregressive Analysis*. Cambridge University Press.
21. Kim, C. J., Piger, J., and Startz, R. (2008). Estimation of Markov regime-switching regression models with endogenous switching. *Journal of Econometrics*, 143(2):263–273.
22. Kim, S. and Mehrotra, A. (2018). Effects of Monetary and Macroprudential PoliciesEvidence from Four Inflation Targeting Economies. *Journal of Money, Credit and Banking*, 50(5):967–992.
23. Kliesen, K. L., Owyang, M. T., and Vermann, K. E. (2012). Disentangling diverse measures: A survey of financial stress indexes. *Federal Reserve Bank of St. Louis Review*, 94(5):369–398.

24. Kremer, M. (2015). Macroeconomic effects of financial stress and the role of monetary policy: a VAR analysis for the euro area. *International Economics and Economic Policy*, 13(1):105–138.
25. Krolzig, H.-M. (1997). Markov-Switching Vector Autoregressions. Modelling, Statistical Interference, and Application to Business Cycle Analysis. *Lecture Notes in Economic and Mathematical Systems*, 454.
26. Krolzig, H.-M. (2006). Impulse-Response Analysis in Markov Switching Vector Autoregressive Models. *Working Paper*, 1(2):1–17.
27. Lanne, M. and Lütkepohl, H. (2008). Identifying monetary policy shocks via changes in volatility. *Journal of Money, Credit and Banking*, 40(6):1131–1149.
28. Leduc, S. and Liu, Z. (2016). Uncertainty shocks are aggregate demand shocks. *Journal of Monetary Economics*, 82:20–35.
29. Malovaná, S. and Frait, J. (2017). Monetary policy and macroprudential policy: Rivals or teammates? *Journal of Financial Stability*, 32:1–16.
30. Meinen, P. and Roehe, O. (2017). On measuring uncertainty and its impact on investment: Cross-country evidence from the euro area. *European Economic Review*, 92(February 2016):161–179.
31. Mishkin, F. S., Matthews, K., and Giuliodori, M. (2013). *The Economics of Money, Banking, and Financial Markets*. Pearson Education.
32. Perez-Quiros, G. and Timmermann, A. (2000). Firm Size and Cyclical Variations in Stock Returns. *The Journal of Finance*, 55(3):1229–1262.
33. Perlin, M. (2010). MS_Regress - The MATLAB Package for Markov Regime Switching Models. *SSRN Electronic Journal*, pages 1–38.
34. Popp, A. and Zhang, F. (2016). The Macroeconomic Effects of Uncertainty Shocks: The Role of the Financial Channel. *Journal of Economic Dynamics and Control*, 69:319–349.
35. Sims, C. A., Waggoner, D. F., and Zha, T. (2008). Methods for inference in large multiple-equation Markov-switching models. *Journal of Econometrics*, 146(2):255–274.

PREPARATION OF TRAINING DATA BY FILLING IN MISSING VESSEL TYPE DATA USING DEEP MULTI-STACKED LSTM NEURAL NETWORK FOR ABNORMAL MARINE TRANSPORT EVALUATION

Julius Venskus and Povilas Treigys
Vilniaus Universitetas, Klaipedos Universitetas (Lithuania)

Abstract

Highly-loaded seaports have extremely complex and intensive marine vessel traffic, which generates large volumes of traffic data. Meteorological conditions and maritime vessel type influence maritime traffic and they must also be taken into account in order to train the model capable of recognizing the abnormal movement of the sea transport. Real data often misses some data values, such as type of vessel or its status.

Data that influences maritime traffic must also be taken into account in order to train the model capable of recognizing the abnormal movement of the sea transport. Typically, in every region missing data differs.

This paper reviews methods of obtaining vessel traffic and meteorological data and filling missing vessel type data in regions such as Rotterdam, Gdansk, Malmö, Copenhagen, Kiel and Lübeck.

Author describes possibilities to obtain technical parameters of ships and existing limitations while taking into account the available vessel data. By assessing the characteristics of the received traffic data, the regionalization of the meteorological data and the assignment to a specific maritime traffic data vector is performed by using the method of the closest neighbor, depending on the time of traffic in a given region.

A deep multi-stacked LSTM neural network model is trained to fill the missing vessel type data. This model is trained with available vessel type data and used to predict missing values. This paper describes creation and evaluation of this model.

Calendar based forecast of emergency department visits

Cosimo Lovecchio[a], Mauro Tucci[a], Sami Barmada[a], Andrea Serafini[b], Luigi Bechi[b], Mauro Breggia[b], Simona Dei[b], and Daniela Matarrese[c]

[a]Department of Energy, Systems, Territory and Constructions Engineering, University of Pisa, Italy.
[b]AUSL Tuscany South-East, Italy.
[c]AUSL Tuscany Center, Italy.
cosimo.lovecchio@ing.unipi.it

Abstract. In this paper we use a well established method for short-term forecasting to predict the amount of hourly Emergency Department (ED) visits in thirteen different hospitals in the south-east area of Tuscany. Our algorithm belongs to the class of similar shape algorithms and perform the forecast in an unsupervised manner. It exploit an historical dataset containing the patient arrival data, in which similar pattern, filtered on the base of a calendar condition, are selected to predict the incoming visit volume for a tunable number of day ahead.

Keywords: Emergency department, hospital, forecasting, time-series prediction.

1 Introduction

Overcrowding of Emergency Department (ED) is defined as "the situation in which ED function is in a difficult situation primarily because of the excessive number of patients waiting to be taken in charge, undergoing assessment and treatment, or waiting for departure compared to the capacity of the ED"[1]. This must not be confused with major emergencies that are due to clearly different causes, and requires different solutions. Overcrowding is a condition that is strongly associated with the risk of impairment of the quality of care provided: latency in taking charge, delay in carrying out diagnostic tests and in starting treatment, increase in errors and adverse events [2]. According to the Joint Commission on the Accreditation of Healthcare Organizations, one third of sentinel events in the EDs are caused by an overcome of the ED capacities. Overcrowding in the EDs leads to many negative consequences, such as an increase in mortality [3, 4], negative perception by patients [5–7] often resulting from prolonged stay on stretchers without privacy or adequate responses to basic needs, and a higher probability of ED staff "burn-out", that causes a further loss of efficiency and a worsening of the shelters filter function with an increase in overall hospitalization times. It is a widespread problem that has been addressed in recent years with targeted interventions in several countries with universal access

health systems, such as United Kingdom, Canada, Australia and New Zealand [8–12].Trends of ED visits are quite predictable throughout the year and during the different moments of the day, based on seasonal epidemiology and circadian distribution of accesses. The correct management of this trends allows to avoid critical situations, in particular during periods of influenza epidemics [13]. Several factors have been recognized, often acting simultaneously, whether at the presenting of the patient at the ED ("input" factors), along the internal path to the PS ("throughput") or at the patient discharge/transfer ("output" factors). Input factors refer to the numerous ED visits mainly due to seasonal epidemiology, while throughput factors indicate the length of the patient's stay in ED. Finally, output factors are influenced by the difficulty of hospitalization, due to lack of available beds and the difficulty of discharge, especially for patients with social problems. It has been widely demonstrated that throughput and output factors contribute the most to the system overload and, unlike input factors, can be significantly modified by adopting appropriate organizing strategies [14]. The overcrowding of ED depends on two factors: - Crowding: the critical increase in both the admissions and permanence within ED of patients who are completing the diagnostic-therapeutic process; - Boarding: the accumulation in ED of patients who have already completed the care process but who, for various reasons, cannot be discharged from ED [15].

1.1 Crowding

The analysis of the level of ED crowding is mainly addressed to 2 areas: the access phase (how many patients arrive, how, by whom, at what time of day, etc.) and the "process" phase, i.e. the whole clinical and therapeutic path within ED. In Tuscany, the analysis of data on the trend of time bands, especially with reference to color codes, confirms an inappropriate use of ED instead of other settings (70-75% 8-20 vs 30% 20-8, usually ¡10% of the admissions of 24-8) [16]. In the population there is the belief ED is the starting point of many of the diagnostic-therapeutic pathways 'subjectively' considered urgent, while family doctors are considered for the continuation of the pathway and follow-up. It is necessary to redistribute the inappropriate share of demand through an intervention strategy that crosses several treatment processes. Another contribution to crowding is represented by people affected by chronic diseases, already followed by other services both at the local and hospital level, which experience a high percentage of repeated admissions for the same disease (heart failure, complicated diabetes, etc.). Investigation of these patients involves repetitions of laboratory and instrumental tests that unnecessarily absorb a large number of resources, and which would not be necessary if the patients had addressed the doctors who treat them. Countermeasures to the phenomenon of crowding include the redistribution of tasks within the assigned staff, the activation of available staff, and the detention in service of "disassembly" staff. This also applies when crowding has been largely generated by boarding, which absorbs time and staff work, contributing to the progressive increase in waiting time. In

this case, the actions must be supplemented by those necessary for the proper management of boarding [16, 17].

1.2 Boarding

The accumulation in ED of patients who have already completed the care process, is largely due to the waiting for the bed, mainly in the medical area. These are mainly elderly people with comorbidities with high absorption of resources who remain for a long time in unsuitable environments. In many cases, the demand for hospitalization is generated by the hospital facilities themselves, where these patients are already being treated, reaching, in some cases, about 10-15% of hospitalizations. In addition, there are chronic patients with repeated hospitalizations for the same disease (heart failure, COPD, complicated diabetes, etc.), mainly intended for the medical area. The clinical evolution of these patients is in many cases gradual and progressive and this could have allowed the organization of hospitalization, when appropriate, without the need to access the ED that, in fact, becomes only the place of waiting for the bed. The number of these admissions can also represent 20-25% of admissions in the medical area and often involves more admissions during the year, always through the ED. This "avoidable boarding" is about 30 and 40% of the phenomenon. To solve the problem of boarding, the whole hospital must work together to ensure the balance between supply and demand at various stages of the treatment process. For this reason it is necessary to effectively manage the flows of incoming and outgoing patients, to optimize the emergency and planned routes and to make more efficient use of the hospital beds [16].

1.3 Forecast Motivation and Methodology

If on one hand the internal queues and patient flow management is a crucial aspect to consider in order to reduce overcrowding, improve the quality of service and reduce operating costs, on the other by an accurate forecast of the ED services demand enable proper planning of the clinical resources amount to activate. Nevertheless the identification of a feasible forecast tool rises some challenges. A first aspect to consider is the quality and quantity of historical informations about a specific scenario. One might be tempted to claim that collecting a large amount of data describing the present, we could be able to predict the immediate future visits volume. Actually, it often turns out that the most accurate source of information which can be used to predict the future behaviour of a physical quantity, it is the past behaviour of the quantity itself. Another important aspect to undertake is the selection of a predictive model. Once the ED patient volumes in a sufficiently long time window has been collected, this type of forecast can be enclosed in the time series prediction framework, one of the most transversal research topic. In fact, a plethora of analytical tools are available to describe temporal dynamics, ranging from classical statistical models [18–25], to more recent artificial intelligence based algorithm. Each of these tools has features which make it more suitable or reliable than the others in a

particular application. In many time series forecasting problems, where human and social activities are predicted, environmental factors affect the resulting collective behavior to a different extent, but among all the external source of influence, calendar patterns play a crucial role [26]. It is well known how, for different calendar day type (working day, holidays, special holidays), different human dynamics (e.g. shopping behavior of buyers, traffic patterns, crowding effects in places of entertainment etc.) can be observed. In this work we use a variation of a popular and well established time series forecasting model belonging to the class of similar shape algorithms (K-nearest neighbors, or knn), to predict several days-ahead hourly patient volumes in 13 ED facilities of a local health centre of Tuscany. Our model (C-knn in the following) includes a control mechanism on the calendar condition for the prediction provided. We evaluate the forecast accuracy by means of two performance indicator, the mean absolute percentage error, MAPE, and variance of absolute percentage error, VAPE, estimators.

2 Model

We apply our model to a dataset of aggregated informations, extracted from the accesses records in the EDs facilities. The facilities analyzed have different characteristics, such as size, services provided (depending on the hospital equipments), or dimension of area served. Each record of the databases extracted from the servers, contain all the data related to a single ED acces, such as date and time of admission, priority code (color code), ED infrastructure, age, sex and other specific informations. Among these we focus on date, time and color code, calculating the aggregated time series (Fig. 1). The data cover a time window starting on 2014-01-01, and ending on 2018-11-14. Performance analysis of our algorithm was carried out by exploiting the last available year of information (test set), while the remaining data (train set) were used to fit the model metaparameters.

The C-knn algorithm belongs to the class of similar day-based or similar shape methods [26]. The key idea consists in the research, between the available data, for historical days that are characterised by intraday dynamics similar to the recent past (e.g. similar average, maximum values or peaks positions) in order to predict the near future. For a more reliable prediction the set of days in which the search is performed can be bounded by constraints. In particular our algorithm automatically finds the similar profiles in the available data-base, selecting only those whose weekday sequence exactly match the actual one. Let us show in details how the algorithm works. Suppose we are in the day $d_0 \in \mathbb{R}^{24}$, and our goal is the prediction of the hourly accesses in the future N days $f_N = \{d_1, ..., d_N\} \in \mathbb{R}^{24\times N}$. To calculate the prediction we exploit the historical database, assuming that the data covers the accesses history until d0. The steps performed by the C-knn algorithm are:

1. Pick out from the database the accesses profile of the consecutive most recent M days $a_M = \{d_{-M+1}, ..., d_0\} \in \mathbb{R}^{24\times M}$, and subtract from it its mean value

$a_M(0) = a_M - \langle a_M \rangle$. The number of days M to select is a metaparameter of the model.

2. Take all the possible sub-series of M+N consecutive days in the historical database $p(i) = \{d_{i-M+1}, , d_i, ..., d_{i+N}\} \in \mathbb{R}^{24\times(M+N)}$, $i = -N, -N-1, ...$ For each series $p(i)$, the first M days portion will be denoted as $p_M(i) \in \mathbb{R}^{24\times M}$, the last N days as $p_N(i) \in \mathbb{R}^{24\times N}$.
3. Discard from the set $p(i)|_{i=-N,...}$ those elements whose calendar condition on the $p_N(i)$ part is different from the f_N one. We will clarify in the following the "calendar condition" meaning. The remaining bounded set will be used for the reconstruction.
4. Calculate the zero mean profiles $p_M^{(0)}(i) = p_M(i) - \langle p_M(i) \rangle$.
5. Calculate the weighted distances $d(i) = ||p_M^{(0)}(i) - a_M^{(0)}||_{W^2} = ||W \cdot (p_M^{(0)}(i) - a_M^{(0)})||$ where $W \in \mathbb{R}^{24\times M\times M}$ is a weight vector giving different importance to different hours. The coefficient in W are also metaparameters of the model.
6. Select the k most similar $p_M(i)$ (i.e. those whose corresponding $d(i)$ is minimum) and the related $p_N(i)$. The value of k is another metaparameter of the model. We will denote the chosen $p_M(i)|_{i=i_1,...,i_k}$ as the "best profiles" $b_M(j)|_{j=1,..,k}$, and the related $p_N(i)|_{i=i1,...,ik}$ as "best candidates" $c_N(j)|_{j=1,..,k}$.
7. Compute the similarity scores sj by the gaussian kernel $s_j = e^{-d^2(j)/\sigma^2}, j = 1, ..., k.$ The kernel width value σ in the equation above is defined as proportional to the smallest distance $d(j)$, $\sigma = \lambda\, min\{d_j\}$, where λ is a positive constant to be optimised.
8. Reconstruct the recent past a_M by using the similarity scores and the best profiles $\overline{a_M} = \sum_j s_j b_M(j)$ and look for the best scaling factor α^* which minimize the distance between the recent past and the reconstructed one, i.e. $\alpha^* = argmin_\alpha ||\alpha\, \overline{a_M} - a_M||$.
9. The final forecast is finally given by the scaled weighted sum of the best candidates $f_N^* = \alpha^* \sum_j s_j c_N(j)$

2.1 Imposing Calendar Conditions

ED patient volume strongly depends on calendar variables. In addition to seasonal trends, special days or events occurring during the year appear as anomalies compared to other days, as shown in Fig. 1. Volume is in average lower on national holidays and on Sunday, while appear higher on Monday, as can be noted in Fig. 2. Therefore, a day parametrization mapping the calendar pattern turn out to improve the quality of the forecast. In particular, we divide the week days in the three following classes:

1. Working Days: days from Monday to Friday, excluding special holidays.
2. Saturdays: all Saturdays excluding holidays.

3. Holidays: all Sundays and special holidays (Easter Monday, Christmas, New Years Day, etc.).

The third step of the algorithm listed above consists in the elimination of those sequences whose future calendar condition does not match with the actual future we aim to predict.

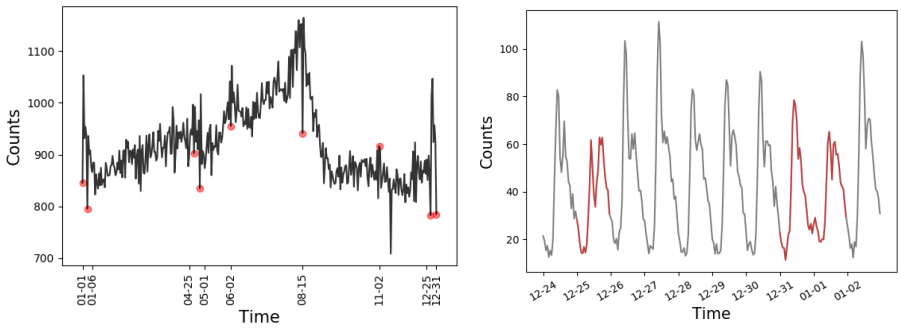


Fig. 1. Week-day dependence of the total patient visits in the 13 ED facilities under study. On the left side: mean daily patient volume during the year (The average is computed on 4 years). Red points highlight some of the national holidays, namely New Years day, Epiphany, Liberation day, May Day, Italian Republic Holiday, Assumption day, All Souls' Day, Christmas, new Year's Eve. On the right side: mean hourly visits amount during Christmas holidays. Red line highlight Christmas day, new Year's Eve and New Years day.

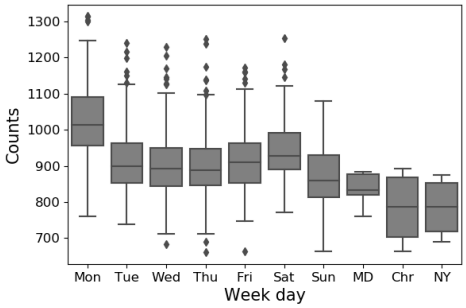


Fig. 2. Boxplot of the daily total accesses during weekdays and some of the national holidays (MD May day, Chr Christmas, NY New Year's day).

3 Model Evaluation

The goodness of the model was assessed evaluating two performance indexes: mean absolute error (MAE), and variance of the absolute Error (VAE). Given a time series $y(n), n \in [1, ..., N]$ and its reconstruction $\overline{y(n)}$, MAE is defined as:

$$\mathrm{MAE} = \frac{1}{N} \sum_{n=1}^{N} \left| y(n) - \overline{y(n)} \right| \tag{1}$$

and VAE as:

$$\mathrm{VAE} = \frac{1}{N} \sum_{n=1}^{N} \left(\left| y(n) - \overline{y(n)} \right| - \mathrm{MAE} \right)^2 \tag{2}$$

The first index reflect the model accuracy, while the second one is a measure of the model stability.

The model parameters which can be optimized are:

1. The number of days in the past M to compare with the historical database.
2. The weight vector W filtering the time sequences.
3. The number of most similar patterns k.
4. The size of the kernel function.

To downsize the computational effort in the parameters tuning we assume W to be diagonal with linearly increasing coefficients, reducing its degree of freedom to only the initial and final values, and $\lambda = 1$. The performance indexes landscape was then obtained by Grid Search over a suitable parameters space, uniformly sampled. In particular, for every parameters combination $\overline{p} = \{M, k\}$, we simulated a true forecast using an incrementally expanding historical set, which was performed iteratively on the last available year accesses. After the forecasts production, $\mathrm{MAE}_i(\overline{p})$ and $\mathrm{VAE}_i(\overline{p})$, $i \in [1, ...13]$, were calculated for each ED facility, for a prediction horizon of 1 day-ahead.

To set a convenient metaparameters combination we finally calculated the total MAE and VAE as $\mathrm{MAE}_T(\overline{p}) = \sum_i \mathrm{MAE}_i(\overline{p})$, $\mathrm{VAE}_T(\overline{p}) = \sum_i \mathrm{VAE}_i(\overline{p})$. A density plot of these two quantities against the M and k is shown in Fig. 3. As can be noted the algorithm performances monotonically increases for both increasing M and k, reaching a plateau region for approximately $M \gtrapprox 25$, $k \gtrapprox 6$. Moreover, MAE and VAE are linearly correlated for the set of parameter explored.

The final setting we adopted to perform the forecast thus are $M = 6$, $k = 32$.

4 Results and Discussion

The resulting scores calculated in correspondence of the selected metaparameters M and k, and for all the single ED structures, are summarized in Table 1. In particular we report MAE and VAE on a daily and hourly aggregated timescale. We also report the Mean Bias Error (MBE), defined as

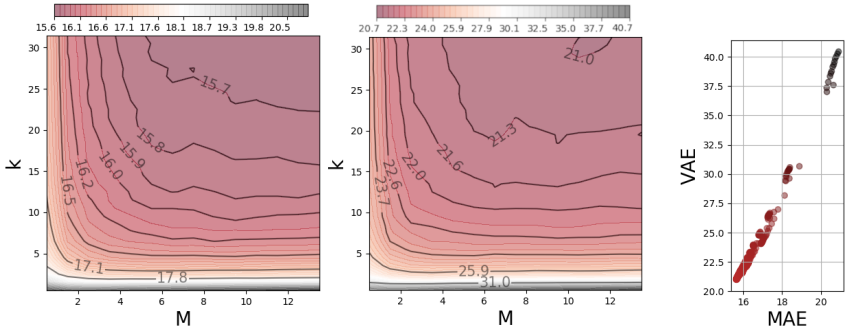


Fig. 3. Density plot of MAE_T (left side) and VAE_T (center) as a function of the optimization parameters M (number of comparison days in the past), and k (number of nearest neighbours). As can be noted, no improvements of the total scores are appreciable for $M \gtrapprox 25$, $k \gtrapprox 6$. MAE dependence of VAE (left side) in the parameters grid explored. As can be noted, there is an almost linear dependence between the two quantities.

$$\mathrm{MBE} = \frac{1}{N}\sum_{n=1}^{N} y(n) - \overline{y(n)}. \tag{3}$$

The MBE quantifies how the model is biased compared to the true time series.

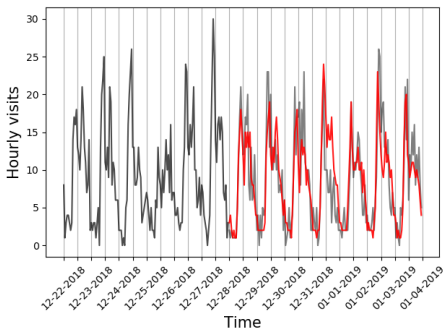


Fig. 4. Incoming visits forecast in correspondence of the new Year's Eve week. Black solid line represent the data portion used to look for similar patterns in the historical dataset, ligth grey the true data and red line their forecast.

In Fig. 4 we show a forecast example of the accesses for the ED 3 in Table 1, in correspondence of the new Year's Eve week. As can be observed, the predicted

ED	Mean daily accesses	MAE(d)	MBE(d)	VAE(d)	Mean hourly accesses	MAE(h)	MBE(h)	VAE(h)
1	38.65	8.3	1.02	44.27	1.61	1.03	0.04	1.06
2	39.72	7.43	-1.45	36.06	1.65	0.95	-0.06	1.06
3	203.8	17.35	1.23	214.68	8.49	2.32	0.05	4.49
4	104.08	13.09	-2.23	111.4	4.34	1.62	-0.09	2.15
5	36.05	7.12	-0.07	30.78	1.5	0.91	0	0.88
6	17.42	5.03	-0.64	19.8	0.73	0.56	-0.03	0.49
7	75.83	11.47	-3.18	157.41	3.16	1.35	-0.13	1.66
8	75.1	11.05	-2.12	141.3	3.13	1.23	-0.09	1.42
9	39.51	7.55	-2.97	39.6	1.65	0.93	-0.12	0.95
10	67.71	10.5	-2.13	82.3	2.82	1.31	-0.09	1.73
11	14.82	4.55	-0.26	14.4	0.62	0.54	-0.01	0.44
12	21.69	5.53	-1.18	22.82	0.9	0.68	-0.05	0.58
13	186.71	19.88	-0.77	251.57	7.78	2.25	-0.03	4.14

Table 1. Table of resulting forecast MAE, VAE and MBE, calculated along the last one year of data, aggregated on daily (MAE(d), VAE(d) and MBE(d)), and hourly (MAE(h), VAE(h) and MBE(h)) timescale.

data (solid red line) adequately resembled the actual data (solid gray line) in the test set.

As can be noted from Table 1, the algorithm performs as better as the average hourly and daily accesses are higher, since for small volumes of patient income the daily dynamics is closer to a random process. For the predicted data the hourly MAE(h) ranges from 87% of the mean hourly accesses in the smaller facility (ED 6), to the 27% in the bigger one (ED 3). On a daily timescale the prediction quality appear to improve, since for the same ED (ED 6 and ED 3) MAE(d) are 29% and 8.5%.

5 Comparison to Other Models

We compared the performance of our forecasting scheme to two alternative prediction systems: a REplication model (RE), and an Artificial Neural Networm model (ANN). In the RE model the N days ahead prediction is obtained by replicating the most recent sequence in the hystorical dataset sharing the same calendar pattern. In this way the prediction will always mimic the most recent matching past. The ANN model consists in a single layer feedforward network composed by 130 hidden neurons (as obtained by metaparameter optimization). The hidden and output neurons activation functions is the rectified linear (ReLU) activation. This model was trained to reconstruct the $p_N(i)$ vectors in the training set in two different way, based on the input provided:

- Only the $p_M(i)$ vectors (see Model section), thus with no information about any calendar pattern.
- The concatenation between $p_M(i)$ and the calendar condition of the days to reconstruct, the latter encoded in a vector in $\mathbb{R}^N$.

Model	MAE(d)	MBE(d)	VAE(d)	MAE(h)	MBE(h)	VAE(h)
replica	10.17	0.06	88.06	1.59	0.00	2.73
ANN (no calendar)	15.54	13.22	123.52	1.39	0.55	2.10
ANN (calendar)	14.11	11.56	117.47	1.38	0.48	2.08
C-knn	9.91	-1.13	89.72	1.21	-0.05	1.62

Table 2. Performances of the tested models. MAE, MBE and VAE for a single model are calculated averaging the scores of all EDs, and has to be compared with the average mean daily accesses 70.85, and the average mean hourly accesses 2.95 in all EDs.

The models performance are listed in Table 2 where we show, for each algorithms, the average scores for all the ED facilities.

As can be noted, the C-knn outperform all the competing models, while the RE alghoritm is the less biased and second best. On the contrary both the ANNs does not perform well, probably because they encode a representation of the full trainig set, thus their predictions tend to be an average of the whole past dynamic which loose some short scale details instead captured by the C-knn, which restrict the set from which the forecast is built only to the more similar temporal patterns.

6 Conclusion

The adverse consequences of ED crowding can be as much severe as clinical staff and ED administrators are unaware of the incoming situation. The ability to predict future input demand can relieve the negative effects of a possible disadvantageous situation, and support structural intervention to maintain performances and help service improvement. The unsupervised algorithm presented here, belonging to the similar shape algorithm category, is able to automatically provide a short-term hourly forecast based on calendar condition, without the need of a training phase, but only exploiting an historical dataset in which similar patterns are picked up. Thanks to this aspects it is suitable to be directly applied to any specific situation, providing accurate and reliable predictions.

References

1. Statement on Emergency Department overcrowding. Australasian college for emergency medicine. Jul 16;S57 (2011)
2. Liu S. W., Thomas S. H., Gordon J. A., Hamedani A. G., Weissman J. S.: A pilot study examining undesirable events among emergency department-boarded patients awaiting inpatient beds, AnnEmergMed 54(3):381-385 (2009)
3. Richardson D. B.: Increase in patient mortality at 10 days associated with emergency department overcrowding, Med J Aust 184(5):213-216 (2006)
4. Sprivulis P. C., Da Silva J. A., Jacobs I. G., Frazer A. R., Jelinek G. A.: The association between hospital overcrowding and mortality among patients admitted via Western Australian emergency departments, Med J Aust 184(5):208–212 (2006)

5. Pines J. M., Iyer S., Disbot M., Hollander J. E., Shofer F. S., Datner E. M.: The effect of emergency department crowding on patient satisfaction for admitted patients, AcadEmergMed 15(9):825831 (2008)
6. Di Somma S., Paladino L., Vaughan L., Lalle I., Magrini L., Magnanti M.: Overcrowding in emergency department: an international issue.Intern Emerg Med., 10(2):171-5 (2015)
7. Chen-Mei H., Li-Lin L., Yun-Te C., Wang-Chuan J.: Emergency department overcrowding: Quality improvement in a Taiwan Medical Center, Journal of the Formosan Medical Association, 118(1):186-193 (2019)
8. Forero R., Hillman K.M., McCarthy S., Fatovich D.M., Joseph A.P., Richardson D.B.: Access block and ED overcrowding, Emerg Med Australas, 22:119-35 (2010)
9. Gilligan P., Winder S., Ramphul N., OKelly P.: The referral and complete evaluation time study, Eur J Emerg Med, 17:349-53 (2010)
10. Bullard M.J., Villa-Roel C., Bond K., Vester M., Holroyd B., Rowe B.: Tracking emergency department overcrowding in a tertiary care academic institution, Healthc Q 12:99-106 (2009)
11. Richardson D.: Access block point prevalence survey, The Australasian College for Emergency Medicine (2008)
12. Sun B.C., Hsia R.Y., Weiss R.E., et al.: Effect of emergency department crowding on outcomes of admitted patients, Ann Emerg Med, 61:605–11.e6 (2013)
13. Amodio E., Cavalieri d'Oro L., Chiarazzo E., Picco C., Migliori M., Trezzi I., Lopez S., Rinaldi O., Giupponi M.: Emergency department performances during overcrowding: the experience of the health protection agency of Brianza AIMS Public Health, 5(3): 217–224 (2018)
14. Asplin B.R., Magid D.J., Rhodes K.V., Solberg L.I., Lurie N., Camargo C.A. Jr: A conceptual model of emergency department crowding. Ann Emerg Med., 42(2):173–80 (2003)
15. Higginson, I.: Emergency department crowding. Emergency Medicine Journal, 29, 437-443 (2012)
16. Piani Aziendali per la gestione del sovraffollamento in Pronto Soccorso (PGSA)-Linee di indirizzo, regione Toscana. Available at `https://www.frgeditore.it/images/cop/pdf/titolo-1/razionalizzazione/toscana/all_dgr_974`
17. McCarthy ML, Ding R, Pines JM, Zeger SL. Comparison of methods for measuring crowding and its effects on length of stay in the emergency department.AcadEmerg Med., 18(12):1269-77(2011)
18. Jones, S. S., Thomas, A., Evans, R. S., Welch, S. J., Haug, P. J., Snow, G. L.: Forecasting daily patient volumes in the emergency department. Academic Emergency Medicine, 15(2), 159–170 (2008)
19. Schweigler, L. M., Desmond, J. S., McCarthy, M. L., Bukowski, K. J., Ionides, E. L., Younger, J. G. .: Forecasting models of emergency department crowding. Academic Emergency Medicine, 16(4), 301–308 (2009)
20. Peck, J. S., Benneyan, J. C., Nightingale, D. J., Gaehde, S. A.: Predicting emergency department inpatient admissions to improve sameday patient flow. Academic Emergency Medicine, 19(9), E1045–E1054 (2012)
21. Boyle, J., Jessup, M., Crilly, J., Green, D., Lind, J., Wallis, M., Fitzgerald, G.: Predicting emergency department admissions. Emerg Med J, 29(5), 358-365 (2012)
22. Kadri, F., Harrou, F., Chaabane, S., Tahon, C.: Time series modelling and forecasting of emergency department overcrowding. Journal of medical systems, 38(9), 107 (2014)

23. Afilal, M., Yalaoui, F., Dugardin, F., Amodeo, L., Laplanche, D., Blua, P.: Forecasting the emergency department patients flow. Journal of medical systems, 40(7), 175 (2016)
24. Zor, C., ebi, F.: Demand prediction in health sector using fuzzy grey forecasting. Journal of Enterprise Information Management, 31(6), 937–949 (2018)
25. Zhang, Y., Luo, L., Yang, J., Liu, D., Kong, R., Feng, Y.: A hybrid ARIMA-SVR approach for forecasting emergency patient flow. Journal of Ambient Intelligence and Humanized Computing, 1–9 (2018)
26. Barmada, S., Raugi, M., Tucci, M.: A multiobjective optimization algorithm based on selforganizing maps applied to wireless power transfer systems. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 30(3-4), e2145 (2017)

Technische Universität Darmstadt | Alexanderstr. 10 | 64283 Darmstadt

Poster Submission for
ITISE 2019
September 25th-27th, Granada, Spain

Recurrence quantification analysis and network models to support the psychotherapeutic change process

Björn Mattes[1], Simone Bruder[2], Bernhard Schmitz[3]

[1]Diagnostik, Evaluation und Intervention, Technische Universität Darmstadt, [2]Psychosomatik und Psychotherapie, Darmstädter Kinderkliniken Prinzessin Margaret, Darmstadt, [3]Pädagogische Psychologie, Technische Universität Darmstadt

Objective This proof-of-concept shows how the process of therapeutic change can be supported by personal electronic devices. By following the daily ups and downs of symptom strength and mood-related variables, we demonstrate the feasibility of constructing an idiographic psychological system model in the case of an adolescent outpatient of a day treatment clinic.
Method We created an R-based shiny application that directly analyses the incoming data of key psychosomatic and psychological symptoms of the patient. Both patient and therapist have access to the visualized results of the online-analysis and use the application for therapeutic feedback sessions (e.g. Schiepek et al. (2016)).
Result We compare the results of network models that are based on classical vector-auto regression to recurrence quantification analysis and evaluate their use in supporting the therapeutic change process. We also conduct online p-factor analysis and show, how the cross-lagged relationships of the latent factors can be valuable feedback to the patient.
Conclusion By making subsequent use of daily online monitoring patient and therapist gained new insights into the mental network structure and the relating state-dynamics. The case study shows how the analysis of time-series data can provide a plausible insight into the mental structure of a patient and how the developed open source application can be used to relate to the patients' state, cognitions, emotions and behaviour.

M.Sc.Psych.
Björn Mattes

Fachbereich
Humanwissenschaften

Institut für
Psychologie

Psychologische
Diagnostik, Evaluation
und Intervention

Alexanderstr. 10
64283 Darmstadt

Tel.+49 6151 16-23936
Fax +49 6151 16-24073
mattes @psychologie.tu-darmstadt.de

21.06.2019

Short-term solar power forecasting using clustered VAR model over South Korea

Jin-Young Kim, Chang Ki Kim, Hyun-Goo Kim, Yung-Seop Lee, Yong-Heack Kang

152 Gajeong-ro Yuseong-gu, Daejeon, 34129, Republic of Kroea
hyungoo@kier.re.kr

Abstract. Solar power forecasting is a key role for large-scale photovoltaic penetration to distributed generation and management system. This paper proposes an empirical statistical model for forecasting hourly power generation up to three hours as a function of remotely sensed irradiance, UASIBS/KIER (University of Arizona Solar Irradiance Based on Satellite-Korea Institute of Energy Research) over the South Korea. The model uses the vector autoregression for multivariate irradiance time series over a period of about three years (2014 to 2016). Forecasting hourly power generations were additionally conducted using k-mean clustering to classify the national scale forecasting system. The results showed that the mean absolute percentage error with maximum 87.8%.

Keywords: Solar power forecasting, vector autoregression, clustering

1 Introduction

Solar power forecasting is getting important in order to reduce the uncertainty of solar energy generation and perform efficient management. Many studies on the solar energy forecasting and predictability analyses has been carried out so far[1,2]. However less studies has been performed in South Korea because renewable energy penetration were very low compared with fossil and nuclear energy.

To support the opened electricity market and safe solar energy supply, we has been attempted to establish short-term solar power forecasting model and evaluate its predictability within three hours. Section 2 describes of the brief description of the used model and datasets. The results were summarized in Section 3.

2 Methodology

To construct and evaluate an solar power generation forecasting model and its performance, remotely sensed irradiance and observed power supply time series were collected at 215 stations throughout the South Korea. Remotely sensed irradiance and observed power supply data were taken from the University of Arizona Solar Irradiance Based on Satellite –Korea Institute of Energy Research (UASIBS/KIER) model developed by KIER [3] and Korea Power Exchange, respectively.

adfa, p. 1, 2011.

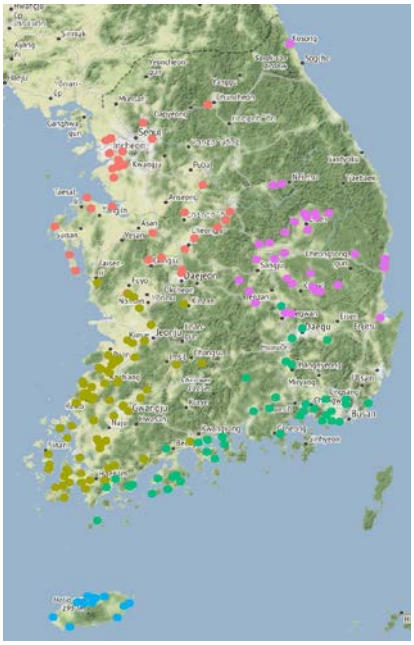
The statistical hourly solar power forecasting proposed in this study employed the vector autoregreesion(VAR) method by considering the periodic correlation between irradiance and observed power generation. The model is an extension of the autoregression model with univariate time series to a multivariate time series model as following.

$$Z_t = \emptyset_1 Z_{t-1} + \cdots + \emptyset_p Z_{t-p} + \epsilon_t, t = 1, \ldots, T,$$

where, $Z_t = \emptyset_1 \begin{matrix} Z_{1t} \\ \vdots \\ Z_{nt} \end{matrix}, \emptyset_i = \begin{matrix} \emptyset_i & \ldots & \emptyset_i \\ \vdots & \ddots & \vdots \\ \emptyset_i & \ldots & \emptyset_i \end{matrix}, t = 1, \ldots, p, \quad \epsilon_t = \begin{matrix} Z_{1t} \\ \ldots \\ Z_{nt} \end{matrix} \sim MN(0, \sum)$. This were developed for all stations throughout the study area and then these were categorized into similar characteristics of VAR using k-mean clustering analysis in terms of national solar power forecasting system.

3 Preliminary Results

This study successfully developed an effective clustered solar power generation forecasting system using the relationship between irradiance and power generation for hourly increased lead time up to three hours. The VAR model has been classified into five regions for the synoptic scale as shown in Figure. According to the analysis, the solar power forecasting using clustered VAR model was better in terms of predictive accuracy than the solar power forecasting using conventional ARIMA model



References

1. Sobri, S., Koohi-Kamali, S., Rahim, N. A., 2018: Solar photovoltaic generation forecasting methods: A review, Energy Conversion and Management, 156, 459-497.
2. Voyant, C., Notton, G., Kalogirou, S., Nivet, M.-L., Paoli, C., Motte, F. Fouilloy, A., 2017: Machine learning methods for solar radiation forecasting: A review, Renewable Energy, 105, 569-582.
3. Kim, C.K., Kim, H.-G., Kang, Y.-H., Yun, C.-Y., Kim, S.-Y, 2019: Probabilistic prediction of direct normal irradiance derived from global horizontal irradiance over the Korean Peninsula by using Monte-Carlo simulation, Solar Energy, 180, 63-70.

Forecasting Energy Consumption in Residential Buildings using ARIMA Models

Muhammad Fahim and Alberto Sillitti

Institute of Information Systems
Innopolis University, Russia
m.fahim@innopolis.ru
a.sillitti@innopolis.ru

Abstract. Forecasting energy consumption in residential buildings is an important information for energy providers to meet the demands of consumers and plan properly grid resources. Moreover, it can be useful to support residents to reduce their utility bills and save energy. In this paper, we presented preliminary short-term forecasting results by utilizing autoregressive integrated moving average (ARIMA) models. Our analysis also connects the forecasting to the number of residents, employment status, number of electrical appliances, and size of the house. We evaluated our model on a publicly available dataset and optimal parameters were obtained through grid search. The mean absolute percentage error (MAPE) is calculated to quantify the forecasting error (i.e., 17.25%). The obtained results confirmed the applicability of our model to real-life applications.

Keywords: Smart meter · Data analysis · Energy consumption · Prediction · Univariate time-series

1 Introduction

A significant amount of energy is consumed in residential buildings and it is expected to increase in near future. The major increasing factors are heating, ventilation, and air conditioning (HVAC) systems. Moreover, hybrid automobiles are charged through residential electricity. Such factors are contributing to increase the quality of life in urban areas while increasing the amount of carbon dioxide (CO_2) emission. One of the known primary reason of global warming is CO_2 emission that is threatening the ecosystem of our planet. Therefore, every step to reduce energy consumption can contribute to maintain the ecosystem of Earth.

The European Commission aims at increasing the consumer awareness regarding their energy consumption in residential buildings through information communication technology (ICT) solutions [11]. One of the best ICT solution to monitor the energy consumption inside residential building are smart meters. These devices have many advantages over the traditional metering system: they

are able to send the meter reading automatically as well as the energy consumption information periodically (i.e., every few seconds). This data can be processed to provide useful information to stakeholders in smart grid domain. More importantly, they improve consumer ability to make informed decision about energy usage inside houses [1]. Smart meters can be easily installed in existing infrastructure without disturbing the aesthetic of buildings and their installation in residential buildings is exponentially increasing all over the world. In the beginning of 2019, more than 50 million smart meters were installed across 50 countries [2]. Smart meters have ability to log energy consumption and transmit information to the associated grid station. The data is collected in the form of univariate time-series and demands further processing to convert this raw data streams into information. Load forecasting is an important information that can be extracted from such a data and can be useful for both residence and suppliers [5]. This forecasting can be long-term or short-term to predict the future energy consumption [12] [13].

Many statistical and machine learning methods have been developed to analyze time-series data and predict the future behavior. Many of them have been developed to address the problems related to the energy consumption of mobile devices [7] [8] [9]. One of the most effective method is ARIMA that has been applied to many application areas [18] [20] [3] [10] [16]. In this paper, we introduce an ARIMA model that is able to process energy consumption logs for short-term forecasting (i.e., over a day). ARIMA models are highly interpretable and have ability to provide unbiased forecasts. Many researchers already applied ARIMA models to smart meter datasets for predicting the anomaly detection as well as load forecasting [15] [4]. Our work differs from the existing work in terms of finding optimal parameters for ARIMA model and design pipeline. These parameters varies from house to house because it depends on the number of residents, number of electrical appliances, and employment status.

The outline of the paper is the following: We briefly explain the related work in Section 2. The methods and materials are introduced in Section 3 followed by results and discussion that are presented in Section 4. We conclude our paper in Section 5 with possible future directions to extend this work.

2 Related Work

The research community developed many energy consumption forecasting models from simple threshold-based methods to complex deep neural networks and achieved acceptable accuracy [15] [14]. The effort is focused on reducing the energy consumption in residential buildings.

Varun *et al.* [15] developed an ARIMA-based model to detect anomalous consumption and present energy theft attacks that are possible in power grids. For anomalous consumption detection, they predict the energy consumption and calculate the 95% confidence interval. Every point that is outside of specified interval is considered as an anomalous behavior.

Fazil *et al.* [14] developed an accurate model to predict the long-term energy consumption in Turkey. Their approach compares the regression analysis, neural networks and least squares support vector machines (SVM). Their results indicate that a least squares SVM based model is more effective than regression analysis and neural network. Their objective was to provide useful insight to policy makers and energy planners.

Nichiforov *et al.* [19] implemented statistical and neural network model to identify patterns in time series data. They also provide the analysis for both models and reported that ARIMA model performance is better than neural networks in comparison of predictive error values. Furthermore, ARIMA has a simpler structure than neural networks. They have several weeks of collected data by their own designed programmable logic controller (PLA).

In comparison to our implemented model and analysis, we have detailed analysis of time-series data and selected parameters for better understanding. We use a publicly available dataset and also reported the number of occupants and the number of appliances.

3 Methods and Materials

The proposed model is presented in Fig. 1. It consists of data pre-processing, descriptive and stationary statistical test and ARIMA model. The details about each step is provided in the following subsections.

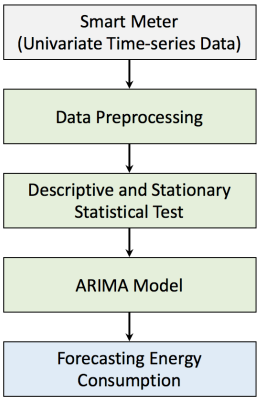


Fig. 1. Univariate time-series data is pre-processed over a pre-defined segmented window of one day. During the time-series analysis, we performed the descriptive and stationary statistical test over the time series. Finally, we estimate the parameters of the ARIMA model and forecast the energy consumption in residential buildings.

3.1 Data Pre-processing

In the proposed forecasting energy consumption model, our first step is data pre-processing. In this phase, univariate time-series data is segmented over a predefined window of a day. One day is the unit analysis of our model that enables to predict the energy consumption accurately. The data stream mean over a day for the period of one month is presented in Fig. 2.

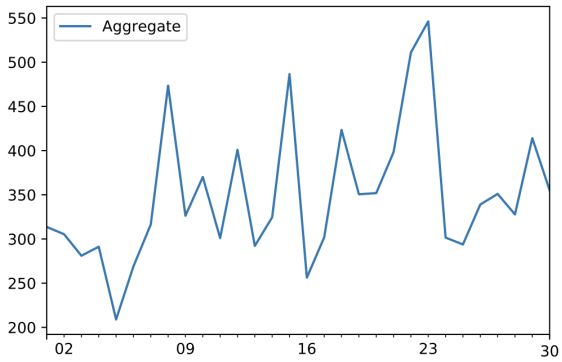


Fig. 2. Average energy consumption for one month over a segmented window.

In Figure 2, the visualization is useful to identify discontinuity in the collected data stream. It may be hard to find the outliers or discontinuity for all observed days using visualization technique. In this case, we also performed a descriptive statistical analysis over the data stream and details are provided in next subsection as well as in results and discussion section.

3.2 Descriptive and Stationary Statistical Test

In this phase, visual and statistical analysis is performed to check the stationary property of time-series data stream. A stochastic processes is stationary when the unconditional joint probability distribution fluctuates around a constant mean and variance during the different time intervals. To identify the stationary property, we plotted the mean and variance over the time-series data stream. Furthermore, we applied the Augmented Dickey-Fuller (ADF) unit root test to time-series data. The test identifies either the time-series data is stationary or not. If the process is not stationary then data probably contains seasonality trends. We can transform non-stationary data stream to stationary by applying common techniques of aggregation, smoothing, and polynomial fitting.

3.3 ARIMA Models

Box-Jenkins introduced ARIMA as a generalized random walk model to determine an accurate forecast, which is based on an explanation of historical data

on a single variable [6]. These models have powerful ability to capture the short range correlations. It combines the autoregression and moving average methods to estimate the forecast of time-series data. To estimate the parameters of the model, it is important to transform time-series into a stationary presentation. The parameter estimation is based on maximizing the likelihood function over the historical data. The model consists of three parameters: the first parameter p is the number of autoregressive terms; the second parameter d is the differencing order; and the third parameter q is the number of lagged. The best parameter values are determined and presented in the following result and discussion section (i.e., Section 4). The general forecasting equation of the model is:

$$\hat{y_t} = \mu + \Phi_1 y_{t-1} + ... + \Phi_p y_{t-p} - \theta_1 e_{t-1} - ... - \theta_q e_{t-q}$$

Where, y_t is the actual value and e_t is the random error of the time-series at time t; Φ are the autoregressive parameters and θ are the moving average parameters. Moreover, autoregressive terms are added while the moving average terms are subtracted.

4 Results and Discussion

We analyzed a publicly available energy consumption dataset REFIT that was collected from residential buildings of United Kingdom (UK) and freely available for research community. This dataset is collected to measure electrical consumption every $6 \sim 8$ seconds. A detailed description of the dataset and related meta-data information can be found in [17]. In this preliminary study of forecasting energy consumption, we presented only one house analysis (i.e., House 1 in the dataset) over a period of six months from June 2014 to Nov 2014. The dataset is pre-processed according to Section 3.1 and presented in Fig. 3.

The dataset was split into training and test sets. The training of the model was performed on four months data while two months data is utilized for testing phase. The socio-economic characteristics of the house is presented in Table 1.

Table 1. The socio-economics details of House 1.

Hosue No.	Occupants	Electrical Appliances	Size
1	2 (Couple)	35	4-bed

A box-and-whisker plot is drawn to visualize the groups of data based on their quartiles in Fig. 4. It can provide the information about outliers. In Fig. 4, few points are outside the box plot and could be outliers. We did not remove such points from smart meter energy consumption data because the dataset description did not described them as outliers. However, removing these possible outliers can increase the forecasting accuracy. We also plotted the data histogram to check the distribution of the data Fig. 5.

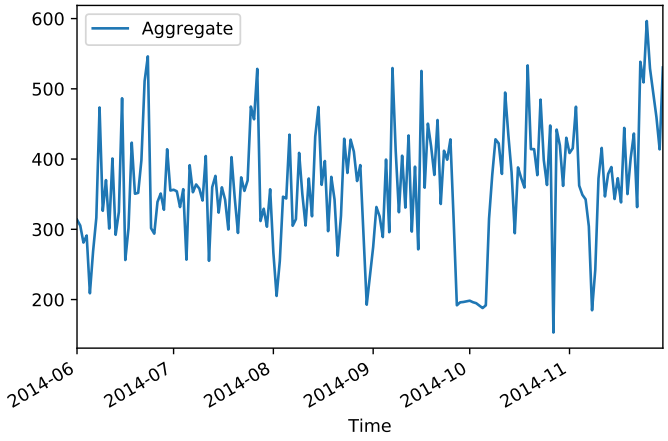


Fig. 3. Average energy consumption over a segmented window of one day for six months.

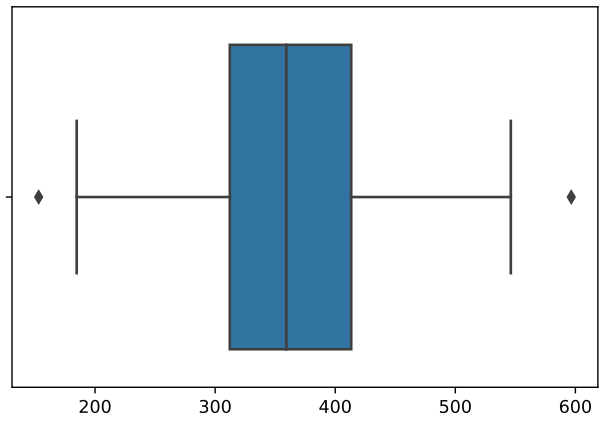


Fig. 4. A box-and-whisker plot over six months.

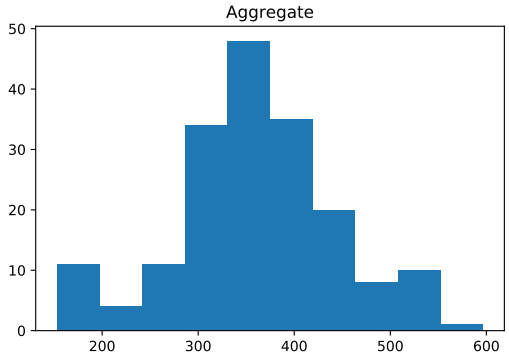


Fig. 5. The distribution of the data is nearly normal and confirms the stability of the time-series.

In Fig. 5, it can be seen that we have almost a normal distribution of the data. While in the next step we check the stationary property of the time-series, rolling mean and standard deviation are presented in Fig. 6.

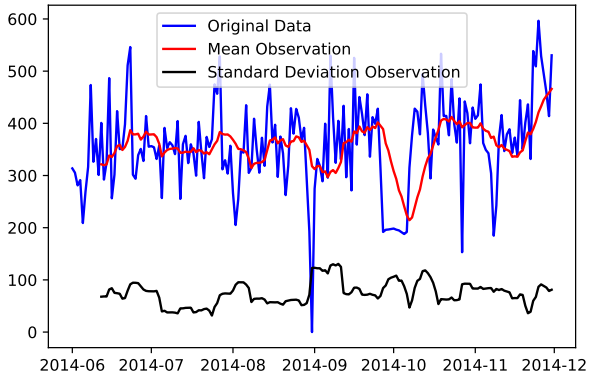


Fig. 6. Rolling mean and standard deviation.

Fig. 6 visually confirms the time-series stationary. However, we also performed the ADF unit root test and it confirms that data has no unit root. This means that the time-series is stationary. To find the best parameter for the model, a grid search is applied and results are reported in Table 2.

According to Table 2, the best parameters are ARIMA (1,1,0). We tested our model during the prediction phase (Fig. 7) and the results confirm the ap-

Table 2. ARIMA model parameter estimation and obtained MAPE.

ARIMA Parameters	MAPE
(p, d, q) = (0, 0, 0)	25.893
(p, d, q) = (0, 0, 1)	22.728
(p, d, q) = (0, 0, 2)	21.532
(p, d, q) = (0, 1, 0)	18.281
(p, d, q) = (0, 1, 1)	18.953
(p, d, q) = (0, 1, 2)	20.052
(p, d, q) = (0, 2, 0)	28.768
(p, d, q) = (0, 2, 1)	18.614
(p, d, q) = (1, 0, 0)	20.916
(p, d, q) = (1, 0, 1)	19.806
(p, d, q) = (1, 1, 0)	**17.253**
(p, d, q) = (1, 1, 1)	20.687
(p, d, q) = (1, 2, 0)	23.515

plicability of the model in real-life settings. We also presented the last week of June, July, August, and September in Fig. 8, 9, 10, 11 respectively. In all cases, we have the MAPE error between 17% and 20% that confirms the applicability of our forecasting model.

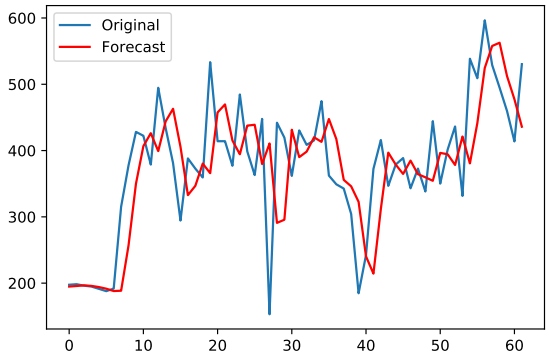


Fig. 7. Forecasts of our ARIMA-based model to predict the energy consumption in residential building.

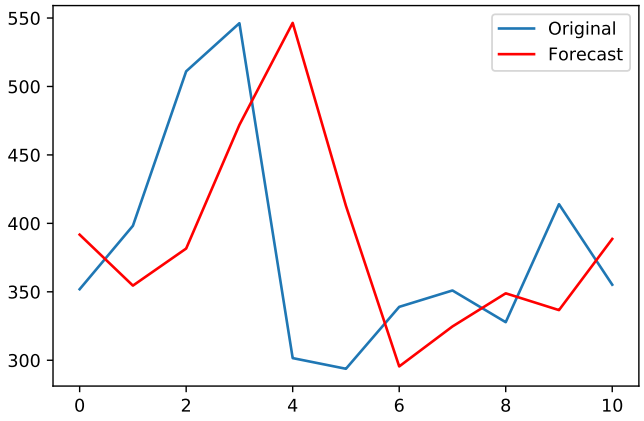


Fig. 8. Forecasts of our ARIMA-based model to predict the last week of June.

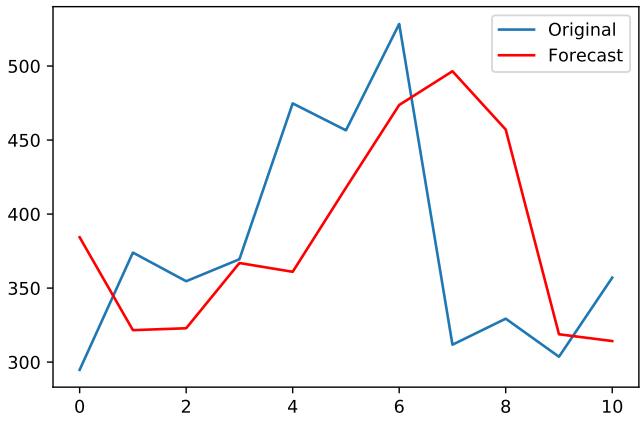


Fig. 9. Forecasts of our ARIMA-based model to predict the last week of July.

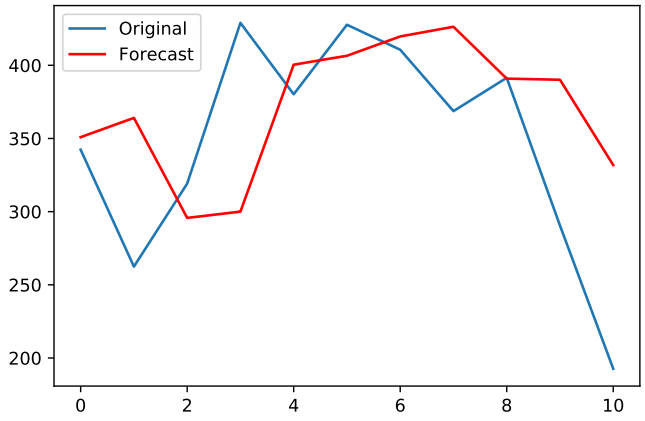


Fig. 10. Forecasts of our ARIMA-based model to predict the last week of August.

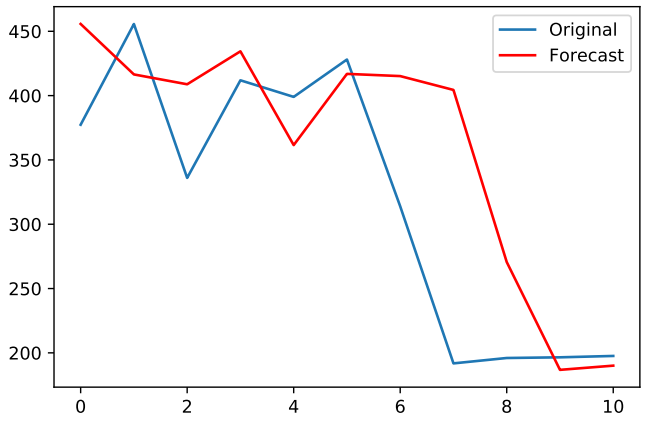


Fig. 11. Forecasts of our ARIMA-based model to predict the last week of September.

5 Conclusion

Forecasting energy consumption plays an important role for consumer as well as energy suppliers. Consumers can save energy and reduce the utility bills while also contributing to maintain the ecosystem of our planet. In this paper, we presented an ARIMA-based forecasting model to predict the energy consumption in residential buildings. We reported our preliminary result in forecasting domain and obtained an acceptable MAPE of 17.25%. The scope of this study is limited to one residential building, while in future work we plan to forecast the energy consumption of different residential buildings while considering the electrical appliances individually.

References

1. Advantages and disadvantages of smart meters. https://www.npower.com/blog/2018/08/29/pros-cons-smart-meter/, accessed: 2019-05-04
2. Smart grid infrastructure grows in emerging markets. https://www.iotworldtoday.com/2019/03/06/smart-grid-infrastructure-grows-in-emerging-markets/, accessed: 2019-05-04
3. Agnieszka, D., Magdalena, L.: Detection of outliers in the financial time series using arima models. In: 2018 Applications of Electromagnetics in Modern Techniques and Medicine (PTZE). pp. 49–52. IEEE (2018)
4. Alberg, D., Last, M.: Short-term load forecasting in smart meters with sliding window-based arima algorithms. Vietnam Journal of Computer Science **5**(3-4), 241–249 (2018)
5. Bâra, A., Oprea, S.V.: Electricity consumption and generation forecasting with artificial neural networks. In: Advanced Applications for Artificial Neural Networks. IntechOpen (2017)
6. Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time series analysis: forecasting and control. John Wiley & Sons (2015)
7. Corral, L., Georgiev, A.B., Sillitti, A., G., S.: A method for characterizing energy consumption in android smartphones. In: 2nd International Workshop on Green and Sustainable Software (GREENS 2013) (2013)
8. Corral, L., Georgiev, A.B., Sillitti, A., G., S.: Method reallocation to reduce energy consumption: An implementation in android os. In: 29th ACM Symposium on Applied Computing (SAC 2014). ACM (2014)
9. Corral, L., Georgiev, A.B., Sillitti, A., G., S.: A study of energy-aware implementation techniques: Redistribution of computational jobs in mobile apps. Sustainable Computing: Informatics and Systems **7** (2015)
10. Deb, C., Eang, L.S., Yang, J., Santamouris, M.: Forecasting energy consumption of institutional buildings in singapore. Procedia Engineering **121**, 1734–1740 (2015)
11. Directive, E.E.: Directive 2012/27/eu of the european parliament and of the council of 25 october 2012 on energy efficiency, amending directives 2009/125/ec and 2010/30/eu and repealing directives 2004/8/ec and 2006/32. Official Journal, L **315**, 1–56 (2012)
12. Fahim, M., Sillitti, A.: An anomaly detection model for enhancing energy management in smart buildings. In: IEEE International Conference on Communications,

Control, and Computing Technologies for Smart Grids (SmartGridComm 2018). IEEE (2018)
13. Fahim, M., Sillitti, A.: Analyzing load profiles of energy consumption to infer household characteristics using smart meters. Energies **12** (2019)
14. Kaytez, F., Taplamacioglu, M.C., Cam, E., Hardalac, F.: Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. International Journal of Electrical Power & Energy Systems **67**, 431–438 (2015)
15. Krishna, V.B., Iyer, R.K., Sanders, W.H.: Arima-based modeling and validation of consumption readings in power grids. In: International Conference on Critical Information Infrastructures Security. pp. 199–210. Springer (2015)
16. Moayedi, H.Z., Masnadi-Shirazi, M.: Arima model for network traffic prediction and anomaly detection. In: 2008 International Symposium on Information Technology. vol. 4, pp. 1–6. IEEE (2008)
17. Murray, D., Stankovic, L., Stankovic, V.: An electrical load measurements dataset of united kingdom households from a two-year longitudinal study. Scientific data **4**, 160122 (2017)
18. Nath, B., Dhakre, D., Bhattacharya, D.: Forecasting wheat production in india: An arima modelling approach. Journal of Pharmacognosy and Phytochemistry **8**(1), 2158–2165 (2019)
19. Nichiforov, C., Stamatescu, I., Făgărăşan, I., Stamatescu, G.: Energy consumption forecasting using arima and neural network models. In: 5th International Symposium on Electrical and Electronics Engineering (ISEEE). pp. 1–4. IEEE (2017)
20. Nyoni, T.: Modeling and forecasting inflation in kenya: Recent insights from arima and garch analysis. Dimorian Review **5**(6), 16–40 (2018)

Predicting hospital admissions with integer-valued time series

Radia Spiga[1,2] , Mireille Batton-Hubert [1], Marianne Sarazin[3,4]

[1] Mines Saint-Etienne, Univ Clermont Auvergne, CNRS, UMR 6158 LIMOS, Institut Henri Fayol, F - 42023 Saint-Etienne France

[2] Service de Santé publique et d'information médicale, Centre Hospitalo-Universitaire, Saint-Etienne, France

[3] Institut National de la Santé et de la Recherche Médicale, UMR en Santé 1136, Paris, France

[4] Centre Ingénierie et Santé, Ecole Nationale Supérieure des Mines, Saint Etienne, France

Introduction

Prediction of seasonal epidemics have been widely treated in the medical literature, with various methods to forecast the future cases of a given disease, when it's about infectious diseases: compartmental methods that forecast the number of persons at each state of the epidemic (susceptible, infected, resistant) are used (1), as well as methods based on time series (ARMA ,ARCH,…)(2), this last method can be successful with large sample of data, assuming their normal distribution. Here, we propose to test time series for count data when the continuous time series are not adapted to predict health activity.

The aim of this work is to predict the number of hospital admissions of future weeks at local scale, using methods for integer-valued time series: INAR(p) and INGARCH(p,q) methods(3–5), we have also test the autoregressive methods for continuous data.

Data

Weekly admissions for pulmonary diseases at the Saint Etienne University Hospital (France), and weekly number influenza like illness (ILI) provided by The French *Sentinelles* Network (the network of general practitioners(6)), from 2011 to mars 2018.

The first step is to describe and characterize these two data, then to apply the appropriate count time series, next multivariate method for count time series will be tested.

Data characterization

Description of data are summarized in table.1, overdispersion is observed for both ILI and hospital data, suggesting a negative binomial distribution, and there were 133 count of 0 on ILI data. Form this results we assume a negative binomial distribution for hospital data (7,8), and zero-inflated negative binomial distribution for ILI data(9,10).

Data fitting and forecasting

Hospital admission time series have been predicted with a better accuracy with the NB INGARCH (1,1), (Figure.1(A)). For the ILI time series ZINB INAR (1) have failed to predict the outbreak of epidemics (Figure.1(B)), higher order of ZINB INAR will be tested.

Perspective

The present work is based on an empirical approach; we are planning to develop a similar approach based on Bayesian methods, combining specificity of the integer value data. The final aim is to predict hospital admissions, including ILI series as an exogeneous variable in the count tie series model(11).

	Number of weeks	mean	variance	min	Max	Number of 0
Hospital admissions for pulmonary diseases	377	17	58.47	4	57	0
Influenza like illness (Sentinelles)	377	8.63	264.62	0	107	133

Table.1: parameter description of hospital and ILI time series from 2011 to Mars 2018

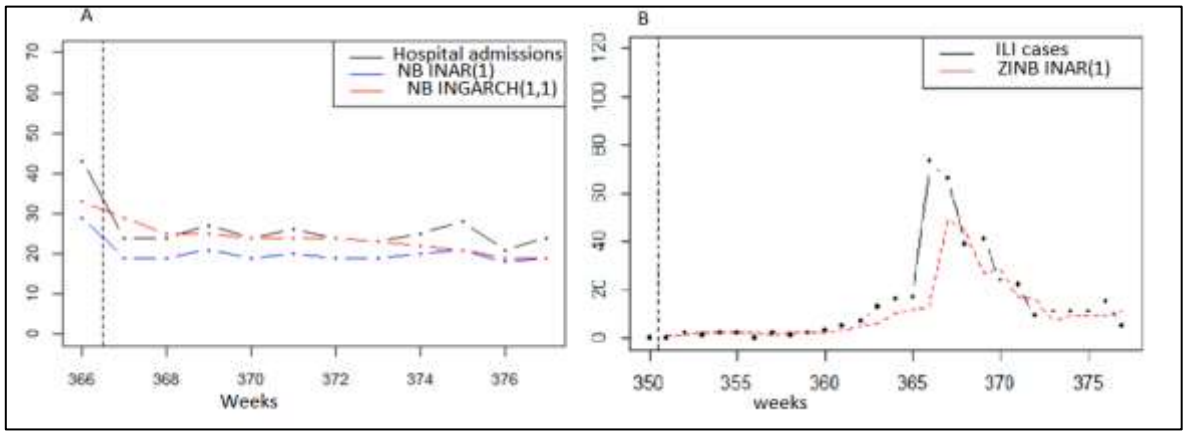


Figure.1: Prediction of hospital admissions (A), and influenza like illness (B). H=10

References

1. GIRALDO JO, PALACIO DH. Deterministic SIR (Susceptible–Infected–Removed) models applied to varicella outbreaks. Epidemiol Infect. 2008;136(5):679-87.

2. Chen Z, Zhu Y, Wang Y, Zhou W, Yan Y, Zhu C, et al. Association of meteorological factors with childhood viral acute respiratory infections in subtropical China: an analysis over 11 years. Arch Virol. avr 2014;159(4):631-9.

3. McKenzie E. Some Simple Models for Discrete Variate Time Series1. JAWRA J Am Water Resour Assoc. 1985;21(4):645-50.

4. Al-Osh MA, Alzaid AA. First-Order Integer-Valued Autoregressive (inar(1)) Process. J Time Ser Anal. 1987;8(3):261-75.

5. Ferland R, Latour A, Oraichi D. Integer-Valued GARCH Process. J Time Ser Anal. 2006;27(6):923-42.

6. Réseau Sentinelles > France >. https://websenti.u707.jussieu.fr/sentiweb/?page=accueil

7. Zhu F. Modeling overdispersed or underdispersed count data with generalized Poisson integer-valued GARCH models. J Math Anal Appl. mai 2012;389(1):58-71.

8. Xu H-Y, Xie M, Goh TN, Fu X. A model for integer-valued time series with conditional overdispersion. Comput Stat Data Anal. déc 2012;56(12):4229-42.

9. Piancastelli LSC, Barreto-Souza W. Inferential aspects of the zero-inflated Poisson INAR(1) process. Appl Math Model. oct 2019;74:457-68.

10. Qi X, Li Q, Zhu F. Modeling time series of count with excess zeros and ones based on INAR(1) model with zero-and-one inflated Poisson innovations. J Comput Appl Math. janv 2019;346:572-90.

11. Pedeli X, Karlis D. Some properties of multivariate INAR(1) processes. Comput Stat Data Anal. nov 2013;67:213-25.

Neural Network approaches for Air Pollution Prediction

Marijana Cosovic[1], Emina Junuz[2]

[1]Faculty of Electrical Engineering, University of East Sarajevo, Istocno Sarajevo, BiH
[2]Faculty of Information Technology, Dzemal Bijedic University, Mostar, BiH

marijana.cosovic@etf.ues.rs.ba, emina@edu.fit.ba

Abstract. According to World Health Organization exposure to elevated levels of air pollutants is directly linked to health issues. Air pollution in the countries of Western Balkans is an evident problem. Sarajevo, the capital of Bosnia and Herzegovina, is at the end of 2018 presented as the world's most polluted city. Previous years it was a runner up being amongst five most polluted cities together with cities from Macedonia and Montenegro.
Air pollution forecasting could be performed using machine learning techniques. Deep learning techniques are proving useful since weather behavior can be analyzed as time series data. We observed hourly, daily, weekly and seasonal periodicities of the air pollutant and meteorological data. Furthermore, this paper evaluates performance of several neural network architectures applied to weather data in greater Sarajevo area for the 2014-2018 period. Urban air quality is of outmost importance to the public hence we compare prediction accuracies of the proposed forecasting models.

Keywords: Machine learning·Air pollution forecast·MLP·LSTM·

1 Introduction

The ever-increasing problem of air pollution in countries of Western Balkans is especially pronounced during the winter season and in the areas where population density is high. Pollution exposure extend from harmful effects on human health [1], [2] to environment [3], [4]. Air quality degradation in different areas of Bosnia and Herzegovina is mainly contributed to vehicle emissions, coal burning heating systems as well as thermal power stations using fossil fuels. First two are the main reasons behind the very frequent peaks of increased air pollution in the city of Sarajevo situated in the valley surrounded by the hills and mountains. In addition to the topography of greater Sarajevo area temperature inversion effect frequently forms clouds at low level with pollutant rich air beneath. We recognize that air pollutants within buildings can be as harmful as the pollution generated outdoors, but the scope of this study is limited to ambient air.

It is often that urban population of city of Sarajevo is exposed to the levels of particulate matter deemed unhealthy according to World Health Organization (WHO). In this

study we consider the most common air pollutants: particulate matter up to ten micrometers (PM_{10}), sulphur dioxide (SO_2), nitrogen dioxide (NO_2), carbon monoxide (CO), and ground level ozone (O_3). In addition, we used atmospheric pressure, temperature, relative humidity and wind speed. We extend on work reported in [1] and aim to predict PM_{10} concentration, based on air pollutant concentrations and major meteorological data, by using machine learning techniques for five years period from 2014.

We explore two different neural network architectures and its application to air pollution forecasting. By exploring a multilayer perceptron (MLP) feedforward neural networks and Long Short-Term Memory (LSTM) recurrent neural networks we choose a suitable approach to deal with nonlinear systems such as air pollution and aim at next-day particulate matter forecast. Considering the data collection process was not the most comprehensive or reliable in the past, although improving recently, we evaluate and address the missing data issue to deliver recommendations on air quality prediction using machine learning techniques.

2 Datasets

Air pollutants' data are collected by the Federal Hydrometeorological Institute (FHMZ) BH [5]. We extend our previous study [1] and consider a five-year period with hourly averaged values for air pollutants and major meteorological variables. Hence, we consider five air pollutants, namely, PM_{10}, SO_2, NO_2, CO and O_3. During the observed period of five years (2014-2018) five measuring stations in the vicinity of the greater Sarajevo area (Otoka, Ilidza, Bjelave, Vjecnica, and Alipasina) were operational during most or part of the observed period. The stations are operated by either the Federal hydrometeorological Institute or the Cantonal Institute for Public Health. Latitude, longitude and altitude of the measuring stations is given in Table 1. In addition, we have obtained continuous measurements (averaged hourly values) of the temperature, relative humidity, pressure, and average wind speed during the observed period. Choosing these variables (date, time, type of day (working week day, Saturday, and Sunday), temperature, relative humidity, pressure, average wind speed, PM_{10}, SO_2, NO_2, CO and O_3) was shown suitable for detecting/forecasting elevated values of particulate matter in the ambient air. The dimension of the feature matrix for each year is 8760x13. The aim of this paper is to structure a forecasting problem similar to [1] in which the pollution of the next day is determined based on previous data.

Table 1. Station location: latitude, longitude and altitude

Station	latitude (°,',")	longitude(°,',")	altitude (m)
Otoka	43 50 54 N	18 21 49 E	512
Ilidza	43 49 40 N	18 18 49 E	499
Bjelave	43 52 03 N	18 25 23 E	631
Vijecnica	43 51 33 N	18 26 04 E	554
Alipasina	43 51 28 N	18 24 44 E	545

2.1 Air Pollution Regulations and Data Acquisition

During the last couple of decades, the European Union (EU) has established various regulations through implementation of standards and objectives for air quality monitoring. In particular, air quality directives for pollutant concentration thresholds that should not be exceeded in proposed period of time are defined.

Hence, in accordance with air quality directive proposed by EU standards the particulate matter up to ten micrometers in size (PM_{10}) concentration computed as maximum daily mean should not exceed 50 µg/m^3 on more than 35 days per year. World health organization recommends annual mean to not exceed 20 µg/m^3.

Principle of beta ray attenuation is used for continuous monitoring of atmospheric particulate matter up to ten micrometers in size (PM_{10}) concentration. HORIBA APDA-371 ambient dust monitor is used for data collection. Two measuring stations, Otoka and Ilidza stations, in the greater Sarajevo area provide measurements of PM_{10} concentration in continuation over the last five years. Bjelave and Vijecnica measuring station have recorded measurements from 2016 until 2018 while Alipasina measuring station provided PM_{10} concentration data for 2014 and 2015. Daily average values for PM_{10} concentration during the 2014-2018 are shown in Fig. 1. The red line in the figure represents a guideline value of 50 µg/m^3 suggested by the World Health Organization (WHO), and that is the maximum tolerated value at any given time. During the winter season that value is exceeded frequently as shown in Fig. 1. Annual mean computed for all the years and on all the locations that are measuring PM_{10} concentration show two-fold to five-fold increase in comparison to value suggested by WHO.

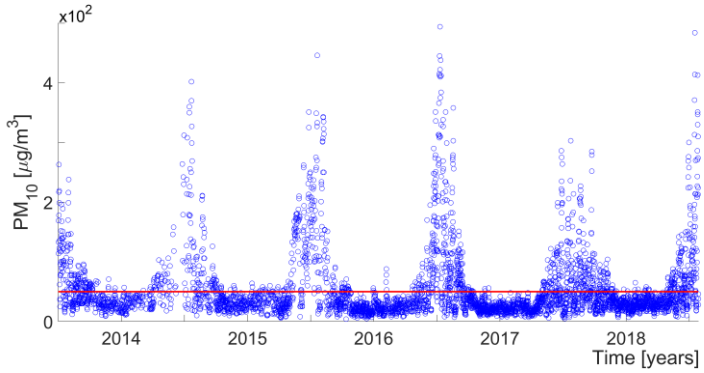


Fig. 1. PM_{10} concentrations measured at five measuring stations in greater Sarajevo area during 2014-2018.

Concentration of sulphur dioxide (SO_2) pollutant in the ambient air computed as maximum daily mean should not exceed 125 µg/m^3 on more than 3 days per year according to EU air quality directives. World health organization, on the other hand, recommends daily mean of SO_2 not exceed 20 µg/m^3. This is due to health effects now known to be associated with much lower levels of SO_2 than previously believed.

Principle of UV fluorescence is used for continuous monitoring of atmospheric SO_2 concentration. HORIBA APSA-370 SO_2 ambient monitor is used for data collection.

It employs proprietary, internal dry-method sampling to achieve the highest levels of sensitivity and accuracy. Two measuring stations, Otoka and Ilidza stations, in the greater Sarajevo area pro-vide measurements of SO_2 concentration in continuation over the last five years. Bjelave and Vijecnica measuring station have recorded measurements from 2016 until 2018 while Alipasina measuring station provided SO_2 concentration data for 2014 and 2015. Daily average values for SO_2 concentration during the 2014-2018 are shown in Fig. 2. The red line in the figure represents a guideline value of 20 μg/m^3 suggested by the WHO, while the blue line represents the maximum tolerated value computed as daily mean required by the EU air quality directive. During the winter season that value is exceeded frequently, and certainly more often than three times per year as it can be observed in Fig. 2.

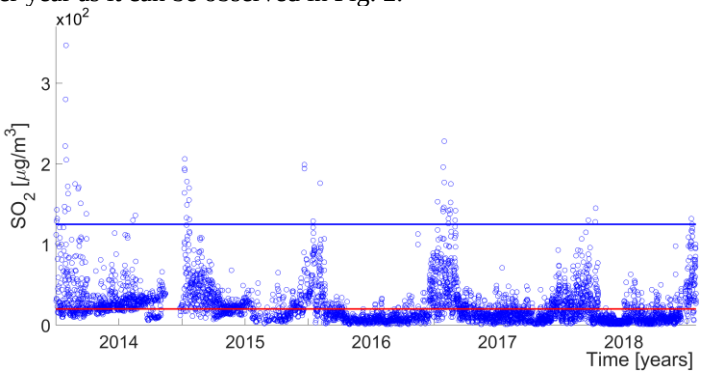


Fig. 2. SO_2 concentrations measured at five measuring stations in greater Sarajevo area during 2014-2018.

Nitrogen dioxide (NO_2) is the air pollutant being monitored in accordance with air quality directive proposed by EU standards with suggested concentration computed as maximum daily mean that should not exceed 85 μg/m^3. It is also suggested that 200 μg/m^3 hourly average should not be exceeded more than 18 days per year.

Principle using a cross-flow modulated semi decompression chemiluminescence method is used for continuous monitoring of atmospheric NO_2 concentration. The Horiba APNA-370 is used for data collection. Two measuring stations, Otoka and Ilidza stations, in the greater Sarajevo area provide measurements of NO_2 concentration in continuation over the last five years. Bjelave and Vijecnica measuring station have recorded measurements from 2016 until 2018 while Alipasina measuring station provided NO_2 concentration data for 2014 and 2015. Daily average values for NO_2 concentration during the 2014-2018 are shown in Fig. 3. The blue line represents the maximum tolerated value computed as daily mean required by the EU air quality directive.

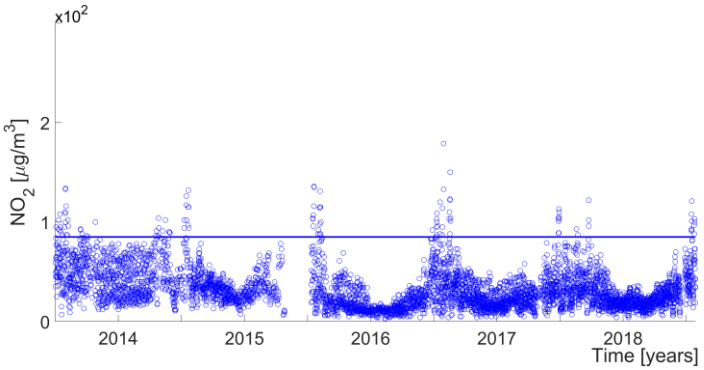


Fig. 3. NO_2 concentrations measured at five measuring stations in greater Sarajevo area during 2014-2018.

During the winter season that value is exceeded frequently, and certainly more often than 18 times per year. In addition, we have observed that Otoka measuring station reported annual mean for 2014, as well as 2016-2018 above 40 µg/m^3.

In accordance with air quality directive the ground level of ozone (O_3) computed as maximum daily 8-hour mean should not exceed 120 µg/m^3 on more than 25 days per year averaged over three years. World health organization recommends this value to be even further decreased to 100 µg/m^3 since research links daily mortality and lower ozone concentrations.

Non dispersive ultra-violet-absorption method is used for continuous monitoring of O_3 concentrations. Horiba APOA-370 ambient ozone monitor is used for data collection. There is no single measuring station in the greater Sarajevo area that measures ground level ozone in continuation of last five years. Bjelave and Ilidza measuring station have recorded measurements from 2016 until 2018. There is approximately 10km air distance between the two measuring stations with first station being at higher altitude by 132m than the second one. Otoka measuring station have records for the years 2015, 2017, and 2018. Vijecnica and Alipasina measuring stations do not have the measuring equipment installed for O_3 monitoring. It has been shown that altitude of the measuring site affects the ground ozone level [6]. Hence, we observe increased values at Bjelave station during the summer months of the last three years. Considering the altitude of Bjelave measuring station and proximity of a busy road, introducing nitrogen oxides from vehicle emissions, it is expected to have increased ground ozone levels during the sunny days of summer months in respect to other measuring stations. During the year of 2016 there were 148 days, followed by 76 days during the year of 2017 as well as during 2018 in which the maximum daily 8-hour mean value of ground level ozone exceeded allowed values. This is somewhat difficult to observe in Fig. 4 since it presents the daily average values for O_3 concentration during the 2014-2018. The red line in the figure represents a guideline value of maximum daily 8-hour mean suggested by the WHO, while the blue line represents the maximum daily 8-hour mean standardized by the EU air quality directive.

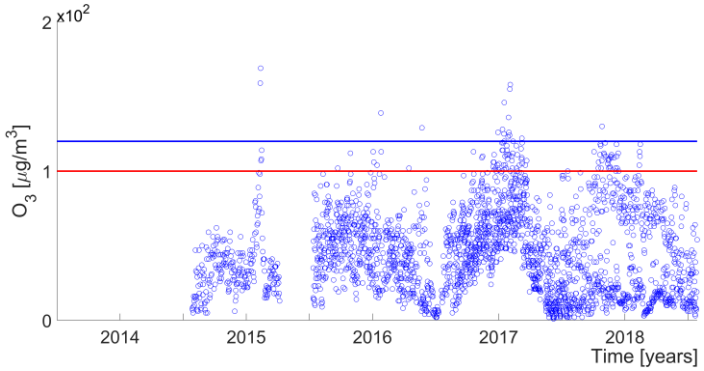


Fig. 4. O_3 concentrations measured at five measuring stations in greater Sarajevo area during 2014-2018.

Carbon monoxide (CO) is the air pollutant being monitored in accordance with air quality directive proposed by EU standards with suggested concentration computed as maximum daily mean that should not exceed 5 mg/m^3. It is also suggested that 10 mg/m^3 hourly average should not be exceeded more than18 days per year. In Fig. 5 we can observe the values of CO concentration within the allowed limits.

Method based on non-dispersion cross modulation infrared analysis is by which continuous monitoring of atmospheric carbon monoxide concentrations is performed. Horiba APMA-370 ambient carbon monoxide monitor is used for data collection. There is no single measuring station in the greater Sarajevo area that measures carbon monoxide concentrations in continuation of last five years. Bjelave measuring station have recorded measurements from 2016-2018, while Vijecnica records carbon monoxide data from 2017 until present. Otoka measuring station have records for the years 2014 and 2015. Ilidza and Alipasina measuring stations do not have the measuring equipment installed for CO monitoring. Daily average values for CO concentration during the 2014-2018 are shown in Fig. 5. The blue line represents the maximum tolerated value computed as daily mean required by the EU air quality directive.

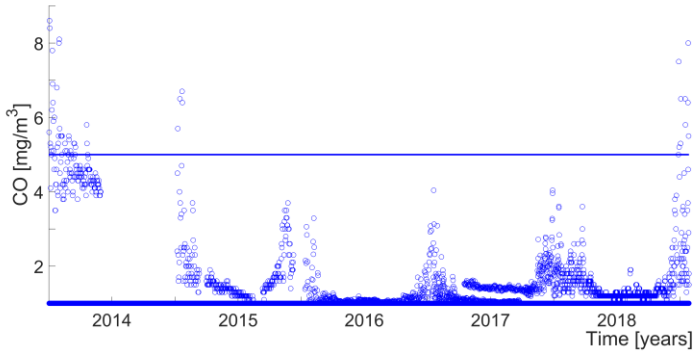


Fig. 5. CO concentrations measured at five measuring stations in greater Sarajevo area during 2014-2018.

2.2 Missing Data

Ambient air pollutant data was continuously measured during 2014-2018 at various monitoring stations in the urban part of Sarajevo city. Only data that has passed control and validation is reported in this study. The table below shows missing daily average air pollutant data. Daily average values are also computed based on hourly average values for those days that are not missing more than six hourly average values. We can observe decreasing trend in missing daily average values in the period 2014-2018 for the measuring stations that are actively collecting data.

Table 2. Missing daily average values categorized by measuring stations for five pollutants during 2014-2018. N/A refers to data not collected at specified station.

Year	Station	PM	SO_2	NO_2	CO	O_3
2014	Otoka	60	52	12	217	N/A
	Ilidza	184	166	121	N/A	N/A
	Bjelave	N/A	N/A	N/A	N/A	N/A
	Vijecnica	N/A	N/A	N/A	N/A	N/A
	Alipasina	263	79	40	N/A	N/A
2015	Otoka	126	175	116	86	N/A
	Ilidza	189	133	231	N/A	N/A
	Bjelave	N/A	N/A	N/A	N/A	N/A
	Vijecnica	N/A	N/A	N/A	N/A	N/A
	Alipasina	68	23	124	N/A	117
2016	Otoka	200	74	198	N/A	N/A
	Ilidza	70	233	19	N/A	75
	Bjelave	118	24	11	19	3
	Vijecnica	155	158	238	N/A	N/A
	Alipasina	N/A	N/A	N/A	N/A	N/A
2017	Otoka	26	62	24	N/A	71
	Ilidza	132	33	34	N/A	34
	Bjelave	46	79	59	25	35
	Vijecnica	42	49	28	44	N/A
	Alipasina	N/A	N/A	N/A	N/A	N/A
2018	Otoka	36	44	16	N/A	14
	Ilidza	47	21	41	N/A	18
	Bjelave	50	48	30	30	23
	Vijecnica	30	26	33	27	N/A
	Alipasina	N/A	N/A	N/A	N/A	N/A

We observe improvement in data collection during the 2014-2018 period in greater Sarajevo area as shown in Table 3. During 2014-2015 hourly average values of PM_{10}, SO_2 and NO_2 air pollutants were collected in three locations, while during 2016-2018 in four locations. Hourly average values of CO were being collected only in one location during 2014-2016 and in two locations over the last couple of years.

Table 3. Missing daily average values categorized by the year for five pollutants during 2014-2018. N/A refers to data not collected at specified station. Number in the bracket represents the number of different data collection locations within greater Sarajevo area.

Year	PM	SO_2	NO_2	CO	O_3	Total
2014	507 (3)	294 (3)	173 (3)	217 (1)	N/A	1191
2015	383 (3)	331 (3)	471 (3)	86 (1)	117 (1)	1388
2016	543 (4)	489 (4)	466 (4)	19 (1)	82 (2)	1599
2017	246 (4)	223 (4)	145 (4)	69 (2)	140 (3)	823
2018	163 (4)	139 (4)	120 (4)	57 (2)	55 (3)	534

Data collection of ground level ozone O_3 was not existent in 2014. Steady improvements over the last four years resulted in three locations for collection of ground level ozone realized in last two years. In addition, we observe that missing hourly average values for most air pollutants have decreasing trend with time in last two years as shown in Fig. 6. Prior to that at Otoka measuring station during 2015 and 2016 we can observe increased number of missing data. In addition, majority of the missing values during 2015 and 2016 occurred in the summer or fall.

Furthermore, we categorized all the missing data according to season of the occurrence. It has been observed that majority of missing data falls into summer and fall season for all observed pollutants and during the observation time of five years. The missing data could be contributed to equipment failure, need for calibration of analyzers and unforeseen problems that are either difficult to recognize at their occurrence or resolve in a short amount of time. In general, calibration of the equipment is done during the summer season. The increased number of missing values during that period could be contributed to the scheduled outages. Same analogy could be applied to Ilidza station increase in missing data during 2015.

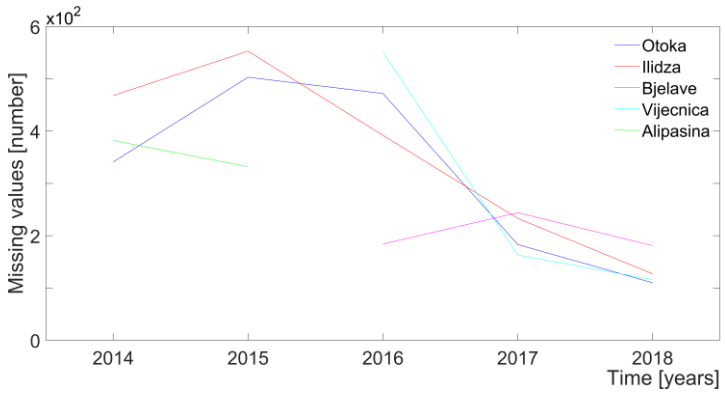


Fig. 6. Total number of missing daily average values for all pollutant concentrations measured at five measuring stations during 2014-2018.

For example, during the year of 2017 at Bjelave measuring station we observe 244 while during the year of 2018 we observe 181 missing daily average values for all five

pollutants combined. The same trend can be observed at Otoka measuring station with 183 missing daily averages for all five pollutants in the year 2017 while in the following year those values decreased to 110. Ilidza and Vijecnica measuring stations follow the decreasing trend in missing daily average values from 233 and 551 in the year 2017 to 127 and 116 in the year 2018. During the 2018, data collection equipment was not available on average per pollutant from 22 to 36 days. This still does not affect the minimum availability of data that should be in the 75% to 90% range.

Considering missing data needs to be addressed we explored generation of methods and tools for effective data assertion [1]. We addressed data imputation and explored statistical and machine learning approaches [7]. Computation and insertion of overall mean, although a fast method for data imputation, introduces side effects such as dataset variance reduction.

We considered seasonal adjustment since our data has seasonal variation. Time-series specific methods of data imputation, such as last observation carried forward (LOCF) and next observation carried backwards (NOCB), were used. Authors in [7] presented imputation model based on machine learning techniques (LASSO regression and Bagging Ensemble). Results reported show improvements in suggested data imputation method in comparison to hot deck methods. We have used the root mean squared error (RMSE) to evaluate different methods of data imputation. Performance of LASSO-Bagging method has shown reduction in RMSE values on our datasets.

3 Air Pollution Forecasting Models

We have considered four seasons for air pollution forecasting: spring, summer, fall, and winter. The dataset for the winter season was collected every hour from January 1, until March 21, as well as December 22, until December 31, of each calendar year considered in the study. The spring season dataset includes data from March 21, until June 21, of each calendar year. The summer season dataset was collected from June 21 until September 23, while the fall dataset was collected from September 23 until December 22. For each of the years (2014-2018) datasets contain 2208 instances for spring (92 days observed), 2256 instances for summer (94 days observed), 2160 instances for fall (90 days observed) and 2136 instances for winter (89 days observed). The exception is the year of 2016 being a leap year and having an additional winter day, hence, 90 winter days observed. As in our previous work [1] we have considered nineteen features for each of the datasets: date, time, type of the day (working week day, Saturday, and Sunday), PM_{10}, SO_2, NO_2, CO, O_3, previous-days PM_{10} (up to seven features for seven previous days), atmospheric pressure, temperature, relative humidity and wind speed. Hence, the dimensions of the spring, summer, fall, and winter feature matrices are 2208x19, 2256x19, 2160x19, and 2136x19 respectively for the regular calendar years and 2208x19, 2256x19, 2160x19, and 2160x19 respectively for the leap year.

The only complete data in terms of all pollutants measured we have for 2014-2018 period at Bjelave measuring station.

3.1 Correlation

Considering variables of the feature matrices are not normally distributed and the relationship between them is not linear [8] we have used Spearman's correlation coefficients (SCC) to measure correlation among air pollutants and meteorological parameters. The Kolmogorov-Smirnov test was used to determine the type of distribution of our data. SCC assesses the monotonic relationship between the variables and has a value between –1 and +1. Results of SCC for Bjelave station during 2016-2018 period data are shown in Table 4. Spearman's correlation coefficient amongst selected features was computed for all four seasons, as well, and shows stronger correlation during the winter season when the air pollution is at its maximum. In addition, positive correlation is observed between PM_{10}, CO, NO_2, and SO_2, while negative correlation is observed between those air pollutants and O_3. We observed increase in positive values of SCC between pairs PM_{10} and CO, PM_{10} and NO_2, PM_{10} and SO_2 as well as CO and NO_2 for period of 2016-2018. For example, SCC computed between values of PM_{10} and CO air pollutants increases in value from 0.21 to 0.8 as illustrated in Table 4, hence we observe stronger relationship between PM_{10} and CO features as the number of missing data decreases. On the other hand, we observe weaker SCC of 0.26 between SO_2 and CO for year 2017 as well as NO_2 and SO_2 for both 2017 and 2018. Spearman's correlation coefficients between different features are computed without prior data imputation.

We also observed negative correlation of all air pollutants and temperature apart from O_3. Atmospheric pressure and relative humidity also have positive correlation coefficient with PM_{10}, CO, NO_2, and SO_2 and a negative correlation with O_3. The largest correlation coefficients are present between PM_{10}, CO, NO_2, and SO_2, confirming the fact that these air pollutants are originating from the same sources and have same monotonic relationship between air pollutants. The values are further improved with missing data decline.

Table 4. Spearman's correlation coefficient amongst selected air pollutant features for Bjelave measuring station for 2016-2018.

	PM_{10}	**CO**	**NO_2**	**SO_2**	**O_3**
PM_{10}	1	0.45/ 0.6/ 0.8	0.29/ 0.59/ 0.74	0.03/ 0.45/ 0.59	-0.24/ -0.39/ -0.33
CO		1	0.74/ 0.75/ 0.90	0.59/ 0.26/ 0.67	-0.45/ -0.44/ -0.32
NO_2			1	0.74/ 0.49/ 0.69	-0.39/ -0.29/ -0.32
SO_2				1	-0.26/ -0.04/ -0.32
O_3					1

We also observe that temperature and wind speed have a negative correlation coefficient with PM_{10}, CO, NO_2, and SO_2 and a positive correlation with O_3. Daily temperatures, relative humidity and wind have effects on O_3 formation. Generally, more beneficial meteorological circumstances for ozone formation are higher temperatures with lower relative humidity, as opposed to lower temperatures with higher relative humidity. Also, depending on wind speed (high/light), we could have dilution or building up of ozone concentration, hence positive correlation of wind speed and ozone concentrations.

3.2 Neural Network Models

The artificial neural network is a parallel distributed processor consisting of many interconnected simple elements, neurons. Each of the elements has the natural ability to store and use knowledge acquired through experience. Properties of neural networks are resistance to failure, learning and adapting to process inaccurate data in unstructured and uncertain environments, modeling and managing multivariable processes and approximating an arbitrary continuous nonlinear function to the desired accuracy.

We used the Keras [9] deep learning Python library with the Theano backend for development and evaluation of deep learning models. Two models [1] were developed for air pollution forecast: MLP and LSTM models. Model based on a backpropagation algorithm for training of fully connected multiplayer perceptron (MLP) neural network is similar to our previous work [10] developed for short-term load forecasting. In general, data flow through the network is spread from input to the hidden layer. The units in the hidden layer receive the value and transfer it to next hidden layer via activation function. In our case, a neural network with two hidden layers is the optimal model for air pollution forecasting. Traditionally, sigmoid and tanh activation functions are used, but the authors in [11] have shown that better performance can be achieved using a rectifier activation function. When passing information through the network, summarized input and output values for each analysis unit are counted. In the output layer a local error is calculated to determine the increase or decrease in weights. We devised MLP models with 13-19 features for the input layer. In forecasting PM10 concentration we opted to generate PM10 values up to seven days prior. A model that minimizes performance measures presented in 3.3 will be chosen. In Keras, using Dense class is one of the ways to define fully connected layers. Network weights were initialized to random numbers using either uniform or Gaussian distribution

We use 10-fold cross validation for determining accuracy on the test dataset, and as we increase the number of hidden layers beyond two, classification accuracy decreases, as noticed in [10]. General guidelines for determining the number of neurons within each hidden layer are used. We selected neural network architecture based on trial and error, but in accordance with the following general guidelines: the number of neurons in hidden layers should be between the sizes of input and output layers, and they should be the sum of 2/3 of the input layer neurons and output layer neurons. Hence, we trained the neural network with two dense hidden layers with 15 and 10 neurons, respectively [10].

While in the feed-forward neural networks information travels in forward direction only, recurrent neural networks (RNN) can maintain information from computation of an earlier input, hence having memory capabilities. RNN performance degrades when long-term dependencies between previous inputs and present targets occur. Implementation of a LSTM (Long Short-Term Memory) cell allowed for better tradeoff concerning RNN performance at one side and lapsed time between previous inputs and present targets on the other side. A LSTM network is RNN composed of LSTM cells. LSTM solve the vanishing gradient problem of RNN by updating the state of each cell in an additive way. We developed a LSTM model for air pollution forecasting in the Keras

deep learning library such that given the meteorological conditions and concentration of pollution of prior days, as well as expected air pollution for the next day, we can forecast air pollution for the next day. All the features are normalized with a zero mean and standard deviation of one. Datasets are split into training and testing datasets, and we fit our LSTM model on 80% of the data and evaluate it on the remaining 20%. We trained the LSTM model with 50 neurons in the first hidden layer. A neuron in the output layer enabled prediction of air pollution. We used an Adam optimization algorithm [12] instead of stochastic gradient descent because of its forthright implementation, computation efficiency and small memory requirements.

3.3 Performance Measures

The mean absolute error (MAE) is the sum of absolute differences between the actual value and the forecast, divided by the number of observations.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| \tag{1}$$

Hence, the mean absolute error is an average of the absolute errors where f_i is the prediction and y_i the actual value as shown in (1) where all individual differences have equal weight. The mean absolute percentage error (MAPE) is another measure of prediction accuracy of a forecasting model, and it is shown in (2). It is the average of absolute errors divided by actual observation values.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{f_i - y_i}{f_i}\right| \tag{2}$$

The mean squared error (MSE) shown in (3) represents the sum of the squared errors divided by the number of observations.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2 \tag{3}$$

The mean square error (MSE) is probably the most commonly used error metric. It penalizes larger errors because squaring larger numbers has a greater impact than squaring smaller numbers.

The root mean squared error (RMSE) shown in (4) represents the sample standard deviation of the differences between predicted and true values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - y_i)^2} \tag{4}$$

4 Results and Conclusion

We have forecasted air pollution in the last week of all four-season datasets for five continuous years from 2014 until 2018. Those periods were excluded during the training step. The air pollution forecast based on the two models (MLP and LSTM) for winter season of each year was performed from March 15 until March 21, respectively. During that period we could observe elevated values of particulate matter in general as well as local increase in the evening hours mostly due to coal burning for heating since all of the measuring stations reside in urban areas. Air pollution forecast for the spring season was performed from June 15 until June 21 of each year considered in the study. Summer season air pollution forecast was performed from September 17 until September 23, and for the fall season from December 16 until December 22. The best prediction results were obtained with one day prior information; hence the performance measure values were the smallest in that case.

We also observed that majority of the missing data during 2016-2018 period belongs to summer and fall season but with different ratios of total number of missing data. For example, distribution of the air pollutant missing data during winter, spring, summer and fall of 2016 is 23.26%, 21.44%, 28.84% and 26.46% of the total number of missing data while for 2017 the same distribution is 18.83%, 23.94%, 28.55%, and 28.68%. In addition, distribution of the air pollutant missing data during seasons of 2018 comprises of 18.35%, 11.99%, 28.46% and 41.2% of the total number of missing data. We have observed over the last three years drop in missing values as illustrated in Table 3. During that time the number of measuring stations increased from 15 to present 17.

By comparison of real and forecasted air pollution concentration on March, 20 of each year for winter season and on June, 20 of each year for spring season using MLP model we obtained the best results with one day prior information. We also observed that the smallest performance measures can be contributed to the datasets with least missing data. This is consistently shown in Table 5. throughout the years of 2016-2018.

Table 5. MAPE and RMSE values for Bjelave station using MLP and LSTM models for Winter, Spring, Summer and Fall season of 2016-2018

Station/ Season		MLP		LSTM	
		MAPE	RMSE	MAPE	RMSE
Bjelave 2016	Winter	0.019	0.028	0.018	0.026
	Spring	0.022	0.040	0.021	0.029
	Summer	0.031	0.045	0.029	0.034
	Fall	0.034	0.051	0.031	0.041
Bjelave 2017	Winter	0.015	0.026	0.012	0.024
	Spring	0.019	0.027	0.018	0.023
	Summer	0.023	0.037	0.019	0.022
	Fall	0.027	0.039	0.021	0.031
Bjelave 2018	Winter	0.016	0.025	0.015	0.021
	Spring	0.018	0.029	0.017	0.028
	Summer	0.024	0.039	0.023	0.036
	Fall	0.028	0.037	0.027	0.035

LSTM model, when compared to MLP model, provides reduced performance measures on all datasets considered. The largest positive correlation is observed consistently between particulate matter and carbon monoxide as shown in Table 4 for 2016-2018. We observe that during the summer and fall season the most missing data comes from PM_{10} and CO pollutants. We conclude that due to this phenomenon performance measures shown in Table 5. for winter and spring season are slightly increased but nevertheless smaller than in case of MLP model. Air pollution forecast based on LSTM model for the remaining seasons were performed for the same dates as in MLP forecasting model.

We have used previously developed [1] models for air pollution forecasting based on artificial neural networks: feed-forward MLP and recurrent LSTM neural networks. In this paper we analyzed the performance of those models applied to datasets collected in five-year period from several location in the Sarajevo city area. Since we encountered missing data problems, we explored statistical and machine learning methods for data imputation. We have considered four seasons for air pollution forecasting: spring, summer, fall, and winter.

The MLP model with two hidden layers was optimal since choosing additional hidden layers caused performance indices to deteriorate. The LSTM model used one hidden layer. We used a cross-validation technique to determine the number of neurons in each of the layers. We concluded that some features had greater effect than others on the forecast, as well as that performance measures were the best for forecast done based on the previous day's information. By using more prior information, performance indices worsened. LSTM model performed slightly better than MLP model for all seasons considered. For future work we will explore LSTM model with additional layers as well as other methods of data imputation in search of reducing performance measures further.

References

1. Ćosović, M., Junuz, E.: Air Pollution Forecasting using Machine Learning Techniques. In: Proceedings of International Conference on Time Series and Forecasting, pp. 264-273 (2018)
2. Colovic Daul, M., Kryzanovski, M., Kujundzic, O.: Air Pollution and Human Health: The case of the Western Balkans, (2019) [Online]. Available: https://www.unenvironment.org/news-and-stories/press-release/air-pollution-responsible-one-five-premature-deaths-19-western
3. De Marco, A., Proietti, C., Anav, A., Ciancarella, L., D'Elia, I., Fares, S., Fornasier, M. F., Fusaro, L., Gualtieri, M., Manes, F., Marchetto, A., Mircea, M., Paoletti, E., Piersanti, A., Rogora, M., Salvati, L., Salvatori, E., Screpanti, A., Vialetto, G., Vitale, M., Leonardi, C.: Impacts of air pollution on human and ecosystem health, and implications for the National Emission Ceilings Directive: Insights from Italy. Environ. Int. **125**, 320-333 (2019)
4. Rodriguez, M. C., Dupont-Courtade, L., Oueslati, W.: Air pollution and urban structure linkages: Evidence from European cities. Renew. Sust. Energ. Rev. **53**, 1-9 (2016)
5. Federal Hydrometeorological Institute (FHMZ) Bosnia and Herzegovina. [Online]. Available: http://www.fhmzbih.gov.ba/latinica/index.php

6. Chevalier, A., Gheusi, F., Delmas, R., Ordonez, C., Sarrat, C., Zbinden, R., Thouret, V., Athier, G., Cousin, J.M.: Influence of altitude on ozone levels and variability in the lower troposphere: a ground-based study for western Europe over the period 2001–2004. Atmos. Chem. Phys. **7**, 4311-4326 (2007)
7. Rosati, G.: Construcci´on de un modelo de imputaci´on para variables de ingresos con valores perdidos a partir de Ensamble Learning. Aplicaci´on a la Encuesta Permanente de Hogares, Revista SaberES, **9**(1), 91-111 (2017)
8. Hauke, J., Kossowski, T.: Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. Quaestiones Geographicae **30**(2), 87–93 (2011)
9. Chollet, F.: Keras, GitHub repository. [Online]. Available5: https://keras.io (2015)
10. Bećirović, E., Ćosović, M.: Machine learning techniques for short-term load forecasting. In: Proceedings of 4th International symposium on environmental friendly energies and applications. Serbia, Belgrade (2016)
11. Nair, V., Hinton, G. E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of ICML, pp. 807-814 (2010)
12. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. Computing Research Repository CoRR. [Online]. Available: https://arxiv.org/abs/1412.6980 abs/1412.6980 (2017)

Long and Short Term Prediction of Power Consumption using LSTM Networks

Juan Carlos Morales[1], Salvador Moreno[1], Carlos Bailón[1], Héctor Pomares[1], Ignacio Rojas[1], Luis Javier Herrera[1]*

Computer Architecture and Technology Department, University of Granada, Spain
*jherrera@ugr.es

Abstract. This work presents an adapted deep learning approach for short and long term prediction of a national power consumption time series. A modified LSTM network based on direct prediction of four hours horizon is presented, and the improvements in efficacy by using external inputs for the prediction are shown. Additionally, an alternative based on Convolutional Neural Networks is applied to the same objective, and results are compared. Finally, the performance of those models for long term time series forecasting is assessed, and a modification of the training process on the LSTM network adapted to long term prediction is shown, leading to relevant improvements in accuracy for this task.

Keywords: Power Consumption Time series · Long Term Prediction · Short Term Prediction · LSTM Networks · Convolutional Neural Networks

1 Introduction and problem description

Deep Learning arose as the most powerful and successful machine learning paradigm at the early years of this decade. Specifically, for time series prediction, text processing and sequence analysis and synthesis, among other problems, Recurrent Neural Networks and their specific forms LSTM and GRU neural networks, have received increasing attention due to their potential and success in their operation. For time series prediction many works related with GRU and LSTM networks have appeared in the recent literature [9][3][5]. Nonetheless, not only recurrent neural networks, but also Convolutional Neural Networks have shown to present interesting capabilities in the extraction of specific patterns from the sequences of data helpful in the prediction of the future [2][4].

In the field of time series prediction, and depending on the specific problem tackled, short term prediction is the most widely approached problem. Attaining the highest performance of the models is usually the main focus in the short term, and different variations in the models architecture and consideration of additional input variables can have an important influence on the short term accuracy. On the other hand, long term prediction implies the construction of a model which is able to learn from its own inputs and be robustly accurate in the further horizon. Experience leads to the fact that optimized models for

short term prediction noisily fail in the long term, and additional techniques are needed to attain reasonable models for the long term.

Power consumption is a critical well known problem in the specific literature [3][2]. Identification of consumption trends is essential in household power management, but also on the larger horizon, governments need precise predictions to manage long term power production. This work deals with the Iberian Peninsula Power Demand series, specifically the data from jaunary 2009 to june 2016 [7] [8].

This works deals with a short and a long term approach for the prediction of this time series. Specifically, a LSTM model for short term prediction of 24 complete values is presented. Moreover, a CNN approach for the same purpose is also presented and compare. Finally the limitations of the optimal models for short term prediction is shown, and a modified training methodology of the LSTM network is presented for long term forecasting with interesting results.

The rest of this work is organized as follows. Section 2 describes the dataset tackled in this work. Section 3 describes the methodology. Section 4 presents the results obtained and comparison with previous works. Finally, in section 5 the conclusions of this work are drawn.

2 Data Description

2.1 Power Consumption dataset

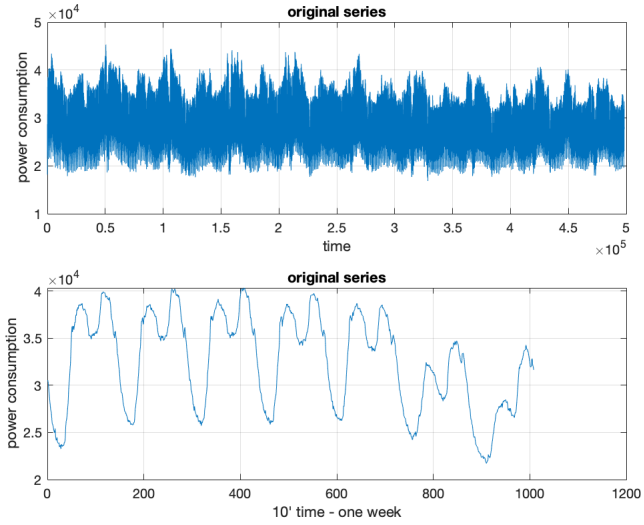
The time series is the Spanish power consumption series taken from [1] [7] [8]. The series is the 10 minutes consupmption from 1 Jan 2009 until the 30 Jun 2016. A shape of the global series is shown in figure 1.a). Also a detailed view of a week consuption is shown in figure 1.b).

The series presents a marked daily seasonality, with certain behaviours in weekends. Also special shapes of the daily consumption can be observed in national holidays, and between different seasons of the year, probably due to the sun rising and falling times of the day (Spain adjusts its clock time twice a year, in expectancy of change in European regulations).

2.2 External data

Additional information that is known could influence the behaviour of the series along the period of time involved was collected. Specifically information related to national public holidays was considered. Moreover weather conditions were also taken into account by collecting the mean, minimum and maximum temperatures and precipitation levels per day, of 10 of the most inhabited cities in Spain (taking into consideration their distribution along the peninsula: Madrid, Barcelona, Valencia, Sevilla, Coruña, Bilbao, Vizcaya, Malaga, Murcia and Alicante). Thus in total, for each series of 144 values (each 10 minutes) of consumption per day, 43 external values were provided: day of the year, day of the week, national holiday, min, mean and max temperature level and precipitation level for each of the aforementioned 10 cities.

Fig. 1. a) Spanish Power Consumption series from 1 jan 2009 to 30 jun 2016. b) detailed view of a week consumption (second week of 2009), in which different daily patterns can be observed (weekdays and weekends).



3 Methodology

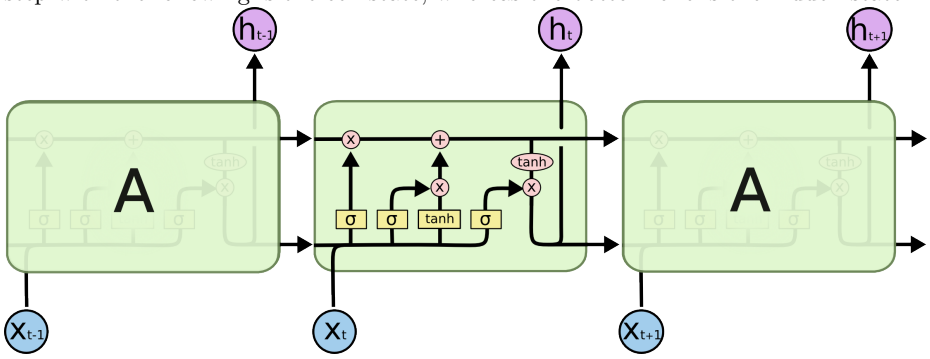
3.1 Short Introduction to LSTM and Convolutional Neural Networks

Since Neural Networks have shown the ability to solve a large variety of problems, many different architectures have been created to focus on particular tasks. Two of such models are Long Short-Term Memory Neural Networks (LSTM) and Convolutional Neural Networks (CNN).

LSTMs have an internal memory that evolves after each time step depending on the input (hidden state). The main difference between LSTMs and the most basic Recurrent Neural Networks (RNN) is the existence of a second internal memory that controls in a more precise way how the hidden state changes at a time step (cell state, see figure 2). This helps in learning not only short-term time dependencies, but long-term as well. Thanks to this evolving state, they are very good for learning data that evolves through time. Another advantage of this type of networks over other RNN architectures is that it avoids the vanishing gradient problem.

On the other hand, CNNs are the state of the art in image and video processing, among other types of problems, due to their ability to learn and extract specific complex patterns from three dimensional data (although they can work on higher dimensions, and even on simpler problems with lower dimensionality). CNNs traditionally take as input a three dimensional matrix and produce

Fig. 2. Basic LSTM architecture, (taken from Colah's blog https://colah.github.io/posts/2015-08-Understanding-LSTMs/). The X and h are the inputs and outputs at each time step. The upper line that connects each time step with the following is the cell state, whereas the bottom one is the hidden state.



another three dimensional matrix as output. This is achieved by applying a convolution operation between the original matrix and a set of filters (see figure 3. For example, for an initial (U x V x W) (height, width, depth) matrix, we can apply a set of T filters (R x S x W) to obtain a (U x V x T) output. These networks perform really well when dealing with images since they are able to capture spatial dependencies. Moreover, thanks to the use of small filters (R and S are usually 3, 5 or 7), and the use of pooling layers after the convolution operations, the number of parameters of the network is reduced drastically.

3.2 Proposed Short Term LSTM network

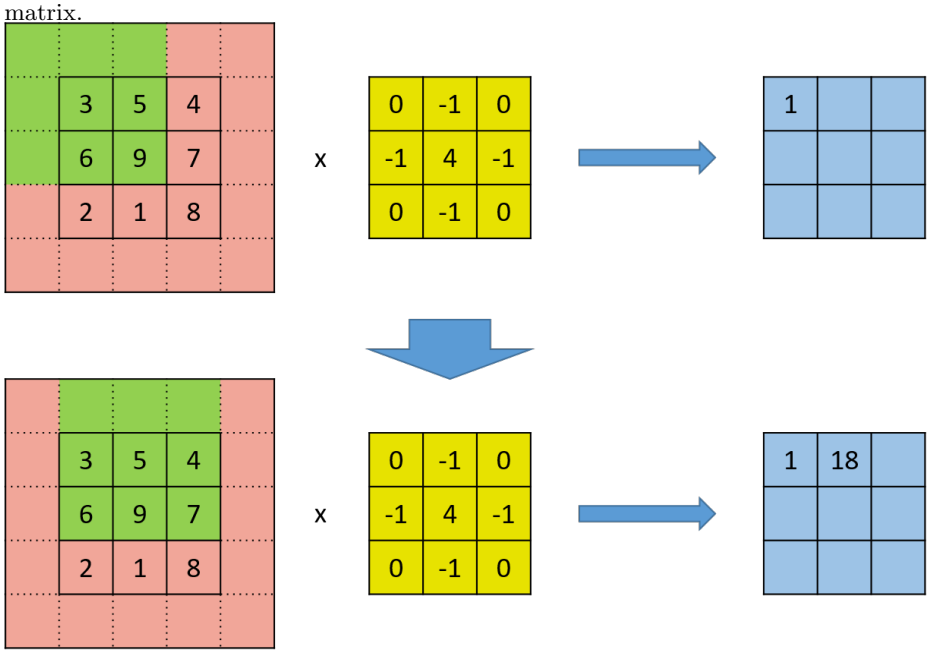
When predicting the power consumption, as mentioned, it is highly important to have a good short term accuracy. To achieve that, we propose a model based on a LSTM that takes as inputs data from the previous 40 hours (240 values) and predicts the next 4 hours (24 values).

The 240 values are split into packets of 24 and the LSTM is fed with them in order at each time step. Except for the final time step, none of the previous outputs of the LSTM are collected. This process acts like an encoder and allows the LSTM to build up its hidden state at each time step. An approach similar to this one has been used in other areas such as machine translation [6].

At the end of the 10th step, the output of the LSTM is collected and concatenated with the external data. The external data used correspond to the external values at the very beginning and at the end of the 24 values we want to predict (86 values). We also add two more external values that correspond to the time data within the day for those two values, for a total of 88 values. The resultant vector is then passed through three fully connected layers to get the final prediction.

The network was trained during 23 epochs using a batch size of 128 and Adam optimizer. During the first two epochs, the learning rate used was 5e-3 and from

Fig. 3. Convolutional operation. The original image (left) is multiplied by the filter (middle) and the results are added to form a single output digit (right). Empty pixels are considered here as 0s (zero padding). The filter is moved through the whole input matrix.



there onwards, the learning rate was 2.5e-5. Moreover, a L2 regularization with a regularization strength of 1e-5 was used to avoid overfitting.

A sketch of the network is shown in figure 4.

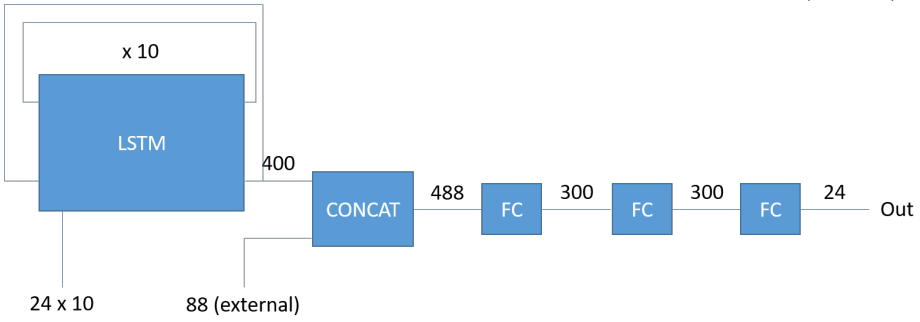
3.3 Proposed Convolutional neural network

As explained in section 3.1, CNNs perform especially well when dealing with images, so the first step in building a CNN to approach the power consumption problem is to distribute the training data in a way that resembles an image.

For that purpose, we have stacked previous days of the data in several rows. Each row is created by concatenating the m previous values to the hour we want to start the prediction (but corresponding to past days) and the n following values, being n the number of values we want to predict. For the last row, only the m previous values are known, so for the following n a placeholder value (all zeros) is used. In this way, at each column the time stamp of the day for all rows is the same. By organizing the data like this, we hope that the network is able to use the spatial information to be able to fill the all zeros region with an actual prediction.

This two-dimensional input is then processed by three convolutional layers and flattened into a vector. Here, in the same way as in the LSTM model, the

Fig. 4. Proposed LSTM network for short term forecasting of 24 values (4 hours).

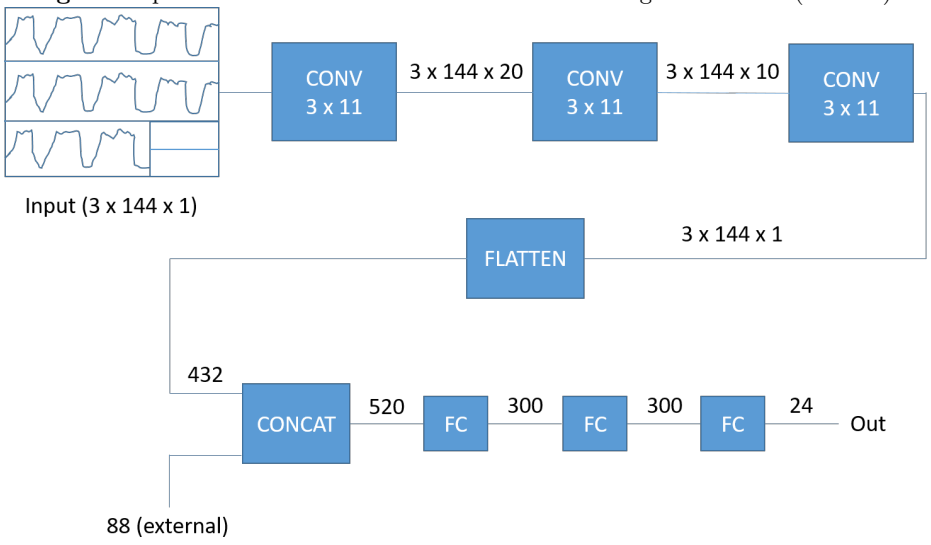


external data is added and the result is processed by three fully connected layers to get the prediction of the n values.

The value of n has been set to 24 and the value of m to 120, to cover a full day per row. The number of rows has been set to three. The kernel for the convolutions has been set to 3 x 11. The network was trained for 17 epochs using a batch size of 64 and Adam optimizer. During the first two epochs, the learning rate used was 2e-3 and from there onwards, the learning rate was 5e-5. As in the LSTM model, a L2 regularization with a regularization strength of 1e-5 was used.

A sketch of the network is shown below in figure 5.

Fig. 5. Proposed CNN network for short term forecasting of 24 values (4 hours).



3.4 Improvements over the LSTM network for long term time series forecasting

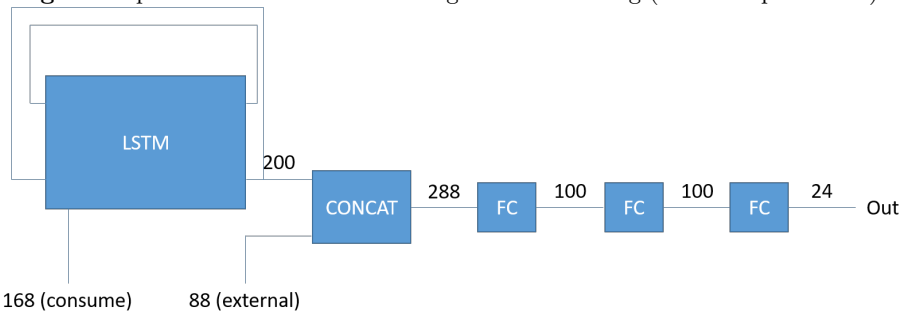
Since the two previous models have been trained for short term forecasting, they are expected to perform poorly when applying the model to obtain a recursive prediction (that is, using the previous prediction to get the next one) over a long time period.

We propose a different LSTM based network to approach the long term prediction. In this case, the LSTM is used in a more conventional way, outputting 24 values at each time step and keeping the hidden state between each prediction. At each time step, the network is fed with the 168 previous values, as well as the external data in the same way as before. The architecture is identical to the short-term network except for the fact that we are outputting results at each time step.

To avoid the network from fixating too much in the previous prediction, we have also trained the network in a recursive way (using the output for the input at the next time step). To reduce the noise while training, every 11 time steps the true data is fed again. The network was trained for 31 epochs using a batch size of 24 and Adam optimizer (the batch size is fixed to be the same as the number of outputs since we only have that number of possible starting points without repeating outputs). During the first 9 epochs, the learning rate used was 5e-4 and from there onwards, the learning rate was 1e-4. A L2 regularization with a regularization strength of 3e-4 was used.

Figure 6 shows a sketch of the network.

Fig. 6. Proposed LSTM network for long term forecasting (recursive prediction).



4 Results

In this work, all experiments were performed on a laptop PC with GPU Nvidia GeForce GTX 760M. Python 3.6.8 and Tensor Flow 9.0 were used for the implementation of the solutions.

Root Mean Square Error (RMSE) was used in the experimentation as a standard performance measure in time series, also in order to be able to compare with previous works.

The dataset was subdivided in training-val-test in a 80%/10%/10% ratio for all three networks. Therefore the test set corresponded to the last 346 dayly values of the series, i.e. from the 12 of July 2015 to the 21 of Jun 2016.

4.1 Short-Term time series forecasting

Results obtained for 4 hours (24 values) ahead using the two alternatives presented in subsection 3.2 and 3.3 are shown in table 1.

Table 1. 4 hours prediction

Short Term Prediction		
Method	**Training RMSE**	**Test RMSE**
LSTM	317.28	337.04
LSTM w external features	301.44	325.36
CNN	345.96	346.90
CNN w external features	314.68	328.32
DFFNN [7]	-	501.14
Deep Learning approach [8]	-	587.47

As it can be observed, the results show an improvement over the ones obtained in previous works, which consider 24 different feed forward models to predict each of the 24 values. In addition, these results show that the introduction of the external values in the model slightly improves the result. It can also be appreciated how the LSTM and the CNN models perform very similar, with a slight advantage for the LSTM one. Figures 7 and 8 show the prediction obtained for the LSTM model.

Fig. 7. Short-term LSTM model over the test set. The dashed red line is the real data while the blue line is the prediction.

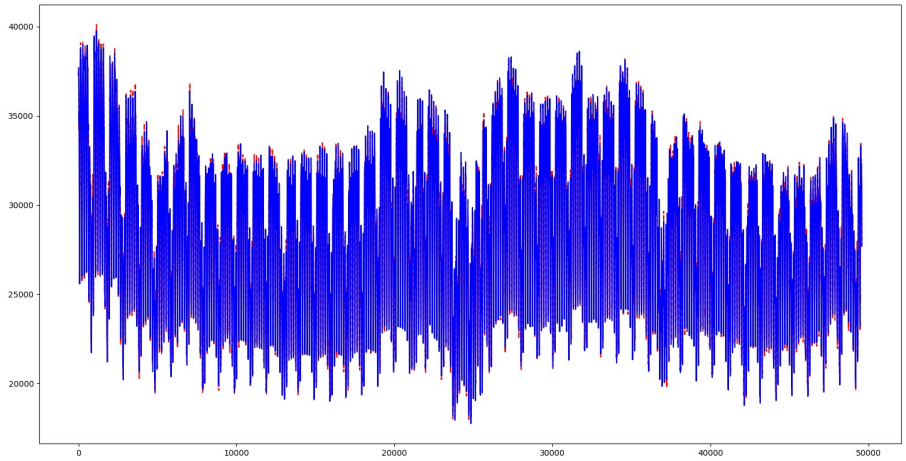
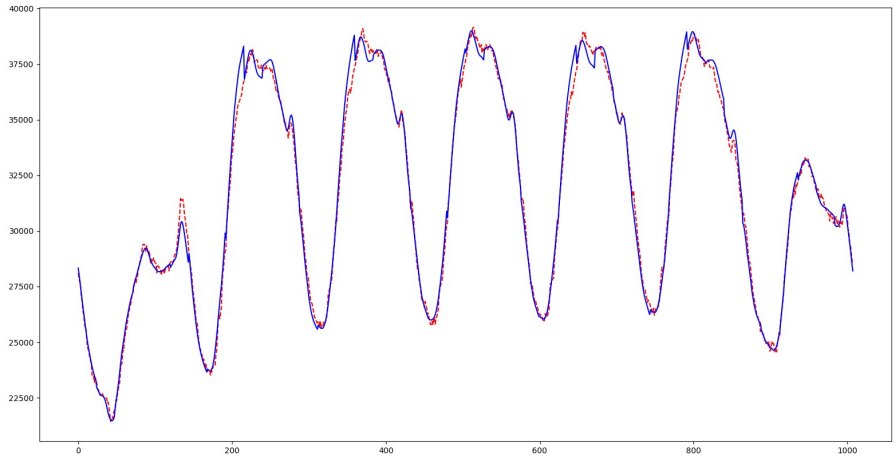


Fig. 8. Close up of the short-term LSTM model prediction for the first week of the test set. The dashed red line is the real data while the blue line is the prediction.



4.2 Long-Term time series forecasting

The results of the recursive prediction over the test set for the LSTM network trained for long-term forecasting is shown in table 2. Also, results of recursive prediction using the models presented in section 4.1 are shown for comparison. In the three of them, the external data have been used. Results are separated in one week ahead recursive prediction, one month ahead and ten months ahead.

The RMSE of both the one week ahead and the one month ahead, are averaged by using 310 different days of the test set as the starting point of the prediction. In the case of the ten months ahead, only 40 days have been used as the starting point. Both, the mean RMSE and the standard deviation are shown for each prediction.

It is to be highlighted too that the application of the long-term LSTM model to short-term forecasting led to a RMSE in the test set of 718.82. Thus the training of the model for long-term presents a worse performance for short-term forecasting as it was expectable.

Table 2. Comparison for Long-term prediction on the test dataset, among the long-term LSTM, the short-term LSTM and the CNN short-term prediction models.

Long Term Prediction			
Horizon	**Short-term model mean RMSE (std)**	**CNN model mean RMSE (std)**	**Long-term model mean RMSE (std)**
one-week-ahead	1366.91 (733.61)	1020.06 (635.74)	925.08 (435.47)
one-month-ahead	1999.79 (634.33)	1500.14 (1063.10)	1127.46 (400.77)
ten-months-ahead	2359.24 (18.06)	2351.03 (933.13)	1286.49 (17.46)

However, as we can seen in table 2, the Long-term model is able to outperform the other two models, especially when the prediction horizon gets longer. The CNN model seems to perform better that the Short-term one for lower horizon predictions, although they get similar results in the ten months ahead prediction. However, the Short-term model shows a higher consistency than the CNN model as the horizon ahead increases, as it can be seen in the standard deviation of the RMSE. In is important to note that the small value for the deviation in the ten months ahead prediction is partly due to the overlapping of almost nine of the months of the test set in the 40 ten months predictions used for the average (which puts more in evidence the bad consistency of the CNN model, since its stardard deviation is still very high).

In the following images we see two close ups of the recursive prediction, figure 9 corresponds to the first week prediction of the test set for all the alternatives, while figure 10 corresponds to the worst prediction for the long-term approach on the test dataset .

Fig. 9. Close up of the prediction over the first week of the test dataset. The three alternatives long-term trained LSTM model, short term LSTM model and CNN short term model are compared.

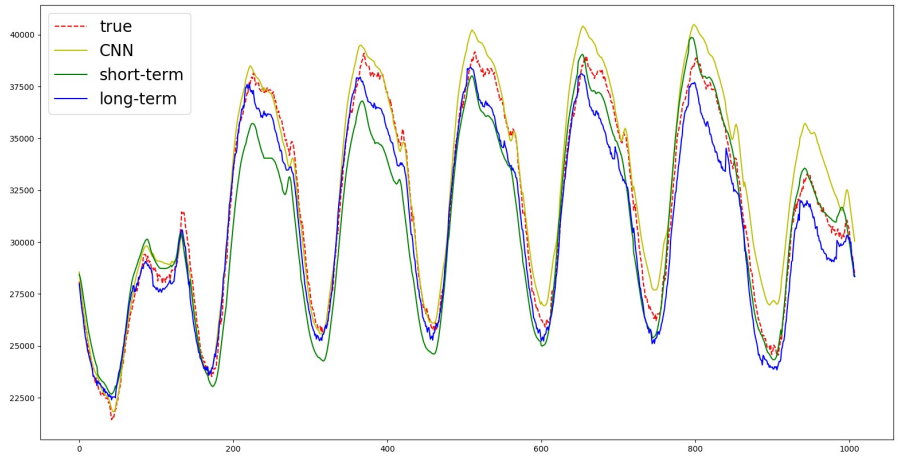
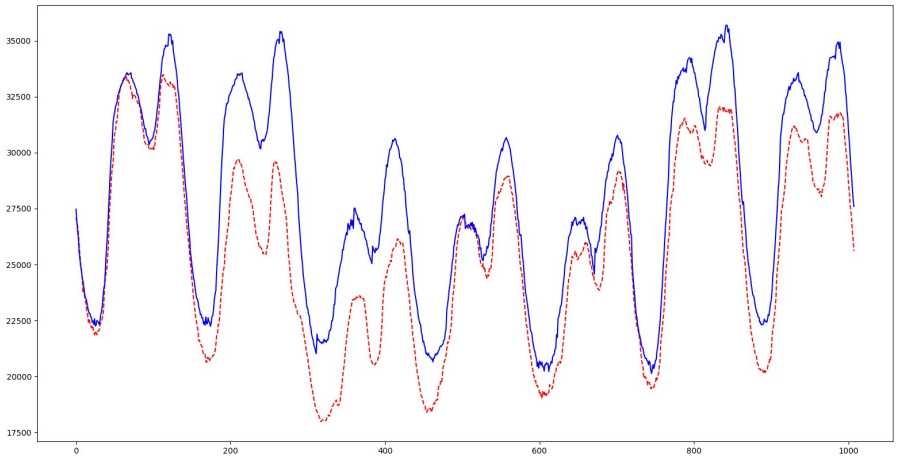


Fig. 10. Close up of the long-term LSTM model prediction over a the worst case on a week ahead prediction. The dashed red line is the real data while the blue line is the prediction.



From the long-term results, it is clear that the long-term recursive prediction is not as good as the normal short-term prediction, which is to be expected.

Nevertheless, the model presents a reasonably good performance since no true consume data has been fed to the network during the pass through the whole test set (aside from the starting point). This shows that the network is able to learn a lot from the external data when it cannot perfectly rely on the previous consume data. This is very important as it implies that having a right prediction on the temperature and precipitation conditions of the future can imply a reasonable power consumption prediction. This can lead, in future works, to a study of the behaviour of the models, and thus of the long-term power consumption under varying weather conditions.

5 Conclusions

This work has presented an adapted deep learning set of approaches for short and long term prediction of a national power consumption time series. Modified LSTM and CNN models for four hours horizon were presented, showing their efficacy in direct 24 values prediction in comparison with single horizon valued models from previous works. Also the improvements by considering external inputs for the prediction are shown. Finally, an alternative LSTM model trained for long-term time series forecasting was assessed, showing important improvements over the previously trained CNN and LSTM models for short term forecasting.

Acknowledgements

This research has been possible thanks to the support of project: RTI2018-101674-B-I00 (Spanish Ministry of Economy and Competitiveness –MINECO– and the European Regional Development Fund. –ERDF).

References

1. https://demanda.ree.es/visiona/home
2. Amarasinghe, K., Marino, D.L., Manic, M.: Deep neural networks for energy load forecasting. In: 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). pp. 1483–1488 (June 2017). https://doi.org/10.1109/ISIE.2017.8001465
3. Angelopoulos, D., Siskos, Y., Psarras, J.: Disaggregating time series on multiple criteria for robust forecasting: The case of long-term electricity demand in Greece. European Journal of Operational Research **275**(1), 252–265 (2019). https://doi.org/10.1016/j.ejor.2018.11.00, https://ideas.repec.org/a/eee/ejores/v275y2019i1p252-265.html
4. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016), http://www.deeplearningbook.org
5. Sagheer, A., Kotb, M.: Time series forecasting of petroleum production using deep lstm recurrent networks. Neurocomputing **323** (10 2018). https://doi.org/10.1016/j.neucom.2018.09.082

6. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3104–3112. Curran Associates, Inc. (2014), http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
7. Torres, J.F., Gutiérrez-Avilés, D., Troncoso, A., Martínez-Álvarez, F.: Random hyper-parameter search-based deep neural network for power consumption forecasting. In: Rojas, I., Joya, G., Catala, A. (eds.) Advances in Computational Intelligence. pp. 259–269. Springer International Publishing, Cham (2019)
8. Torres, J., Galicia de Castro, A., Troncoso, A., Martnez-lvarez, F.: A scalable approach based on deep learning for big data time series forecasting. Integrated Computer-Aided Engineering **25**, 1–14 (08 2018). https://doi.org/10.3233/ICA-180580
9. Yu, R., Gao, J., Yu, M., Lu, W., Xu, T., Zhao, M., Zhang, J., Zhang, R., Zhang, Z.: Lstm-efg for wind power forecasting based on sequential correlation features. Future Generation Computer Systems **93**(Environ. Policy Collect. 2015), 33–42 (2019). https://doi.org/10.1016/j.future.2018.09.054, https://app.dimensions.ai/details/publication/pub.1107641229

Time Series Classification of Automotive Test Drives Using an Interval Based Elastic Ensemble

Felix Pistorius[0000−0002−7727−7206], Daniel Grimm[0000−0003−3743−872X], Marcel Auer[0000−0002−5239−828X], and Eric Sax

Institute for Infomation Processing Technologies,
Karlsruhe Institute of Technology, Karlsruhe, Germany
felix.pistorius@kit.edu
daniel.grimm@kit.edu
marcel.auer@student.kit.edu
eric.sax@kit.edu
https://www.itiv.kit.edu

Abstract. The development of autonomous vehicles leads to an increasing number of necessary test drives with different scenarios and driving maneuvers. These are needed to validate the growing number of functions and to ensure safe driving behaviour. During a test drive, data from a variety of sensors is recorded for vehicle development. A particularly interesting use case for this data is the analysis of individual driving situations of the entire test drive, which are relevant for the refinement of the vehicles under development. For the chassis calibration, for example, it is of special importance to consider track sections with certain surfaces and curve characteristics, since only in these time periods of the recording relevant information for the development can be found. Therefore, this paper presents a method to automatically and robustly retrieve specific sequences in test drives. The method proposed by us is able to identify any desired time segments and thus to select only the relevant segments from the dataset of all test drives. It should be particularly emphasized that our method finds the desired time segments even for datasets that originate from different vehicles, driving styles and routes. This paper therefore makes a valuable contribution to reducing the development effort for vehicles. Our concept is based on an efficient ensemble of the elastic time series classification algorithms Dynamic Time Warping (DTW), Derivative DTW, Weighted DTW and Move-Split-Merge (MSM). Our evaluation results on both synthetic and real-world data are promising.

Keywords: Dynamic Time Warping · Time Series Classification · Ensemble-based classification · Automotive Test Drive.

1 Introduction

Since the introduction of the anti-lock braking system (ABS) in 1978, the number of electronic components in cars has been steadily increasing up to 150 Electronic Control Units (ECUs) in modern cars [14]. Today, the ECUs in a vehicle run

more than 100 million lines of software code that fulfill various functions, for example advanced driver assistence systems [2]. With the development towards autonomous driving, the amount of software and functions continues to grow. In order to validate and verify this amount of software, extensive tests are carried out during the development process. These include test drives in simulated environment as well as real-world test drives. Statistically, with a higher degree of autonomity of the functions that are developed, more test kilometres have to be driven for verification and validation. As an example, for the verification of the "Autobahnpilot" approximately 7 billion km of test drives are required [12]. Vehicles are equipped with different sensors that monitor the behaviour during a drive, for example speed, acceleration or the engines rotation per minute. During a test drive, this data is recorded to leverage the refinement of functions and optimization of vehicle parameters to enable validation. A large part of the information contained does not lie in the single sensor values, but in their change over time. The time series of all sensor signals recorded with the vehicle in a wide variety of tests can provide interesting insights into the driving characteristics and control mechanisms of the vehicle. Hence, the extension of electronic monitoring of various influences on the vehicle not only increases the safety of the vehicle occupants, but also enables a more detailed analysis of vehicle performance during the process of function optimization and represents an increasingly important field of activity.

1.1 Motivation

With regard to the validation of a vehicles functions, the behaviour of the vehicle in specific driving situations such as lane changes or a track section consisting of different curve characteristics are of great interest in function development. These driving situations are each reflected in a distinct signal curve of different sensor signals. In testing of autonomous vehicles a lot of research is focusing on the identification and analysis of such sensor signal sequences, e.g. virtual testing of neural network based driving functions [12]. With regard to real-world test drives an interesting research field is the selection of particular test drives of a huge dataset, that span the greatest possible variety within the selected test drives [8]. In this field of work, the main reason for searching and identifying specific driving situations is to achieve a highest possible coverage of test cases. If a novel driving situation is found in a dataset this situation should be taken into account for testing to ensure test coverage.

Another interesting issue arises when a situation has been found during test in which the function has not yet worked flawlessly. In this case we want to find similar driving situations where the function can be validated or optimized in further suitable test cases. A driving function should behave in the same way in similar situations, independent of the driver or environmental conditions such as weather or the roads surface. Hence, if we optimize and test a function in a well-known condition (e.g. a curve on a test track in a specific test drive), we want to validate the behaviour under similar circumstances in other test drives.

However, most of the concepts that identify specific driving situations in datasets are based on the concept of anomaly or novelty detection. Hence, these approaches find unnkown situations in the dataset [17] [8]. In our case, we want to identify all similar occurances of a well-known driving situation rather than detecting novel situations. Neural networks are frequently used for the problem of novelty detection as well as the identification of similarities, but they require enormous amount of training data. In worst-case we only know one specific test drive in which the driving situation of interest occured. Thus, the amount of relevant data available is not enough to train neural networks. Additionally, neural networks may not be implemented by an engineer without specific knowledge in deep learning. Therefore, we distinguish our work in two main points: First, we want to select similar sequences out of a dataset, rather than novel situations. Second, we want to develop a concept that can be used without deeper knowledge on neural networks and does not require a lot of time and data for training.

In this paper, we present our approach of an automated extraction of characteristic signal patterns from time series datasets. The objective of our approach is to analyse datasets on the basis of a few test drives under known conditions and to efficiently extract a specified driving situation or route section. The driving situation we want to retrieve in a dataset can be specified by an engineer as a particular time interval of interest in our known test drive. Hence, it is given beforehand and can be used as a reference for a classification approach to extract similar driving situations. The classifcation is based on an ensemble of elastic time series algorithms. If the automobile is regarded as a dynamic system, the measurable signals are strongly dependent on external influences, such as the driving style of the driver, the road conditions or the traffic density. On that account, this paper presents an approach that enables classification and identification of similar driving situations independent from the above-mentioned external influences. Robustness of the classification quality against different driving styles or vehicles ensure the wide applicability of our method. An additional benefit of this indipendence is that previous test drives with other vehicles or drivers may be analysed in search of similar driving situations. Another main contribution of this work is that previous knowledge from the test drives can be incorporated into the classification process. For example, if we know that a specific driving situation occurred six times during a test drive, the ensemble classifier extracts exactly six possibly relevant time intervals. We present this application in Sect. 5 on the basis of a drive of a FormulaStudent racing cart.

The structure of this paper is as follows. In Sect. 2 related work in time series classifcation is outlined. More background on classifiers based on elastic distance measures is given in Sect. 3. In Sect. 4 we describe our approach for an ensemble of elastic classifiers to accelerate the classification while ensuring a high precision. The evaluation dataset consisting of both simulated and real world data is presented in Sect. 5, and in Sect. 6 we present and analyse the evaluation results of our ensemble approach. In Sect. 7 we summarise our conclusions, and finally in Sect. 8 we discuss our future work.

2 State of the Art

Algorithms for time series classifcation can be divided into the following categories according to Bagnall et. al. [1]:

- **Time Domain based:** Based on a selected distance measure, the distance of many time series to a certain reference time series is calculated considering all data points of the time series. An elastic distance measure is often used to compensate temporal distortion.
- **Differential based:** Based on the temporal derivation of the time series, similarities with respect gradients are examined for classification.
- **Interval based:** The time series are not classified on the basis of the entire data points, but only under consideration of one or more phase-dependent intervals (e.g. peak or plateau).
- **Form based:** The time series is examined for short sequences of simple forms (e.g. plateau, peak, sine, etc.) representing a class.
- **Dictionary based:** The time series are divided into elementary recurring sequences (words) which are stored in a dictionary. The time series can then be classified according to the sequence and frequency of the words.

Depending on the driving situation, short signal peaks may occur in the signal path, which are characteristic for the respective driving situation to be classified. These can be lost when dictionary based algorithms such as the Bag of Patterns (BOP) are used, since the time series are approximated at intervals using Piecewise Aggregate Approximation (PAA) [10]. This reduces the validity of the classification for time series with high dynamics and variance.

Shapelet-based methods, are not further considered due to their high computational effort compared to the other methods. Though the Fast Shapelet algorithm proposed by Rakthanmanon and Keogh is three orders of magnitudes faster than the respective state-of-the art algorithms, it uses a compression like dictionary based classifiers [13], which may lead to a loss of information necessary for correct classification.

The recording of whole drives and the search for certain sequences in these datasets indicate the use of an interval-based approach, such as Time Series Forests, which splits the time series in smaller sequences. These sequences are classified by decision trees that are based on different thresholds. However, the diversity of the different signal paths in automobiles requires a large amount of training data to create the decision trees in order to achieve a high classification quality [3]. As outlined in Sec. 1, we are not able to use algorithms that require a lot of data for training.

In Sec. 1 we also outlined the external influences on the classification of time series in automotive context such as different vehicles or drivers. These lead to the fact that driving situations that are interesting for classification can proceed faster or slower. Thus, there can be a temporal distortion of the time series to be compared. Therefore, in the automotive context, it makes sense to base the classification on Time Domain Based and Differential Based Classifiers. Due to their ability to ignore stretches in time, they are often called elastic classifiers.

We decided to use an interval based elastic ensemble. This combines the advantages of interval-based and elastic classifiers, since we can only extract the relevant intervals of the time series and use sequences of variable length if a temporal distortion has occurred.

3 Theory of elastic classifiers

Elastic distance measurements allow the calculation of the distance between two sequences to take into account stretching or compression with respect to one dimension. A low distance indicates that the two sequences are similar. According to mathematical definition, elastic distances are symmetrical, non-negative and reflexive, but do not usually fulfill the triangle inequality [4,5]. Since the triangle inequality is not always fulfilled, indirect comparisons of two sequences are not possible. Thus by consideration of the similarity between X and Y, as well as between X and Z no statement can be made about the similarity between Y and Z:

$$X \sim Y \wedge Z \sim Y \nrightarrow X \sim Z. \tag{1}$$

3.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is a method for determining the distance between two time series. It originates from speech recognition, but is now used in many areas of digital signal processing. The basic algorithm goes back to the work of Sakoe and Chiba in 1978 [15]. DTW is based on an elastic distance measure and determines the minimum distance between a reference X and a query Y, taking into account temporal stretching. Here X is the query of length m with

$$X = x_1, x_2, \ldots, x_i, \ldots, x_m \tag{2}$$

and Y the reference of length n with

$$Y = y_1, y_2, \ldots, y_j, \ldots, y_n. \tag{3}$$

The algorithm is based on a $m \times n$ distance matrix D_Δ, in which each element

$$D_\Delta[i,j] = \delta(x_i, y_j) = \|(x_i, y_j)\| = \sqrt{(x_i - y_j)^2} \tag{4}$$

specifies the euclidean distance between the elements x_i and y_j. The most cost-effective path from the first element $D\,[1,1]$ to the last element $D\,[m,n]$ of the distance matrix is known as the warping path. The DTW distance d_{DTW} between the two sequences is calculated using Eq. 5 and can therefore be interpreted as the sum of all elements w_k of the warping path W. A low distance between the reference Y and the query X means that we are able to classfiy X as similiar to Y.

$$d_{DTW}(X,Y) = \sum_{k=0}^{K} w_k \ , \ K \mathrel{\widehat{=}} \text{Length of warping path} \tag{5}$$

If the warping path deviates from the main diagonal of the distance matrix,this is referred to as a time warp. When calculating the distance between two time series, which differ in their amplitude values, the DTW algorithm sometimes produces very pronounced singularities in the assignment. A singularity in the assignment is a point of a time series that is assigned to more than one point of the other time series. [7] In the case of strong intensity, a single point is stretched for a relatively long time in order to find the smallest distance of both sequences. This leads to a distortion of the result, since the query sequence is so strongly altered by the temporal distortion that there is hardly any resemblance to the original query sequence.

3.2 Weighted Dynamic Time Warping

In their paper published in 2011, Jeong et al. pursued the approach of a flattening of the warping path in order to minimize the resulting singularities by introducing a penalty weighting for singular allocations [6]. The algorithm is based on the same distance matrix as when using DTW but with an added penalty. The added penalty weight is dependent on the warping distance $a = |i - j|$ of the time series $X = x_1, x_2, \ldots, x_i, \ldots, x_m$ and $Y = y_1, y_2, \ldots, y_j, \ldots, y_n$. It is calculated using a modified logistic weight function:

$$w(a) = \frac{w_{max}}{1 + e^{-g(a - m_l)}} \text{ mit } m_l = \frac{l}{2} \tag{6}$$

Here w_{max} is the maximum penalty weight, which is freely chosen depending on the application, g the steepness of the logistic function and l the length of the sequence under consideration.

3.3 Derivative Dynamic Time Warping

Derivative Dynamic Time Warping's (DDTW) approach is based on the paper of the same name by Keogh and Pazzani from 2001 [7]. The great weakness of the DTW algorithm is that it can only deal with temporal distortion of the reference signal. If there is an offset or a change in amplitude between the two sequences, the DTW algorithm tries to represent this change by temporal stretching or compression. A point of one sequence is often assigned to many points of the other sequence. One speaks of a singularity in the assignment of the two time series. The problem of strongly pronounced singularities has already been explained in Sect. 3.1. In order to avoid this, DDTW aims at an assignment that is oriented to the local minima and maxima of the two sequences. DDTW proceeds similarly to the DTW algorithm, but uses with

$$x'[i] = \frac{2(x_i - x_{i-1}) + (x_{i+1} - x_{i-1})}{4}. \tag{7}$$

a kind of derivation of the sequences. The derived sequences are then used to calculate the distance matrix. Thus, according to Eq. (7), the same conditions apply to the calculated distance as when using DTW [5]:

$$d_{DDTW}(X, Y) = d_{DTW}(X', Y') \tag{8}$$

3.4 Move-Split-Merge

According to Stefan et. al., the basic idea behind Move-Split-Merge (MSM) is to use three specific operations to align two time series [16]. Each applied operation causes certain costs, which are added up in the course of the process. The costs of the most favorable sequence of operations thus represent the measure of similarity. The three operations are Move, Split and Merge. The move operation changes the value of an element, the split operation adds another element of the same value to the time series after the respective element, and the merge operation deletes the second element of two consecutive elements of the same value.

Since each operation causes costs $C > 0$, the MSM distance satisfies the triangle inequality and is therefore a full metric by mathematical definition, which distinguishes it from other elastic distance measures such as DTW, DDTW, and WDTW [16]. A full metric allows for mathematically correct comparisons of distances such as "distance of X to Y equals three times the distance of Z to Y".

4 Ensemble approach for test drive classification

The dataset to be examined is divided into many query sequences with an approximate length of the reference sequence ($n \approx m$) by means of a sliding window. Each of the sliding windows X is compared to our reference window Y yielding a classification if X is similar to Y. If X is similar to Y, we found our reference time series in the dataset we analyze. In order to take better account of the temporal stretching of the signal, windows of different widths are selected. This is useful, for example, if two vehicles with different speeds are compared. The width L_W of the sliding window depends on the reference sequence of length n with $L_W = \alpha \cdot n$ and $\alpha = [0.75, 1, 1.25]$. This selection of pre-factors α yielded the best results in our evaluation. A fifth of the respective window width was selected here as the step size.

Each individual window as well as the selected reference sequence is Z-normalized, so that they are mean-free ($\mu = 0$) and apply to the standard deviation $\sigma = 1$. This prevents offsets in the signal value from falsifying the calculated distance [11]. As an example, the detection of a hazard braking maneuver based on the vehicle speed at different initial speeds can be mentioned.

The variety of signal characteristics recorded in an automobile can only be classified with a limited quality using a fixed algorithm. The concept of the ensemble is based on the elastic methods DTW, DDTW, WDTW and MSM.

Fig. 1 shows the schematic structure of the ensemble, which is divided into the following steps:

1. **Preselection of the relevant windows:** The DTW distance to the reference sequence is calculated for every window and the sequences with the smallest distance are selected for closer examination using the other algorithms. The calculation time required to calculate the DTW distance of a window mainly depends on the length n of the selected reference, since $n \approx m$. The amount of windows required (l) therefore only affects the complexity of the selection linearly: $\mathcal{O}(lmn) \approx \mathcal{O}(ln^2)$.

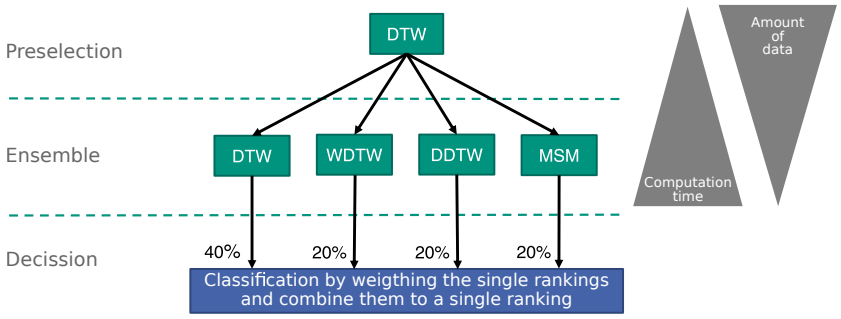


Fig. 1. Ensemble approach for the classification of time series consisting of preliminary selection, classification using DTW, DDTW, WDTW and MSM, and subsequent weighted majority voting

2. **Ensemble classification:** To every window selected in the first step, the DDTW, WDTW and MSM distances are now calculated. Each algorithm creates a ranking of the considered sequences with ascending distances.
3. **Ensemble decision:** The rankings created in the previous step are now weighted and included in the overall ranking of the filtered sequences. The ranking of the DTW distances is included in the decision with 40 % the weight relative to the remaining distances. This is due to the fact that only the DTW algorithm allows a free temporal distortion of the sequences. This would be inferior with equal distribution of the vote weight in the three procedures with limited temporal allocation in each decision.

5 Test Case

To evaluate our ensemble approach, a suited dataset for testing is needed. We use both simulated as well as real-world data. These two are outlined in the following.

5.1 Simulated Test Drive

To evaluate the our ensemble concept, a reproducible test scenario was created and simulated using the simulation environment CarMaker®[1]. The Nürburgring Nordschleife shown in Fig. 2 was chosen as the test track. The track contains many corners, steep inclines, changing road surfaces and, with a length of almost 21 km, offers a dataset for the tests of the ensemble without repetition. One full lap on the track was simulated with different vehicles. The group of selected vehicles includes the luxury sports car Audi R8, the hybrid SUV Lexus NX300h and the small convertible Peugeot 207CC - a wide selection of different cars. The drives of the different vehicles were simulated with different driver profiles. In

[1] CarMaker® is a registered trademark of the IPG Automotive GmbH

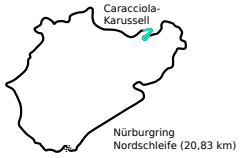


Fig. 2. Map of the Nürburgring Nordschleife with colored Caracciola-Karussel

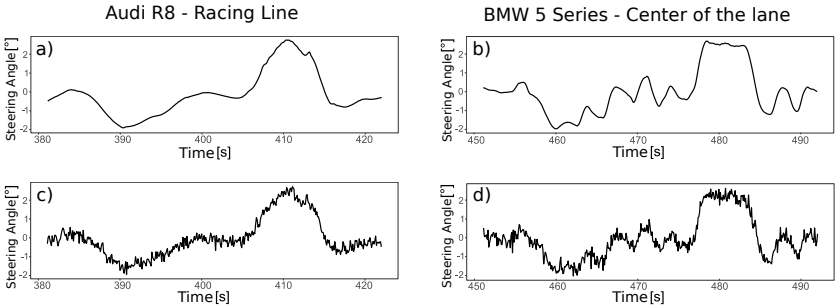


Fig. 3. Passages of the Caracciola-Karussel with Audi R8 and BMW 5 Series. Sequences c) and d) are afflicted with a mean value-free, Gaussian distributed noise (AWGN ($\mu = 0, \sigma = 0.2$)).

CarMaker®, the driver profile determines the maximum speed and how much the corners may be cut (racing line vs. middle of driving lane). By the different vehicles and driver profiles it is ensured that a high variance is contained in our data set, which is supposed to show the robustness of our method.

For the test of our concept, the time series of the steering angle is analyzed to find the passage of a certain section of the route in the dataset, the Caracciola-Karussel. We chose the Caracciola-Karussel as the driving situation of interest for our analysis, since driving in this curve yield high lateral accelerations due to the narrow curve radius. As a consequence, this driving situation may be of particular interest for the optimization of driving functions. As an illustration the passage of the Caracciola-Karussel with the BMW 5 Series is chosen as the reference Y. The BMW's driver profile is set to a passive driving style and follows the middle of the road without cutting the corners. For the evaluation of the results, the drive of the Audi R8 on the Nordschleife is chosen as an example for the examined dataset, since this has the largest differences to the reference regarding the signal shape. This sequence is shown in the top-left of Fig. 3. The Audi drives over the test track with an aggressive driving style on the ideal racing line. For further tests, the recorded data were superimposed with a mean value-free, Gaussian distributed, white noise process. The aim here is to evaluate the influence of noisy data to the quality of the ensemble-based classifcation as this may occur in a real test drive. The sequence of the Audi R8 with added noise can be seen in the bottom-left of Fig. 3. On the right side of Fig. 3, the reference sequence of the BMW 5 Series is shown as a comparison.

5.2 Tests on real-world driving data

The ensemble was not only evaluated with simulated test data, but also with real-world vehicle data. For this purpose, data of an electric cart that takes part in the Formula Student Competition is used. The recorded drive includes several laps on a circuit. Instead of the steering angle, the vehicle speed is taken into account this time and the passage of a certain curve sequence of the circuit is selected as reference. The ensemble is used to extract all passages of this section of the track in the dataset for later examination of the chassis settings.

6 Evaluation of the results

The detection of the Caracciola-Karussel passage, as described in Sect. 5.1, was successfully performed in all simulated test datasets. The passage was also succesfully retrieved using different vehicles and driver profiles. Fig. 4 shows the evaluation results for the comparison of the Audi R8 test drive with added noise we have shown in the bottom-left of Fig. 3. Here, the three windows that were found to be most similar to the reference (BMW 5 Series) are marked. For each time window the corresponding MSM distance to the reference is given. The MSM distance, as described in Sect. 3.4, allows a direct comparison because it fulfills the triangle inequality equation. The time window of the route section searched for (1) has a distance to the reference that is 3.16 (3.93) times shorter than the windows following in the ranking (2 and 3). From this, it can be concluded that the route section searched for can be extracted with great certainty. These results can also be reproduced with any other sections of the circuit.

The test with real-world data also produced positive results. In the first lap of about 180 seconds we selected a specific curve as our reference. In the top of Fig. 5 the reference is marked blue and the first correctly identified occurance in the following lap is marked in green. As can be seen in the bottom of Fig. 5, every occurance of the searched track section could be found in the dataset.

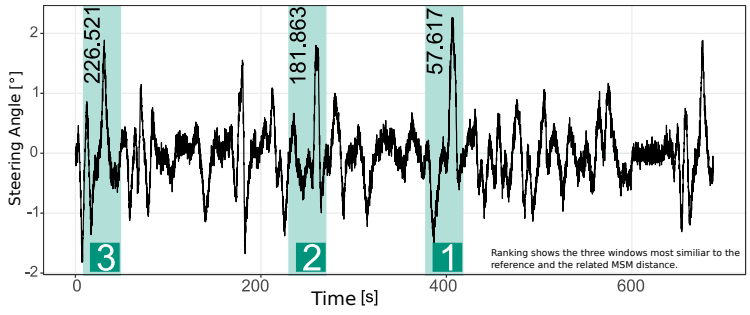


Fig. 4. The best three sequences after the ensemble classification with indication of the corresponding MSM distance. The steering angle data have been superimposed with a mean value-free noise.

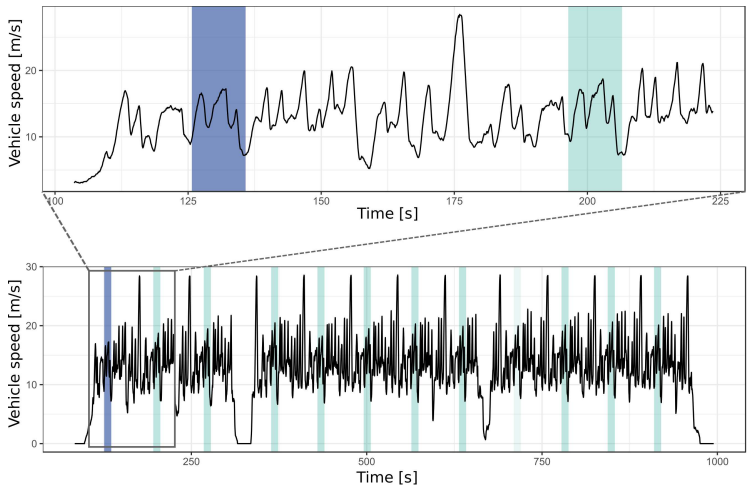


Fig. 5. Application of the ensemble to the vehicle speed recorded in an Formula Student racing cart. The reference is marked blue, the found similar sequences green.

As alternative to our ensemble all four algorithms could have been evaluated on the whole dataset. However we reduced the relevant the relevant time windows with preselection and hence, a reduction of the computing time by a factor of up to 18 could be achieved. An evaluation for different reference sequneces has shown that the speed up is independent of the length of the reference.

7 Conclusion

We proposed an interval based elastic ensemble classifier for time series data. The data is split up into smaller sequences using a sliding window approach which then are compared to a chosen reference. In order to investigate the ensemble regarding classifcation quality and the reduced computing time, a test scenario with different vehicles and driver profiles on the Nordschleife of the Nürburgring was designed in CarMaker®. Furthermore, tests with real-world driving data of a Formula Student electric cart were performed.

The results of these tests of the elastic ensemble showed a high classification quality and could also be reproduced for noisy data. In addition, the implemented preselection of the interesting time windows, compared to the separate application of the considered algorithms, allowed a faster classification up to factor 18. It was shown that time series classification methods, which originate from the field of speech recognition, are suitable for the effective and profitable analysis of vehicle data.

8 Future Work

Currently, the classification of the dataset is based on the observation of a single signal. This univariate classification severely limits the number of classifiable driving situations. The ensemble shall be tested for a multivariate classification of driving situations. Many events in road traffic cannot be classified on the basis of a single sensor signal. Here a form of information fusion is necessary. An overtaking manoeuvre is characterised, for example, by a slight steering deflection to the left, a temporary acceleration and a subsequent slight steering deflection to the right.

During the implementation, possible extensions or changes of the ensemble were identified, which seem promising to further reduce the required computing time. As shown, an appropriate preselection of the interesting time windows when using the ensemble can shorten the computing time by a factor of up to 18. However, since the methods used are all based on distance matrices, it makes sense to run the calculation of these matrices parallel on a GPU instead of serially on the CPU. A further acceleration of up to factor 34 can theoretically be achieved here [9].

Another conceivable application of the ensemble is error detection in data from Hardware-in-the-loop (HIL) test benches. Here, certain physical processes of individual vehicle parts are carried out and monitored again and again in a simulated vehicle environment. An inverted application of the ensemble would be imaginable here, as the ranking would not be based on the smallest but on the largest distance. Thus, the ensemble automatically finds the test runs in which errors have occurred. For example, the material fatigue and failure susceptibility of vehicle doors to repeated opening and closing can be investigated.

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **31**(3), 606–660 (2017). https://doi.org/10.1007/s10618-016-0483-9
2. Charette, R.N.: This car runs on code. IEEE spectrum **46**(3), 3 (2009)
3. Deng, H., Runger, G., Tuv, E., Vladimir, M.: A time series forest for classification and feature extraction. Information Sciences **239**, 142–153 (2013). https://doi.org/10.1016/j.ins.2013.02.030
4. Deza, M.M., Deza, E.: Encyclopedia of Distances. Springer Berlin Heidelberg, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52844-0
5. Górecki, T., Łuczak, M.: Multivariate time series classification with parametric derivative dynamic time warping. Expert Systems with Applications **42**(5), 2305–2312 (2015). https://doi.org/10.1016/j.eswa.2014.11.007
6. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern Recognition **44**(9), 2231–2240 (2011). https://doi.org/10.1016/j.patcog.2010.09.022
7. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: Grossman, R. (ed.) Proceedings of the First SIAM International Conference on Data Mining, April 5 - 7,

2001, Chicago, IL, USA, pp. 1–11. Science and industry advance with mathematics, SIAM, Philadelphia, Pa. (2001). https://doi.org/10.1137/1.9781611972719.1
8. Langner, J., Bach, J., Ries, L., Otten, S., Holzäpfel, M., Sax, E.: Estimating the uniqueness of test scenarios derived from recorded real-world-driving-data using autoencoders. In: 2018 IEEE Intelligent Vehicles Symposium (IV). pp. 1860–1866 (June 2018). https://doi.org/10.1109/IVS.2018.8500464
9. Li, Q., Kecman, V., Salman, R.: A chunking method for euclidean distance matrix calculation on large dataset using multi-gpu. In: Drăghici, S. (ed.) The Ninth International Conference on Machine Learning and Applications. pp. 208–213. IEEE Computer Society, Los Alamitos, Calif (2010). https://doi.org/10.1109/ICMLA.2010.38
10. Lin, J., Li, Y.: Finding structural similarity in time series data using bag-of-patterns representation. In: Winslett, M. (ed.) Scientific and statistical database management, Lecture Notes in Computer Science, vol. 5566, pp. 461–477. Springer, Berlin (2009). https://doi.org/10.1007/978-3-642-02279-1_33
11. Mueen, A., Keogh, E.J.: Extracting optimal performance from dynamic time warping. In: KDD 2016, pp. 2129–2130 (2016)
12. Pfeffer, R., Ukas, P., Sax, E.: Potential of virtual test environments for the development of highly automated driving functions using neural networks. In: Bertram, T. (ed.) Fahrerassistenzsysteme 2018. pp. 203–211. Springer Fachmedien Wiesbaden, Wiesbaden (2019)
13. Rakthanmanon, T., Keogh, E.: Fast shapelets: A scalable algorithm for discovering time series shapelets. In: Gosh, J. (ed.) Proceedings of the 2013 SIAM International Conference on Data Mining, pp. 668–676. SIAM, Society for Industrial and Applied Mathematics, [Philadelphia, PA] (2013). https://doi.org/10.1137/1.9781611972832.74
14. Reif, K.: Automotive Mechatronics. Springer Fachmedien Wiesbaden, Wiesbaden (2015). https://doi.org/10.1007/978-3-658-03975-2
15. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978). https://doi.org/10.1109/TASSP.1978.1163055
16. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE Transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (2013). https://doi.org/10.1109/TKDE.2012.88
17. Theissler, A., Dear, I.: Detecting anomalies in recordings from test drives based on a training set of normal instances. In: Proceedings of the IADIS International Conference Intelligent Systems and Agents 2012 and European Conference Data Mining. pp. 124–132 (01 2012)

6h International Conference on time Series and Forecasting
ITISE 2019
25th – 27th September 2019
Granada – Spain

Modeling recession curves in a karstic aquifer

Gonzalez-Herrera R.A., Zetina-Moguel C.E., Sanchez y Pinto I.,
Casares-Salazar R.

ABSTRACT

The analysis of recession curves (RC) is used in research, administration and management of water resources. In karst systems it has been used to study the dynamics of groundwater, particularly in the process of groundwater discharge during the dry season. There is a diversity of models, published in the literature, based on different physical principles; examples of them are the Boussinesq (BM) and Maillet (MM) models, which were derived from the theoretical equations of groundwater flow.

A study was conducted whose objective was to determine the model that best represents the hydrodynamics of a karst aquifer located in a plane area. To achieve the above, statistical criteria were defined to select the model; a statistical analysis strategy was implemented to compare the estimates of the parameters of greatest hydrological interest of the chosen decay models. The data used come from three wells located in the northern plain of Yucatan, Mexico, where hydraulic heads were daily measured for different periods of time.

The scenarios considered for the analysis were: I.- Two wells with the same time intervals with standardized data of: different initial hydraulic head (ho); ho similar; ho modified. II.- A single well, starting from an established hydraulic head and with the same period of discharge time. The adjustment of the models to the observed RCs was carried out by means of numerical methods and parameters estimation by means of a maximum likelihood function. To choose the appropriate model, the following statistical criteria were established: the sum of squared errors (SSE); the adjusted coefficient of determination (R2adj); a function of maximum likelihood and the Akaike criterion.

For the Boussinesq model, estimates of the α parameter were explored, using fixed values of the parameter n, in the range of 0.5 to 5. The comparison between the estimates of α was made based on hypothesis tests (t to compare between two values of α and F for the comparison between more than two values of α); quasiprobability distributions obtained by a maximum likelihood estimator were also used. The Boussinesq model proved to be the most appropriate. The best estimates of α (BM) for hydrological interpretations are obtained with values of the parameter n between 0.5 and 1.5.

The HJ-Biplot Visualization of the Singular Spectrum Analysis Method

Alberto Oliveira da Silva (✉) [0000-0002-3496-6802] , Adelaide Freitas[0000-0002-4685-1615]

Department of Mathematics - University of Aveiro
Aveiro, 3810-193, Portugal
albertos@ua.pt

Abstract. Time series data usually emerge in many scientific domains. The extraction of essential characteristics of this type of data is crucial to characterize the time series and produce, for example, forecasts. In this work, we take advantage of the trajectory matrix constructed in the Singular Spectrum Analysis, as well as of its decomposition through the Principal Component Analysis via Partial Least Squares, to implement a graphical display employing the Biplot method. In these graphs, one can visualize and identify patterns in time series from the simultaneous representation of both rows and columns of such decomposed matrices. The interpretation of various features of the proposed biplot is discussed from a real-world data set.

Keywords: Singular Spectrum Analysis, NIPALS algorithm, Biplots.

1 Overview

Singular Spectrum Analysis (SSA) is a non-parametric method and a suitable tool to perform exploratory analysis on time series [6]. The Basic SSA schema is the version that deals with the description and identification of the structure of a one-dimensional real-valued time series. Basic SSA can be described as two successive stages: *decomposition* and *reconstruction*. The first one is subdivided into step 1, the *embedding*, and step 2, the *Singular Value Decomposition* (SVD), while the second consists of two other phases, the *grouping* and the *diagonal averaging*. The primary purpose is to decompose the original time series into the sum of a few interpretable components, such as trend, oscillatory shape (e.g., seasonality) which should be separated from a noise component [5].

For any matrix, the factorization given by SVD allows practical graphical representations of both rows and columns of the matrix employing biplots methods [2, 3]. Biplots provide easier interpretations, are much more informative than the traditional scatterplots, and might facilitate the work in the grouping step in SSA. Several types of biplots can be constructed depending on how the three factors identified by SVD are aggregated to obtain only two factors. Herein, the option is the biplot method proposed by Galindo [3], called HJ-biplot, which yields a simultaneous representation of both rows and columns of a matrix of interest with maximum quality [3].

The main objective of this paper is to propose a new exploratory procedure to visualize and identify patterns in the time series through the construction of an HJ-biplot from the results of the SVD step on the Basic SSA. Moreover, this work suggests an alternative approach to obtain the factorization referred in step 2 (first stage) based on the *Nonlinear Iterative Partial Least Squares* (NIPALS) algorithm [11] instead of the usual SVD method. Although it provides equivalent results concerning the singular vectors and the singular values, it empowers the SSA to deal with missing values in the data, without employing any imputation method, since NIPALS is a suitable tool to treat this problem [10, 12]. That occurs because, in each iteration of the NIPALS algorithm, only present data are considered in the regressions performed, ignoring the missing elements. This is equivalent to defining all missing points in the least squares objective function as zero.

The paper is organized as follows. In Section 2, we provide a short description of the theoretical background related to the methods involved in this work. In Section 3, we propose a biplot approach to the SSA method and some possible interpretations of it. In Section 4, we perform an application of the proposed technique by using real-world data set. Final conclusions are contained in Section 5.

2 Methods

2.1 Basic Singular Spectrum Analysis

The Basic SSA is a model-free tool used to recognize and identify the structure of a time series [5]. As before mentioned, it is composed of two complementary stages, as follows.

First Stage: Decomposition.
Consider a real-valued time series $\mathrm{Y} = (y_1, \dots, y_N)$ of length N. Let the integer value L $(1 < L < N)$ be the so-called *window length*, as well as $K = N - L + 1$. Hereupon, the *embedding* procedure, that is the first step of the Basic SSA, consists in representing Y in K lagged vectors, $\mathbf{x}_1, \dots, \mathbf{x}_K$, each one of size L (L-lagged vectors), i.e., $\mathbf{x}_j = (y_j, \dots, y_{j+L-1})$, $1 \leq j \leq K$. This sequence of K vectors forms the trajectory matrix $\mathbf{X} = [\mathbf{x}_1 : \dots : \mathbf{x}_K]$, that has as its columns the L-lagged vectors. Step 2, the SVD step, results in the singular value decomposition of the trajectory matrix. Consider that rank($\mathbf{X}$) is equal to d, and the matrix $\mathbf{S}$ is defined as the product $\mathbf{X}'\mathbf{X}$. So, the SVD of $\mathbf{X}$ is the decomposition in the form

$$\mathbf{X} = \sum_{i=1}^{d} \sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i', \tag{1}$$

where λ_i, $i = 1, \dots, d$, are the eigenvalues of the matrix $\mathbf{S}$ arranged in decreasing order of magnitudes ($\lambda_i > 0$), $\{\mathbf{v}_1, \dots, \mathbf{v}_d\}$ is the orthonormal system of the eigenvectors of $\mathbf{S}$ associated with the eigenvalues $\lambda_1, \dots, \lambda_d$, and

$$\mathbf{u}_i = \mathbf{X}\mathbf{v}_\mathbf{i}/\sqrt{\lambda_i}. \tag{2}$$

The elements of the triple $\sqrt{\lambda_i}, \mathbf{u}_i, \mathbf{v}_i$ are also known as *singular values*, *left* and *right singular vectors* of **X**, respectively. Besides, defining

$$\mathbf{X}_\mathrm{i} = \sqrt{\lambda_i}\mathbf{u}_i\mathbf{v}_i', \tag{3}$$

one can represent **X** as a sum of d 1-rank matrices, i.e.,

$$\mathbf{X} = \mathbf{X}_1 + \cdots + \mathbf{X}_d. \tag{4}$$

Second Stage: Reconstruction.
Once the expansion (4) has been determined, the third step of the SSA starts with the partitioning of the index set $\{1, \dots, d\}$ into disjoints subsets $I_j, j = 1, \dots, p$. Let

$$\mathbf{X}_I = \sum_{i \in I} \mathbf{X}_i \tag{5}$$

and the decomposition can be written as

$$\mathbf{X} = \mathbf{X}_{I_1} + \cdots + \mathbf{X}_{I_p}. \tag{6}$$

The intention of the grouping procedure is the separation of the additive components of the time series [6]. The objective of the next phase, the *diagonal averaging* step, is to transform each matrix of the *grouping* decomposition into a new time series [5]. At this point, as in [6], it is convenient to define: $\mathbb{M}_{L,K}$ as the space of the matrices of dimension $(L \times K)$; $\mathbb{M}_{L,K}^{(H)}$ the space of Hankel matrices of dimension $(L \times K)$; the embedding operator $\mathcal{T}: \mathbb{R}^N \mapsto \mathbb{M}_{L,K}$ as $\mathcal{T}(Y) = \mathbf{X}$; and the projector $\mathcal{H}$ of $\mathbb{M}_{L,K}$ to $\mathbb{M}_{L,K}^{(H)}$, that carries out the projection by changing entries on auxiliary diagonals (where $i + j$ is a constant) to their averages along the diagonal. So, the diagonal averaging procedure corresponds to obtaining

$$\tilde{Y}^{(k)} = \mathcal{T}^{-1}[\mathcal{H}(\mathbf{X}_{I_k})] \tag{7}$$

and, then

$$Y = \sum_{k=1}^{p} \tilde{Y}^{(k)}. \tag{8}$$

2.2 PCA through NIPALS

The NIPALS algorithm belongs to the Partial Least Squares family, a set of iterative algorithms that implement a wide range of multivariate explanatory and exploratory techniques. The NIPALS is designed as an iterative estimation method for Principal Component Analysis (PCA), that computes the principal components through an iterative sequence of simple ordinary least squares regressions [10, 11]. It produces a singular value decomposition (SVD) of a matrix regardless of its dimensions and the presence of missing data [10]. Again, considering that the trajectory matrix has rank d, the method decomposes **X** as a sum of d 1-rank matrices in terms of the outer product of two vectors, a score $\mathbf{t}_i$ and a loading $\mathbf{p}_i$, so that

$$\mathbf{X} = \mathbf{t}_1\mathbf{p}_1' + \cdots + \mathbf{t}_d\mathbf{p}_d'. \quad (9)$$

The elements of the scores vector $\mathbf{t}_i$ are the projections of the sample points on the principal component direction, while each loading in $\mathbf{p}_i$ is the cosine of the angle between the component direction vector and a variable axis [4]. The NIPALS first computes $\mathbf{t}_1$ and $\mathbf{p}_1$ from $\mathbf{X}$ and, then, the outer product $\mathbf{t}_1\mathbf{p}_1'$ is subtracted from $\mathbf{X}$ to calculate the residual matrix $\mathbf{E}_1$. After, $\mathbf{E}_1$ is used to compute $\mathbf{t}_2$ and $\mathbf{p}_2$, and the residual $\mathbf{E}_2$ is calculated subtracting $\mathbf{t}_2\mathbf{p}_2'$ from $\mathbf{E}_1$, and so on until to obtain $\mathbf{t}_d$ and $\mathbf{p}_d$. The NIPALS algorithm is shown in Algorithm 1.

Algorithm 1. NIPALS internal relations.

NIPALS

Input: $\mathbf{E_0} = \mathbf{X}$
Output: $\mathbf{P} = |\mathbf{p_1}: \ldots: \mathbf{p}_d|, \mathbf{T} = |\mathbf{t_1}: \ldots: \mathbf{t}_d|$
for all $\boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{d}$ **do**
step 0: initialize $\mathbf{t}_i$
step 1:
repeat
step 1.1: $\mathbf{p}_i = \mathbf{E}_{i-1}'\mathbf{t}_i/\mathbf{t}_i'\mathbf{t}_i$
step 1.2: $\mathbf{p}_i = \mathbf{p}_i/\|\mathbf{p}_i\|$
step 1.3: $\mathbf{t}_i = \mathbf{E}_{i-1}\mathbf{p}_i$
until convergence of $\mathbf{p}_i$
step 2: $\mathbf{E}_i = \mathbf{E}_{i-1} - \mathbf{t}_i\mathbf{p}_i'$
end for

From the internal relations in each iteration of the NIPALS algorithm, and after normalizing $\mathbf{t}_i$, such that

$$\mathbf{t}_i^* = \mathbf{t}_i/\|\mathbf{t}_i\| \Leftrightarrow \mathbf{t}_i = \sqrt{\mathbf{t}_i'\mathbf{t}_i}\,\mathbf{t}_i^*, \quad (10)$$

the following equations can be verified [10]:

$$\mathbf{E}_{i-1}'\mathbf{E}_{i-1}\mathbf{p}_i = \lambda_i\mathbf{p}_i \quad (11)$$

$$\mathbf{E}_{i-1}\mathbf{E}_{i-1}'\mathbf{t}_i^* = \lambda_i\mathbf{t}_i^*, \quad (12)$$

where $\lambda_i = \mathbf{t}_i'\mathbf{t}_i$ is the eigenvalue of both matrices $\mathbf{E}_{i-1}'\mathbf{E}_{i-1}$ and $\mathbf{E}_{i-1}\mathbf{E}_{i-1}'$, as well as $\mathbf{p}_i$ and $\mathbf{t}_i^*$ are their corresponding eigenvectors. Thus, the NIPALS decomposition of $\mathbf{X}$ can be written as

$$\mathbf{X} = \sqrt{\mathbf{t}_1'\mathbf{t}_1}\,\mathbf{t}_1^*\mathbf{p}_1' + \cdots + \sqrt{\mathbf{t}_d'\mathbf{t}_d}\,\mathbf{t}_d^*\mathbf{p}_d'. \quad (13)$$

Now, define the matrix $\mathbf{\Sigma}$ as a diagonal matrix containing the singular values $\sqrt{\mathbf{t}_i'\mathbf{t}_i}$ arranged in decreasing order. So, one can write the matrix form of the expansion (13) as

$$\mathbf{X} = \mathbf{T}^*\mathbf{\Sigma}\mathbf{P}', \quad (14)$$

where $\mathbf{T}^*$ is the scores matrix whose column vectors $\mathbf{t}_i^*$ are orthonormal, and $\mathbf{P}$ is the loadings matrix whose column vectors $\mathbf{p}_i$ are also orthonormal.

2.3 HJ-Biplot

The term biplot is due to Gabriel [2] and is associated to a graphical representation that reveals essential characteristics of multivariate data structure, e.g., patterns of correlations between variables or similarities between observations [7]. Consider a target data matrix $\mathbf{Z}$ of dimension $(I \times J)$, and its decomposition in the form

$$\mathbf{Z} = \mathbf{A}\mathbf{B}', \tag{15}$$

where $\mathbf{A}$ is a matrix of dimension $(I \times Q)$, and $\mathbf{B}$ is a matrix of dimension $(J \times Q)$. The matrices $\mathbf{A}$ and $\mathbf{B}$ create two sets of points, and if $Q = 2$, then the rows and columns of $\mathbf{Z}$ can be simultaneously represented into a two-dimensional graph called biplot, in which the rows of $\mathbf{A}$ are reproduced by points and the columns of $\mathbf{B}'$ are expressed as vectors connected to the origin (arrows). Thus, the biplot displays the row markers $\mathbf{a}_1, \dots, \mathbf{a}_I$ of $\mathbf{Z}$, as well as its column markers $\mathbf{b}_1, \dots, \mathbf{b}_J$, so that the inner product $\mathbf{a}_i'\mathbf{b}_j$ is the element z_{ij} of $\mathbf{Z}$ [8]. Very briefly, the interpretation of the biplot representation can be performed as follows:

1. The distance between points corresponds to how different the associated individuals are (dissimilarities), mainly if they are well represented;
2. The size of the arrow is proportional to the standard deviation of the associated variable. The longer the arrow, the greater the standard deviation;
3. The cosine of the angle between arrows approximates the correlation between the variables they represent. Thus, if the angle is next to 90° it indicates a poor correlation, while an angle close to 0° or 180° suggests a strong correlation, being positive in the first case and negative in the other.

The most popular biplot is the classic one [2], in which the metric of the columns is preserved. This version is also designated by GH-biplot [8]. An essential property of the GH-biplot is that the biplot vectors have the same configuration of the data matrix columns and the quality of representation of columns is maximum. By choosing row and column markers properly, the HJ-biplot allows representing the rows and columns simultaneously in the same Euclidean space with optimal quality for both [3].

To construct an HJ-biplot version based on NIPALS instead of SVD as proposed in [3], it's enough to demonstrate the relationship between $\mathbf{t}_i^*$ and $\mathbf{p}_i$, as will be done next.

From the equation (12), multiplying it to the left by $\mathbf{E}_{i-1}'$, it becomes

$$\mathbf{E}_{i-1}'\mathbf{E}_{i-1}(\mathbf{E}_{i-1}'\mathbf{t}_i^*) = \lambda_i(\mathbf{E}_{i-1}'\mathbf{t}_i^*) \tag{16}$$

Next, the vector normalization of $(\mathbf{E}_{i-1}'\mathbf{t}_i^*)$ results in $\mathbf{E}_{i-1}'\mathbf{t}_i^*/\sqrt{\mathbf{t}_i'\mathbf{t}_i}$, i.e., the vector $\mathbf{p}_i$. Proceeding in the same way with respect to equation (11), and multiplying it to the left by $\mathbf{E}_{i-1}$ we have

$$\mathbf{E}_{i-1}\mathbf{E}_{i-1}'(\mathbf{E}_{i-1}\mathbf{p}_i) = \lambda_i(\mathbf{E}_{i-1}\mathbf{p}_i). \tag{17}$$

After, $(\mathbf{E}_{i-1}\mathbf{p}_i)$ is normalized, which produces $\mathbf{E}_{i-1}\mathbf{p}_i/\sqrt{\mathbf{t}_i'\mathbf{t}_i}$, i.e., the vector $\mathbf{t}_i^*$. Hence,

$$\sqrt{\mathbf{t}_i'\mathbf{t}_i}\mathbf{p}_i = \mathbf{E}_{i-1}'\mathbf{t}_i^*, \tag{18}$$

and

$$\sqrt{\mathbf{t}_i'\mathbf{t}_i}\mathbf{t}_i^* = \mathbf{E}_{i-1}\mathbf{p}_i. \tag{19}$$

To unify the biplot axes scales similarly to what is done in [3], the following designation is done

$$\mathbf{a}_i = \mathbf{E}_{i-1}\mathbf{p}_i = \sqrt{\mathbf{t}_i'\mathbf{t}_i}\mathbf{t}_i^* \tag{20}$$

$$\mathbf{b}_i = \mathbf{E}_{i-1}'\mathbf{t}_i^* = \sqrt{\mathbf{t}_i'\mathbf{t}_i}\mathbf{p}_i. \tag{21}$$

Substituting (18) into (20), it follows that

$$\mathbf{a}_i = \mathbf{E}_{i-1}\mathbf{b}_i/\sqrt{\mathbf{t}_i'\mathbf{t}_i}, \tag{22}$$

and plugging (20) in (21) we get

$$\mathbf{b}_i = \mathbf{E}_{i-1}'\,\mathbf{a}_i/\sqrt{\mathbf{t}_i'\mathbf{t}_i}. \tag{23}$$

Thus, from (22) and (23), the coordinates of the *i*-th column are expressed as a function of the coordinates of the *i*-th row and vice versa. As a consequence, it allows the representation of the rows and columns in the same Cartesian coordinates system. Moreover, these expressions of the column and row coordinates lead to the maximum quality of the representation for rows and columns in the same system [3]. Considering the matrix form of the NIPALS decomposition in (14), it is worth to mention that for the configuration of the HJ-biplot, we have

$$\mathbf{A} = \mathbf{T}^*\mathbf{\Sigma}, \tag{24}$$

$$\mathbf{B} = \mathbf{P}\mathbf{\Sigma}, \tag{25}$$

and so,

$$\mathbf{X} \neq \mathbf{A}\mathbf{B}'. \tag{26}$$

3 The SSA-HJ-Biplot

The trajectory matrix that will be decomposed by the NIPALS algorithm at the second step of the first stage of the SSA has some peculiarities in relation to the usual multivariate data matrix. Instead of individuals and variables, the rows and columns of the trajectory matrix represent L-lagged and K-lagged vectors of a time series, respectively. That said, after the decomposition of $\mathbf{X}$, a row marker in the HJ-biplot denotes a K-lagged vector and is depicted in the graph as a point. In turn, a column marker repre-

sents a L-lagged vector, being that an arrow symbolizes it. One of the goals of SSA-HJ-biplot is to assist in the grouping step and for this, building more than one SSA-HJ-biplot may be needed. The first SSA-HJ-biplot uses the 1st and the 2nd principal components (PC), the next one uses the 2nd PC and the 3rd PC, and so on as long as the remain components can explain the variability of the data, which is given by

$$PC_{\%}^{(i)} = \mathbf{t}_i'\mathbf{t}_i / \sum_{j=1}^{d} \mathbf{t}_j'\mathbf{t}_j, \tag{27}$$

or visually through the scree plot of the singular values ($\sqrt{\mathbf{t}_i'\mathbf{t}_i}$) [5].

The window length L has to be large enough so that each L-lagged vector captures a substantial part of the behavior of the time series [5], but at the same time, it permits the interpretability of the graphics display. A window length equals to $N/2$ provides both capabilities because it allows for the most detailed decomposition [5]. The interpretation of the first SSA-HJ-biplot is performed in terms of:

1. The proximity of points. Biplot points whose Euclidean distances are small imply similarity in the behavior of the associated K-lagged vectors;
2. The length of the biplot vectors. If the arrows are roughly the same size, this indicates that the L-lagged vectors have standard deviation also close, which suggests that the process is stationary in the variance;
3. The angle formed between biplot vectors. If the angle between the two arrows is next to 0°, it hints a strong and positive autocorrelation between the two L-lagged vectors associated (negative if next to 180°). If the angle is close to 90°, it is expectable an autocorrelation near to zero.

It is worth to take in mind the percentage of explained variability represented by the first two components, since the higher the percentage, the better the quality of the adjust of the SSA-HJ-biplot [3].

As a rule, a singular value represents the contribution of the corresponding PC in the form of the time series. As the tendency generally characterizes the shape of a time series, its singular values are higher than the others, that is, they are the first eigenvalues [1]. On the other hand, when two singular values are close enough, i.e.,

$$\sqrt{\mathbf{t}_i'\mathbf{t}_i} \approx \sqrt{\mathbf{t}_h'\mathbf{t}_h},$$

this is an evidence of the formation of plateaus in the scree plot and indicates that the associated SSA-HJ-biplot is informative about the oscillatory components of the time series [5], as long as the PC explain high variability of the data.

4 Example

In this Section, an SSA-HJ-biplot is constructed to a time series that contains the records the carbon dioxide concentration in the Earth's atmosphere, measured monthly from January of 1965 to December of 1980 at an observing station on Mauna Loa in Hawaii [9], referred as T.S. CO2 in this work and that is represented in **Fig. 1**. Two auxiliary plots in **Fig. 2** provide some hints for what to expect in an SSA-HJ-biplot analysis in the data. In **Fig. 2** (b), where the 1st PC is plotted against an index $j =$

$1, \ldots, K$, the presence of a trend component in T.S. CO2 is manifest, and this should emerge somehow in the first SSA-HJ-biplot, i.e., in the biplot where the axes are the 1st and 2nd PCs. **Fig. 3** brings the first SSA-HJ-biplot, where one can verify that the 1st PC explains 67% of the data variability, i.e., the trend direction. A channel formed by two dotted lines helps in the perception of the presence of the trend, although the 2nd PC contributes to attenuate the slope if compared with the plot in **Fig. 2** (b). Each one of the biplot points (in red) represents a K-lagged vector, and its corresponding label indicates the month in which the lagged vector starts. In this sense, accordingly to the graph legend, a point labeled as "O" means a K-lagged vector starting in October of some year, and a label "D" symbolizes that the respective K-lagged vectors begins in December, and so on. These points are the row markers, determined by the rows of $\mathbf{T}^*\mathbf{\Sigma}$, that is $\mathbf{a}_i' = \mathbf{t}_i'$, $i = 1, \ldots, L$.

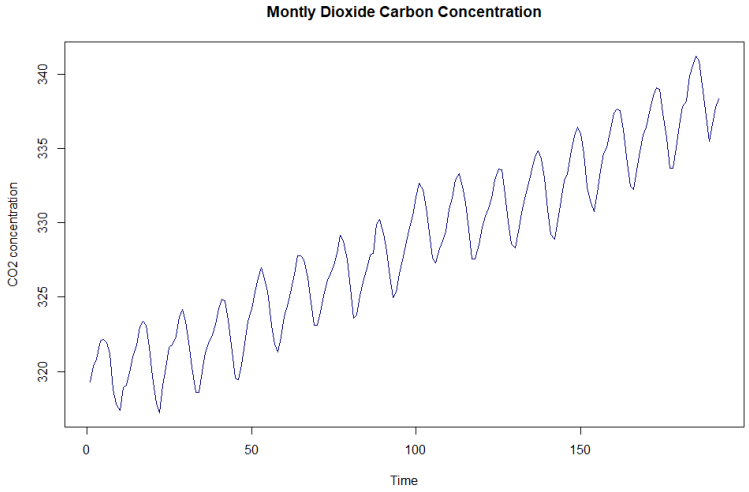


Fig. 1. Carbon dioxide concentration in the Earth's atmosphere measured monthly from January of 1965 to December of 1980 at an observing station on Mauna Loa in Hawaii.

According to the biplot theory, near points indicate similarity in the behavior of the lagged vectors, e.g., the points tagged as A, Y, and U in **Fig. 3**, i.e., the K-lagged vectors starting in April, May, and June. But not only that. Considering the labeling procedure before mentioned, the SSA-HJ-biplot is also capable of capturing the behavior of the months, since April, May, and June correspond precisely to the periods in which the highest concentration of carbon dioxide occurs in the atmosphere. It means that the points in the first SSA-HJ-biplot can represent not only the K-lagged vectors that start in a given month but also the month itself.

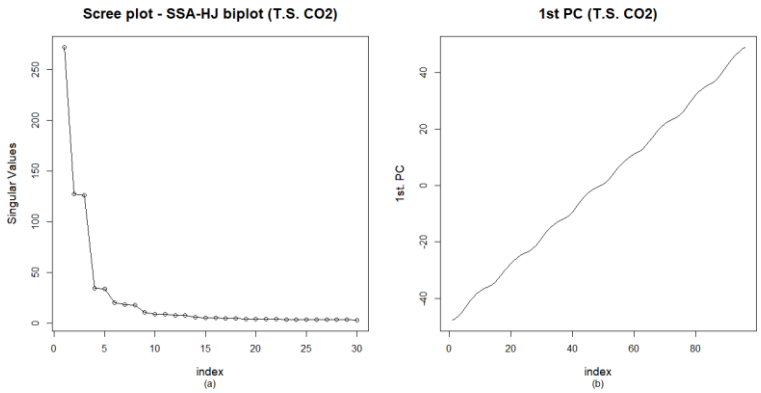


Fig. 2. Auxiliary plots in the SSA-HJ-biplot analysis.

In **Fig. 3**, the SSA-HJ-biplot represents the column markers (the L-lagged vectors) as black arrows up to the sixth L-lagged vector (tagged as $L1$ until $L6$), ordered from top to bottom. From the seventh L-lagged vector onwards the pattern repeats itself, and so they were plotted in gray. It means that the first group of arrows, which is at the top, refer to the L-lagged vectors beginning in January and July, just below those as starting in February and August, and so on. The angle between two consecutive arrows Li and Lj, such that $i = 1, \dots, 5$ and $j = i + 1$, indicates a strong autocorrelation between the respective L-lagged vectors since Li and Lj form very sharp angles. As for $L1$ and the others up to $L6$, the angles range from something close to 0 to something close to 90 degrees, which suggests a fading of the autocorrelations. And this cycle repeats from $L7$ periodically, which suggests the non-stationarity also in the seasonality.

Fig. 4 shows the SSA-HJ-biplot formed by the 2nd and 3rd PCs, while **Fig. 5** exhibit the SSA-HJ-biplot constructed from the 4th and 5th PCs. Along with the first SSA-HJ-biplot, these are the only ones that produce interpretable results or evidence some pattern in the time series, being that these results are in agreement with the one verified in the scree plot of the singular values in **Fig. 2** (a), where the pair of points related to $\sqrt{\mathbf{t}_2'\mathbf{t}_2}$ and $\sqrt{\mathbf{t}_3'\mathbf{t}_3}$ are around at the same level, the same with respect to $\sqrt{\mathbf{t}_4'\mathbf{t}_4}$ and $\sqrt{\mathbf{t}_5'\mathbf{t}_5}$. In the SSA-HJ-biplot of **Fig. 4**, there are well defined 12 groups of row markers, where each one of these groups refers to a K-lagged vector that starts for a specific month. Also, the column markers associated with each one of these groups show strong autocorrelation between the L-lagged vectors. All of this indicates a seasonal pattern, with peaks and valleys separated by 12 months. In turn, the SSA-HJ-biplot of **Fig. 5** groups the lagged vectors two by two, e.g., January and July, February and August, and so on. Interpreting this together with the biplot of **Fig. 4**, where these same groups occur but in the opposite directions, one can conclude that the valleys tend to be six months behind the peaks.

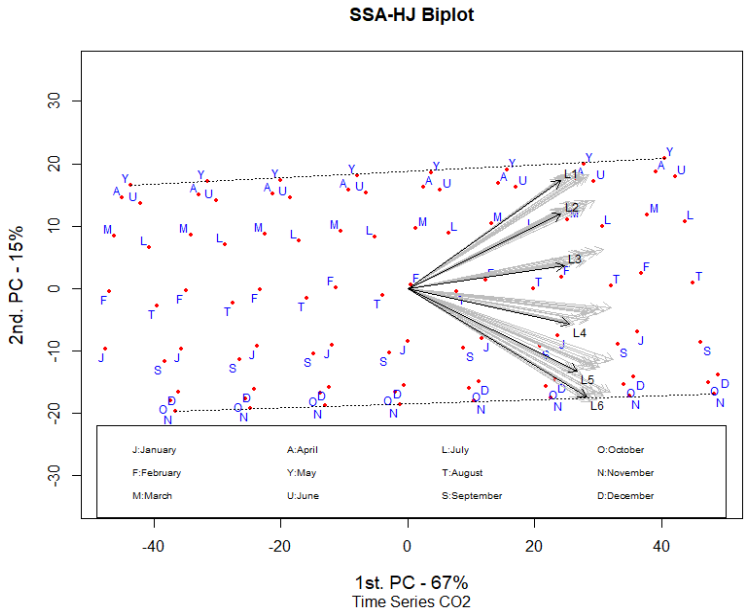


Fig. 3. First SSA-HJ-biplot of the T.S.CO2 trajectory matrix decomposition.

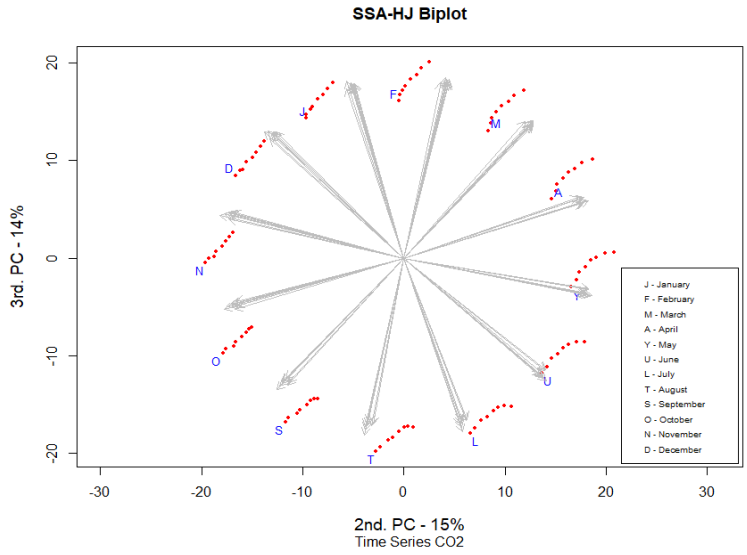


Fig. 4. The second SSA-HJ-biplot whose axes are the 2nd and 3rd PCs.

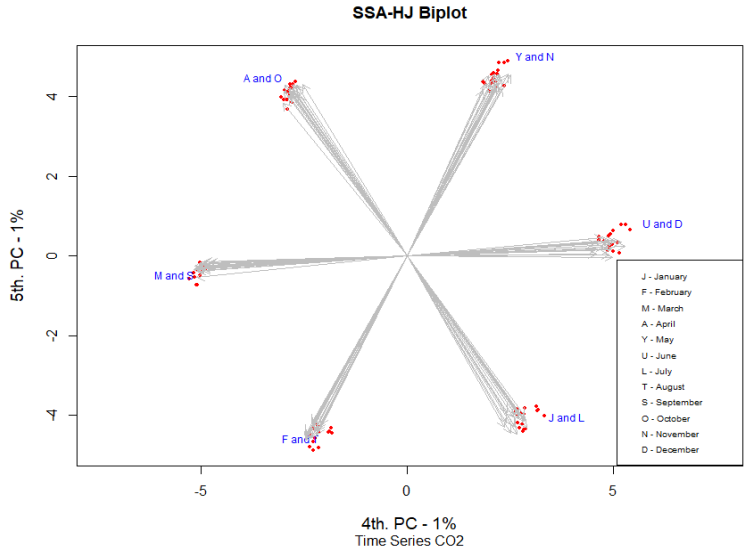


Fig. 5. The third SSA-HJ-biplot whose axes are the 4th and 5th PCs.

Therefore, the result of the grouping step for the decomposition of the T.S. CO2 should be $\mathbf{X}_1$ and $\mathbf{X}_2$, the first corresponding to the trend component, and the second describing the seasonal component, in which

$$\mathbf{X}_1 = \sqrt{\mathbf{t}_1'\mathbf{t}_1}\,\mathbf{t}_1^*\mathbf{p}_1', \tag{28}$$

and

$$\mathbf{X}_2 = \sum_{i=2}^{5}\sqrt{\mathbf{t}_i'\mathbf{t}_i}\,\mathbf{t}_i^*\mathbf{p}_i', \tag{29}$$

with the rest being related to the noise component.

5 Conclusions

This paper attempts to provide an alternative way to visualize and understand the underlying structure of the trajectory matrix, that is the result of the embedding step of the SSA. The HJ biplot visualization method appears to be a promisor exploratory technique adequate to the purposes of this work since it provides interpretability to the results of the SVD step as was illustrated by an application. The SSA-HJ-biplots and auxiliary graphics provided a visual solution for the decomposition of the analyzed time series, properly separating the trend and the oscillatory component, using biplot axes up to the fifth PC. Also, allowed the identification of all relevant eigentriple, composed by the singular values $\sqrt{\mathbf{t}_i'\mathbf{t}_i}$, by the left singular vectors $\mathbf{t}_i^*$, and by the right singular vectors $\mathbf{p}_i$, $i = 1, \ldots, 5$, to perform the grouping step. The study also revealed that the SSA-HJ-biplot points, representative of the row markers ($\mathbf{a}_i'$) and symbol of

the K-lagged vectors that begin in a given period of the series (months in this specific case) could also depict the period itself in terms of dissimilarities, being possible to visually verify the months with the highest and lowest levels of CO_2 concentration in the atmosphere throughout the years. The SSA-HJ-biplot built with the 1st and 2nd PCs proved yet to be useful in dealing with autocorrelations between the column markers, which are drawn as arrows and represent the L-lagged vectors. This study is promising in the sense that the SSA-HJ-biplot has a great potential as an exploratory tool to analyze the structure of a univariate time series due to its visual appeal in such a complex issue.

Acknowledgments.
The authors were supported by Fundação para a Ciência e a Tecnologia (FCT), within project UID/MAT/04106/2019 (CIDMA).

References

1. Alexandrov, T.: A method of trend extraction using Singular Spectrum Analysis. REVSTAT, Statistical Journal, **7**(1), 1-22 (2009)
2. Gabriel, K.: The biplot graphic display of matrices with application to principal component analysis. Biometrika, **58**(3), 453-467 (1971)
3. Galindo, M.P.: An alternative of simultaneous representation: HJ-biplot. Questiió, **10**(1), 13-23 (1986)
4. Geladi, P., Kowalsky, B.R.: Partial Least Squares regression: a tutorial. Analytica Chimica Acta, **185**, 1–17 (1986)
5. Golyandina, N., Nekrutkin, V., Zhigljavsky, A.: Analysis of Time Series Structure: SSA and Related Techniques. 1st ed. Chapman & Hall/CRC, Boca Raton, Florida (2001)
6. Golyandina, N., Shlemov, A.: Variations of Singular Spectrum Analysis for separability improvement: non-orthogonal decompositions of time series. Statistics and its Interface, **8**(3), 277–294 (2015)
7. Greenacre, M.: Biplots in Practice. FBBVA, Bilbao, Biscay (2010)
8. Nieto, A.B., Galindo, M.P., Leiva, V., Galindo, P.V.: A methodology for biplots based on bootstrapping with R. Colombian Journal of Statistics, **37**(2), 367–397 (2014)
9. NOAA Homepage, https://www.esrl.noaa.gov/gmd/ccgg/trends/, last accessed 2019/05/24
10. Vinzi, V.E., Russolillo, G.: Partial Least Squares algorithms and methods. WIREs Comput Stat, **5**, 1–19 (2013)
11. Wold, H.: Estimation of principal components and related models by iterative least squares. In: Krishnaiah, P.R. (ed.), Multivariate Analysis, New York: Academic Press, 391–420 (1966)
12. Wold, S., Albano, C., Dunn, W.J. III, Esbensen, K., Hellberg, S., Johansson, E., Sjostrom, M.: Pattern recognition: finding and using regularities in multivariate data. In: Martens, H., Russwurm, H. (eds.), Food Research and Data Analysis, London: Applied Science Publishers, 147–189 (1983)

Linear regression model for prediction of multi-dimensional time-point forecasting data

Shrikant Pawar[1, 2]* & Aditya Stanam[3*]

[1]Department of Computer Science, Georgia State University, 34 Peachtree Street, 30303, Atlanta, GA, USA.
[2]Department of Biology, Georgia State University, 34 Peachtree Street, 30303, Atlanta, GA, USA.
[3]College of Public Health, The University of Iowa, UI Research Park, #219 IREH, 52242-5000, Iowa City, Iowa, USA.

*Contributed equally
Correspondence: spawar2@gsu.edu

Abstract:

Machine Learning (ML) has been promising in predicting financial series, the direction of the medicine, stock market, macroeconomic variables, accounting balance sheet information and many other applications. Regression models a target prediction value based on independent variables. Different regression models differ based on the kind of relationship between dependent and independent variables they are considering and the number of independent variables being used. The time series signature is a collection of useful features that describe the time series index of a time-based data set. It contains a wealth of features that can be used to forecast time series that contain patterns. The user can learn methods to implement machine learning to predict future outcomes in a time-based data set. Testing the data with linear regression provided higher R-Squared, Adj R-Squared and F-Statistic values. The t-statistic was greater than 1.96 with a p-value less than 0.05. MAPE (Mean absolute percentage error) and MSE (Mean squared error) were reasonably lower. The paper concludes that linear regression can be an effective ML for forecast data provided dimension reductions have been taken into considerations.

Keywords: Regression, Forecasting, Machine Learning

Introduction:

Machine Learning methods (ML) have been widely used in the field of forecasting. The advantage of using ML are its non-linear algorithms with minimization function [1]. ML has been promising in predicting financial series, the direction of the medicine, stock market, macroeconomic variables, accounting balance sheet information and many other applications [2-5]. Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables [6-17]. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on the kind of relationship between dependent and independent variables they are considering and the number of independent variables being used [18]. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). Regression technique finds out a linear relationship between x (input) and y (output). The regression line is the best fit line for the model. This mathematical equation of linear regression can be generalized as follows: $Y = \beta 1 + \beta 2X + \epsilon$; where, $\beta 1$ is the intercept and $\beta 2$ is the slope [19]. Collectively, they are called regression coefficients. ϵ is the error term, the part of

Y the regression model is unable to explain. By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. Cost function (J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y) can be minimized using Gradient Descent. The Gradient Descent function can be derived as follows [20-22]:

$$minimize \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2 \quad J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

The time series signature is a collection of useful features that describe the time series index of a time-based data set. It contains a wealth of features that can be used to forecast time series that contain patterns. The user can learn methods to implement machine learning to predict future outcomes in a time-based data set. The objective of this article is to test linear regression model for prediction of multi-dimensional time-point forecasting data.

Materials and Methods:

i. Data Collection: Forecasting data was collected from the repository dedicated to the M4 forecasting competition, organized by Spyros Makridakis (https://www.mcompetitions.unic.ac.cy/). The data is categorized as Train and Test set of the competition with additional information per series including their ID (M4id), domain (category), frequency (Frequency), number of forecasts requested (Horizon), seasonal periods (SP) and starting date (StartingDate). All the analysis code repositories and raw datasets are deposited on authors GitHub account which can be found at: https://github.com/spawar2/Forcasting-pipelines

ii. Regression Analysis: Regression was performed using functions ‘lm()’ and ‘predict()’ using packages 'DMwR', ‘tidyquant’, ‘timetk’, and ‘broom’. Parameters like R-Squared, Adj R-Squared, F-Statistic, Std. Error, t-statistic, Mean absolute percentage error (MAPE), Mean squared error (MSE), and Min_Max Accuracy values were used for testing the performance of test data prediction. A simple correlation between the actuals and predicted values can be used as a form of accuracy measure. MAPE, MSE values were calculated with library 'DMwR' with function regr.eval(). A higher correlation accuracy implies that the actuals and predicted values have similar directional movement, i.e. when the actuals values increase the predicted values also increase and vice-versa. The MinMax Accuracy can be calculated with following formula:

$$MinMaxAccuracy = mean\left(\frac{min\left(actuals, predicteds\right)}{max\left(actuals, predicteds\right)}\right)$$

Results and Discussion:

Testing the data with linear regression provided higher R-Squared, Adj R-Squared and F-Statistic values. The t-statistic was greater than 1.96 with a p-value less than 0.05. MAPE (Mean absolute percentage error) and MSE (Mean squared error) were reasonably lower while the Min_Max Accuracy ranged from 58-62 % for a 3, 5 and 10 fold validations (Table 1). Figure 1 compares residuals vs fitted, residuals vs leverage, theoretical quantiles and fitted values for test data. Although, we did found low accurate Min_Max Accuracy, the prominent factor of which would be multi-dimensionality of the data. Some techniques like principal component analysis would help in reducing the dimensions, and improve the residuals. The paper concludes that linear regression can be an effective ML for forecast data provided considerations on dimension reductions.

Acknowledgments

No external funding was utilized for the analysis of this paper.

Author contributions

AS and SP contributed to the conception and design as well as the drafting of the manuscript. All authors read and approved the final paper.

Disclosure

The author reports no conflicts of interest in this work.

Supplementary files

1. Daily-test.csv and Daily-train.csv: Testing and Training datasets.

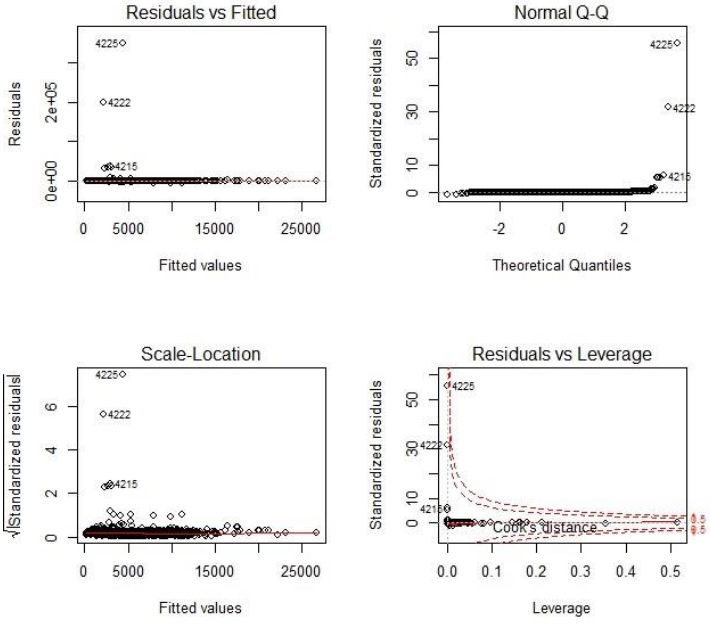
Tables:

Table 1: Residual values for testing datasets.

	3 Fold	5 Fold	10 Fold
Min	-30615.3		
Median	88.8		
Max	28758.4		
Multiple R-squared	0.9368		
Adjusted R-squared	0.9367		
F-statistic	4.181e+04		
p-value	< 2.2e-16		
mae	3.337513e+03		
mse	3.150222e+07		
rmse	5.612684e+03		
mape	1.861367e+00		
Min_max_accuracy	62.27354	58.33132	58.15781

Figures

Figure 1: compares residuals vs fitted, residuals vs leverage, theoretical quantiles and fitted values for test data.



References

1. Hamid SA, Habib A. Financial Forecasting with Neural Networks. Academy of Accounting and Financial Studies Journal. 2014;18(4):37–55.
2. Qiu M, Song Y. Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model. PLOS ONE. 2016;11(5):1–11. 10.1371/journal.pone.0155133
3. Kock AB, Teräsvirta T. Forecasting Macroeconomic Variables Using Neural Network Models and Three Automated Model Selection Techniques. Econometric Reviews. 2016;35(8–10):1753–1779. 10.1080/07474938.2015.1035163
2. Pawar, S, et al. Statistical analysis of microarray gene expression data from a mouse model of toxoplasmosis. BMC Bioinform. 12(Suppl. 7), SA19 (2011)
3. Pawar S, Donthamsetty S, Pannu V, Rida P, Ogden A, Bowen N, Osan R, Cantuaria G, Aneja R (2014) KIFCI, a novel putative prognostic biomarker for ovarian adenocarcinomas: delineating protein interaction networks and signaling circuitries. J Ovarian Res 7:53
4. Ashraf M, et al. (2018) A side-effect free method for identifying cancer drug targets. Sci Rep.
5. Gabor MR, Dorgo LA. Neural Networks Versus Box-Jenkins Method for Turnover Forecasting: a Case Study on the Romanian Organisation. Transformations in Business and Economics. 2017;16(1):187–211.
6. Marr B. The Top 10 AI And Machine Learning Use Cases Everyone Should Know About, Forbes; 2016.
7. Pawar, S, et al. Computational identification of indispensable virulence proteins of Salmonella typhi CT18. Curr. Top. Salmonella Salmonellosis (2017)
8. Pawar S, et al. (2011) Statistical analysis of microarray gene expression data from a mouse model of toxoplasmosis. BMC Bioinform 12(Suppl 7):A19
9. Zhang GP, Qi M. Neural network forecasting for seasonal and trend time series. European Journal of Operational Research. 2005;160(2):501–514. 10.1016/j.ejor.2003.08.037
10. Alpaydin E. Machine Learning: Introduction to Machine Learning. The MIT Press; 2004.
11. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: Data mining, Inference, and Prediction, Second Edition Springer; New York; 2009.

12. Hyndman RJ, Koehler AB. Another look at measures of forecast accuracy. International Journal of Forecasting. 2006;22(4):679–688. 10.1016/j.ijforecast.2006.03.001
13. Goodwin P, Lawton R. On the asymmetry of the symmetric MAPE. International Journal of Forecasting. 1999;15(4):405–408. 10.1016/S0169-2070(99)00007-2.
14. Lahiri, C, et al. Interactome analyses of Salmonella pathogenicity islands reveal SicA indispensable for virulence. J. Theor. Biol. 363, 188–197 (2014)
15. Lahiri, C, et al. Identifying indispensable proteins of the type III secretion systems of Salmonella enterica serovar Typhimurium strain LT2. BMC Bioinform. 13(Suppl. 12), SA10 (2012)
16. Pawar, S, et al. In silico identification of the indispensable quorum sensing proteins of multidrug resistant Proteus mirabilis. Front. Cell. Infect. Micro-Biol. 8, 269 (2018)
17. Venables WN, Ripley BD. Modern Applied Statistics with S 4th ed New York: Springer; 2002.
18. Breiman L. Classification and Regression Trees. Boca Raton, FL: Chapman & Hall; 1993.
19. Therneau T, Atkinson B, Ripley B. rpart: Recursive Partitioning and Regression Trees; 2015.
20. Schölkopf B, Smola AJ. Learning with kernel: Support Vector Machines, Regularization, Optimization and Beyond. The MIT Press; 2001.
21. Meyer D, Dimitriadou E, Hornik K, Weingessel A, Leisch F. e1071: Misc Functions of the Department of Statistics, Probability Theory Group, TU Wien; 2017.
22. Rasmussen CE, Williams C. Gaussian Processes for Machine Learning. The MIT Press; 2006.

Occupancy Forecasting using two ARIMA Strategies

Tiên Dung CAO, Laurent DELAHOCHE, Bruno MARHIC, Jean-Baptiste MASSON.

Keywords: Multiple Logistic Regression, Occupancy Forecast, Time Series Analysis, Nonlinear regression model, ARIMA.

Abstract
We present an occupancy forecast method in a smart home context based on the exploitation of environmental measures such as CO2, sound or relative humidity. This article presents our machine learning algorithm and prediction strategy. It is based on two levels of data exploitation. The first level is "supervised learning" to obtain past occupancy from sensor measurements. It is achieved with a multiple logistic regression algorithm. The second level consists in two main steps. During the first step ARIMA learns and trains the model, using the past occupancy data from level 1. During the second step ARIMA predicts the future occupancy. The innovative part of our paper is that we compare two different ARIMA's (de-seasonalised). The first is the "day-sequence-time-series" (a serial ARIMA). The second is the "daily-slice-time-series" (a parallel ARIMA). We conclude by analyzing the performance of our occupancy prediction paradigm.

1 Introduction

The context of our study is energy efficiency. Energy efficiency has been achieved in recent years by working on the insulation of the building envelope. This strategy has achieved optimal levels of energy performance. Additional gains are now to be sought in optimal thermal regulation. The strategy is to permanently adapt the comfort situation to the living situation. To do this, it is necessary to automatically characterise the activity of the occupants in the building. In today's innovative technological design for smart buildings, the key problem we are faced with is understanding the consumer's behaviour. Our occupancy forecast strategy will in future allow for energy savings in a smart building context. The control/command strategy of the heater will be presented in an upcoming paper. In this article, we address the principle of our method of occupancy prediction.

The method of occupancy forecasting exposed in this paper contains one remarkable contribution: we compute two original ARIMA strategies for the forecast of occupancy. The first is a "Day Sequence" Time Series, which is a common process and the second is a "Daily Time Slice" Time Series which is an unusual process. The second ARIMA consists in forecasting the probability of occupancy of just one time slice (30 minutes). Then, with a loop, we reconstitute a full day by assembling all the time-slices results. We present a comparative analysis of our two ARIMA models against several criteria (error, reliability, temporal consistency, etc.). Finally, we propose conclusions and perspectives for using our prediction algorithms in an intelligent regulation paradigm in the context of energy saving.

2 Related works

Characterisation of human activity and the ability to predict, it is a major issue in many disciplinary fields. Many proposals for methods have already been suggested in the medical field (such as personal assistance), in the energy efficiency field and in many others. In [1], a complete monitoring architecture is presented, including home sensors and cloud-based back-end services. In this article, supervised techniques for behavioural–data analysis are proposed using regression methods and ARIMA. By means of inductive and deductive reasoning, the authors of the article [2] introduce a framework to detect occupant activity and potentially wasted energy consumption. This framework consists of three sub-algorithms for action detection, activity recognition and waste estimation. Unsupervised clustering models are used to detect the occurred actions. In paper [3] a new approach to modeling human behaviour patterns is suggested. The authors use Markov chains to determine an unsupervised model of human behaviour and to detect the deviation over time. Deviating behaviour is revealed through data clustering and analysis of associations between clusters and data vectors representing adjacent time intervals. The activity recognition is also used in [4], which proposes learning customized structural models for common user activities in order to predict the trend of energy consumption. The recognition algorithm is based on recursive structures of user activities obtained from raw sensor readings. Artificial neural networks (ANN) are used in [5] [6] [7] to manage resident activity recognition in Smart Homes. The authors in [5] tackle three ANN algorithms for human activity recognition, namely: Quick Propagation (QP), Levenberg Marquardt (LM) and Batch Back Propagation (BBP). In the same way, an unsupervised learning strategy is used in [8] to improve activity recognition in smart environments. In [9] and [10] the Support Vector Machines (SVM) are used to address the same problem.

3 Theoretical framework

3.1 The Multiple Variables Logistic Regression (MLR):

Logistic Regression is a statistical learning algorithm developed by David Cox in 1958. Its purpose is to reconstruct a qualitative variable Y as a function of one (simple regression) or several (multiple regression) explanatory variables $X_1, \dots, X_k$. A discussion on logistic regression (and variants) can be found in detail in the book by Hastie et al. [11]. The main idea is to express certain log-odds as linear functions of the X_i, using equations similar to classical linear regression.

- When Y is binary, it suffices to define$p(x) = Pr(Y = 1 \vee X = x)$ and to assume that its log-odds is a linear function of the explanatory variables:

$$log\,\frac{p(x)}{1-p(x)} = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k \qquad \text{(Eq. 1)}$$

where the coefficients$\beta_0, \beta_1, \dots, \beta_k$are parameters to be estimated.

- The MLR finds estimates for the parameters $\beta_0, \beta_1, \ldots, \beta_k$ by maximizing the log-likelihood function L(β)with the Newton–Raphson iterative method (the solution has no closed form): at each step, the estimates are updated by

$$\beta_{current} = \beta_{previous} - \left(\frac{\partial^2 L}{\partial \beta^2}\right)^{-1} \frac{\partial L}{\partial \beta} \quad \text{(Eq. 2)}$$

- Once we have estimated $\widehat{\beta_0}, \ldots, \widehat{\beta_k}$, we obtain an estimated probability function:

$$\hat{p}(x) = \frac{1}{1+e^{-(\hat{\beta}_0+\hat{\beta}_1 x_1+\cdots+\hat{\beta}_k x_k)}} \quad \text{(Eq. 3)}$$

- This $\hat{p}(x)$ is calculated at each instant of measurement, giving the posterior probability of occupation as a function of time, as mentioned at the end of § 3.1. Its interpretation is as follows: when it is close to 1, the measurements indicate that the occupant is present; when it is close to 0, the measurements indicate that the occupant is absent; when it is close to 0.5, the measurements could be associated with either presence or absence.

3.2 Pre-processing with STL

The STL method decomposes a time series into the sum of three components: seasonal, trend, and residual (or remainder) using Loess (non-linear regression technique) [15]. An STL decomposition of our data is shown in Figure 1 below. Here, the seasons correspond to days. We call *de-seasonalised data* the residual component. It will be handed to several ARIMA strategies (§ 4.1). Finally, we will add the trend and the seasonal components back to the ARIMA results to obtain occupancy probability forecasts: we will call this operation *re-seasonalising* the data.

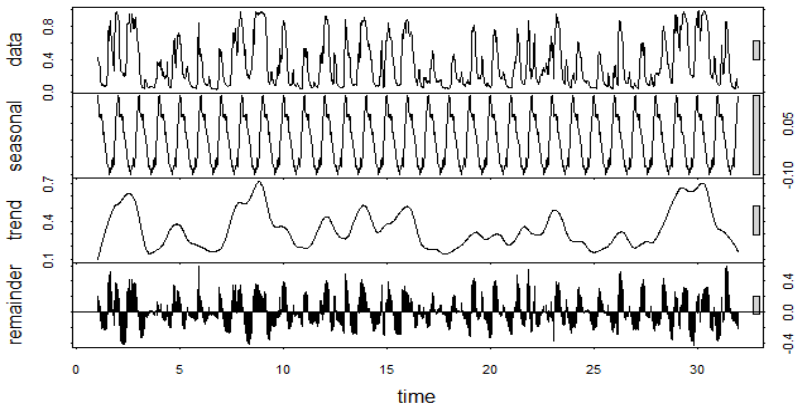


FIGURE 1: STL DECOMPOSITION

3.3 ARIMA

An ARIMA (Auto Regressive Integrated Moving Average) [12] model is a statistical model for analyzing and forecasting time series data. Adopting an ARIMA model for a time series assumes that the underlying process that generated the observations is an ARIMA process; *i.e.* Stationarity [13]. The data will follow the same general trends and patterns as in the past [14]. This may seem obvious but helps to motivate the need to confirm the assumptions of the model in the raw observations and in the residual errors of forecasts from the model.

$$y_t \rightarrow \underset{(long\ term)}{AR\ filter} \rightarrow \underset{(stochastic\ trend)}{Integration\ filter} \rightarrow \underset{(short\ term)}{MA\ filter} \rightarrow \underset{(white\ noise\ error)}{\varepsilon_t}$$

FIGURE 2: ARIMA MODEL

4 Proposed Method

We propose an occupancy prediction method in a smart home context based on the exploitation of the measurements of the sensors disseminated in the building. Our paradigm is based on two consecutive steps integrating the learning process:

- to determine occupancy probability (MLR) based on sensor data
- to forecast occupancy in the near future (STL-ARIMA)

4.1 Forecasting step using ARIMA class model by de-seasonalising data with 2 strategies

Strategy 1 (Serial): "Day Sequence" Time Series Processes Model

This model handles the time series in a classical way: the probabilities of occupancy form a single sequence treated by ARIMA. Here, we assume that whole days will follow the same general trends and patterns as in the past. However, we separate the weekdays from the weekends, and work independently on the two resulting samples (in one, Fridays are followed by Mondays, and in the other one, Sundays are followed by Saturdays). We implement our new "weekday" database and the two types of seasonal variables in the Day Sequence Time Series process (database=[2016,1] (weekday and 30 minute step phases)) and we forecast one day ahead (48 steps of 30 minutes each) with the STL-ARIMA function. The STL ARIMA can be written as:

$$F_{t_d+j}^{days} = L_{t_d}^{d} + jT_{t_d}^{d} + S_{t_d+j-M_d}^{d} + \varepsilon_d \qquad \text{(Eq. 4)}$$

$M_d = 48$ (days sub-seasons slice per 30 minutes).	L: Level
ε: Errors	T: Trend
j: numbers step-ahead forecasts	S: Seasonal
t_d: our benchmark time	

Strategy 2 (Parallel): "Daily Time Slice" Time Series Processes Model.

This model handles the time series in an innovative way: we define 48 time slices per day (each 30 minutes long) and then form a sample for each time slice. We designed this model to take advantage of the regularity per time slices on multiple days (the

occupant's "habits"). Hence, 48 instances of ARIMA are performed on shorter sequences than in Strategy 1. For instance, one ARIMA handles only the probabilities of presence for the time-slice 8:00 to 8:30 am, each data point coming from a different day. Therefore, we use the same database as in Strategy 1, converted into a probability matrix [42x48] that corresponds to 42 days and 48 slices of time per day. This strategy can be seen as 48 "parallel" ARIMAs, whereas Strategy 1 consists of 48 "serial" ARIMAs. We also forecast one day ahead with the STL-ARIMA function, but here it just corresponds to one step in time (one day) for each of the 48 slices (30 minutes). This can be written as:

$$F_{t_{w_i}+1}^{weeks} = L_{t_{w_i}}^{w} + T_{t_{w_i}}^{w} + S_{t_{w_i}+1-M_w}^{w} + \varepsilon_{w_i} \quad \text{(Eq. 5)}$$

$M_w = 5$ (weekdays season)	L: Level
ε: Errors	T: Trend
i: numbers slice of time	S: Seasonal
t_{w_i}: our benchmark time.	

4.2 Implementation algorithm

To determine occupancy (the variable of interest), we use data from Netatmo(c) and infrared sensors disseminated in the environment: we have relative humidity (Hr%), CO2 (ppm), and infrared measurements (PIR 0/1) to determine whether or not the occupant is present. We obtain the probability of occupancy by supervised learning, fitting PIR as a function of the others with multiple logistic regression (MLR). Then, we aim to compare and/or combine two forecast algorithms based on ARIMA models, differing by the strategy for reassembling the time samples: the "Day Sequence" time series" (48 serial ARIMAs) and the "Daily Time Slice" time series" (48 parallel ARIMAs).

The prediction data are reorganised (split) in order to set data to both ARIMA (§4.1). At the "end" of the process the parallel ARIMA forecast are merge together in order to obtain an entire day. The serial ARIMA provides a day forecast directly. Figure 3 illustrates the sequence of computations involve in this method.

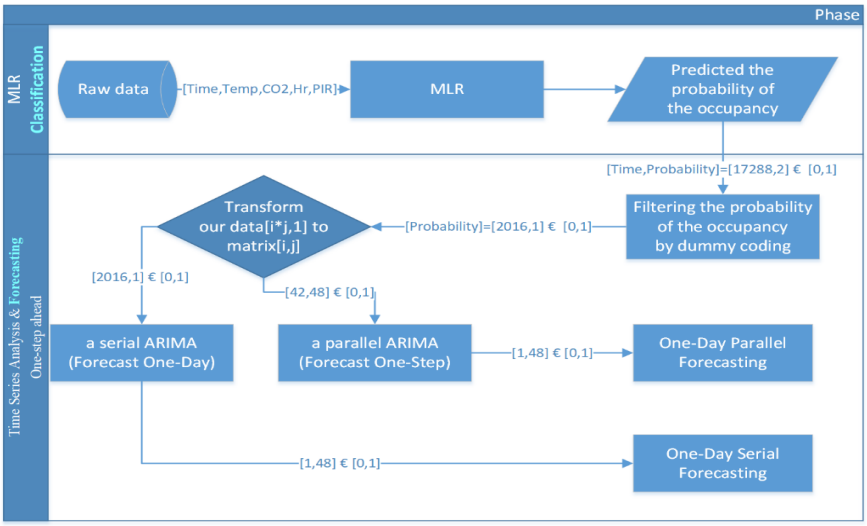


FIGURE 3: PROPOSED ALGORITHM

5 Results and Discussion

5.1 The raw data in the Learning phase

Our perception system is composed of four sources. For each source the sampling rate of the raw data is 5 minutes. The input of the data is almost synchronous. The sensors' data include room temperature (°C), CO2 levels (ppm), relative hygrometry (% Hr) and passive infrared (PIR, 0/1), as shown in the Figure 4. This dataset covers the period stretching from 1 January to 28 February 2017. We esteem that this time range is sufficiently long to evaluate the occupancy behaviour.

date	Temperature	Humidity	CO2	PIR
01/01/2017 12:01	22,3	27,6	1092	1
01/01/2017 12:06	22,3	27,5	1070	1
01/01/2017 12:11	22,3	27,5	1068	1
01/01/2017 12:16	22,4	27,3	1058	1
01/01/2017 12:21	22,4	27,3	1065	0
01/01/2017 12:26	22,4	27,0	1038	1
01/01/2017 12:31	22,5	26,9	1037	1
01/01/2017 12:36	22,6	26,7	1033	1
01/01/2017 12:42	22,7	26,6	1021	1
01/01/2017 12:47	22,8	26,5	1027	1
01/01/2017 12:52	22,9	26,4	1016	1
01/01/2017 12:57	23,0	26,2	1009	1
01/01/2017 13:02	23,1	26,0	993	1
01/01/2017 13:07	23,2	26,0	999	1
01/01/2017 13:12	23,3	25,9	997	0
01/01/2017 13:17	23,3	25,9	990	0
01/01/2017 13:22	23,2	25,7	961	0
01/01/2017 13:27	23,2	25,6	942	0
01/01/2017 13:32	23,2	25,5	913	0
01/01/2017 13:37	23,2	25,7	938	0
01/01/2017 13:42	23,2	25,5	894	1
01/01/2017 13:47	23,2	25,5	866	1

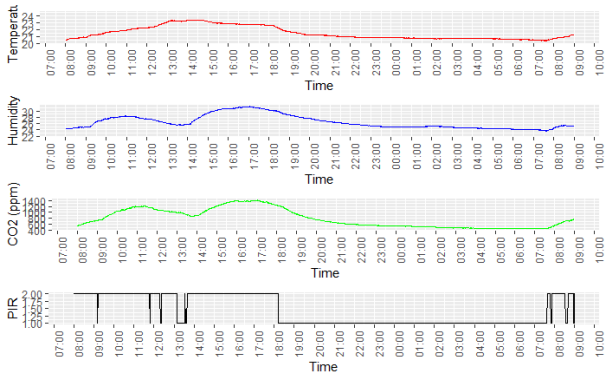


FIGURE 4: THE RAW DATA FOR 1 DAY

The raw dataset is used to train and test a classification in order to determine occupancy probability. The PIR data is only used for the MLR (Multiple Variable

Logistic Regression) classification, to supervise (training) and to control the estimation (testing). The purpose of this classification is to replace all raw data by a new dataset that represents the occupancy probability as a function of time.

5.2 The time series data (Occupancy probability)

In the Figure 5 the reader will find our times series data that was rendered by the Learning phase (MLR).

DateTime	Probab Occup
01/01/2017 00:01	0,472024
01/01/2017 00:06	0,424827
01/01/2017 00:11	0,409648
01/01/2017 00:16	0,427371
01/01/2017 00:21	0,440148
01/01/2017 00:26	0,407419
01/01/2017 00:31	0,43276
01/01/2017 00:36	0,407419
01/01/2017 00:41	0,427659
01/01/2017 00:46	0,409933
01/01/2017 00:51	0,392438
01/01/2017 00:56	0,379828
01/01/2017 01:01	0,374939
01/01/2017 01:06	0,336805
01/01/2017 01:11	0,334485
01/01/2017 01:16	0,365237
01/01/2017 01:21	0,343813
01/01/2017 01:26	0,362828
01/01/2017 01:31	0,379828
01/01/2017 01:36	0,375215
01/01/2017 01:41	0,353532
01/01/2017 01:46	0,387488
01/01/2017 01:51	0,37035
01/01/2017 01:56	0,379828

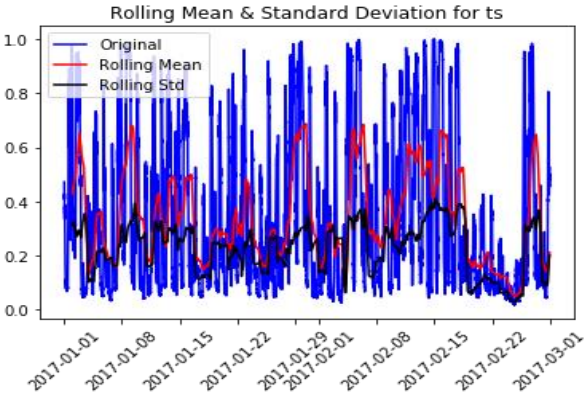


FIGURE 5: THE TIME SERIES DATA – MLR RESULTS (OCCUPANCY PROBABILITY)

Because we are using past data to predict future data, we should assume that the data will follow the same general trends and patterns as in the past. This general statement holds for most training data and modelling. The rolling mean and standard deviation look like they change over time. There may be some de-trending and removing seasonality involved. Applying log transformation, and first-order differencing makes the data more stationary over time. This makes the data suitable to be used in our ARIMA models.

5.3 Forecasting Probabilities

In Figure 6 below, the dataset covers the 01/01/17-28/02/17 period, and we forecast the next day's hourly results (the 48 steps period) with the 2 strategies described in § 4.1 (grey and orange curves). To assess the accuracy of the forecast, we use as reference the output of the MLR classification of a known day (01/03/2017, blue curve). All values are *re-seasonalized*.

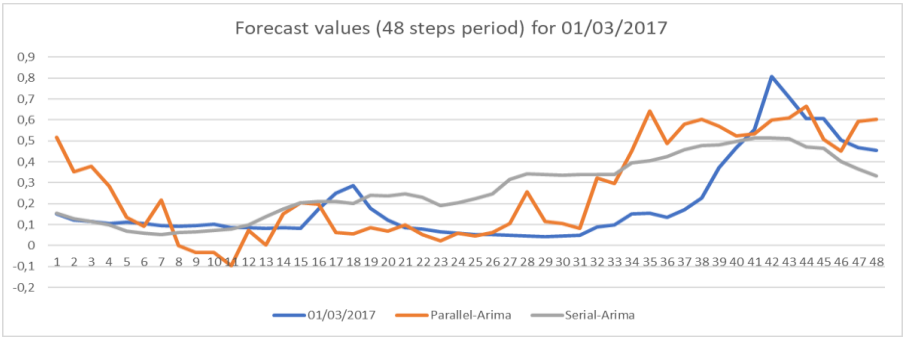


FIGURE 6: OCCUPANCY FORECASTING PROBABILITIES – BOTH ARIMAS – 1 DAY

The STL-ARIMA forecasting with the "Day Sequence" process (Serial) tends to overestimate the occupancy probability and is smoother. The STL-ARIMA forecasting with the "Daily Time Slice" process (Parallel) is jagged: sometimes it overestimates, sometimes it underestimates. Both manage to anticipate the rise of the occupancy probability, but a little too soon (34 units instead of 39 time units, so about 2h30 in real time).

The difference in smoothness is not very surprising, since the Serial strategy corresponds to a single autoregressive model whereas the Parallel strategy corresponds to 48 intertwined models: in the Serial, the auto-regression equation uses actually successive data (previous half hours); in the Parallel, each forecast point is obtained as a function of more distant points in time (previous days).

In Figure 7 below, we report only the Serial strategy forecast (previous grey curve), with the associated confidence interval. In Figure 8 below, we report only the Parallel strategy forecast (previous orange curve), with the associated confidence interval.

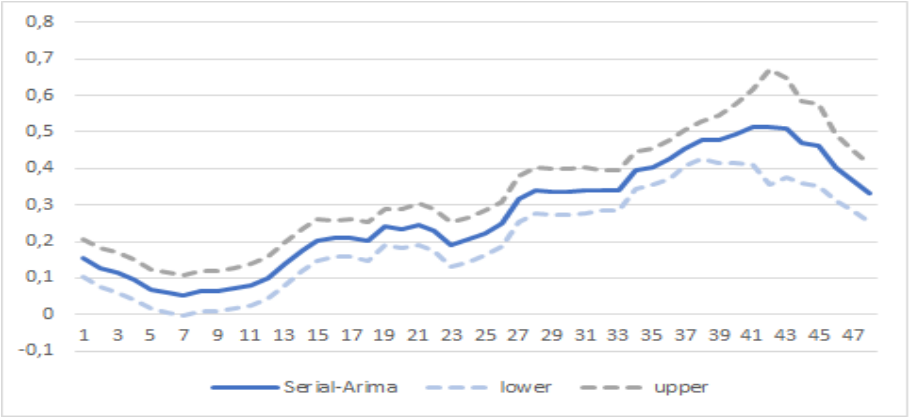


FIGURE 7: STL-ARIMA (SERIAL) FORECASTING WITH 95% CONFIDENCE INTERVAL

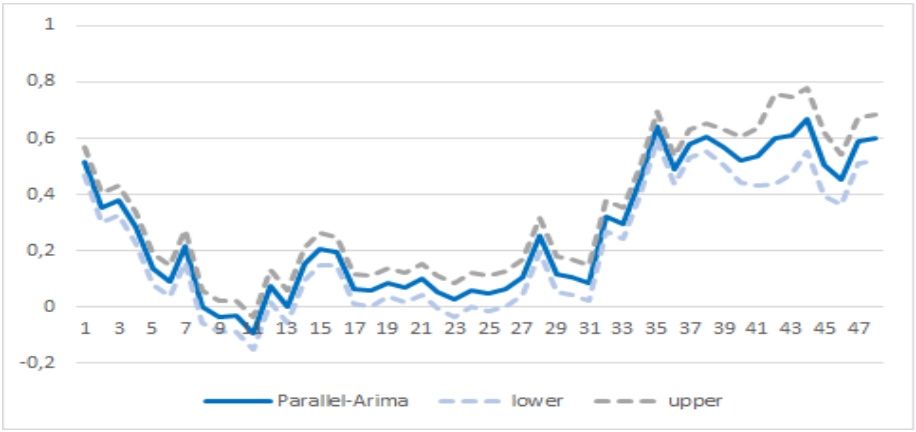


FIGURE 8: STL-ARIMA (PARALLEL) FORECASTING WITH 95% CONFIDENCE INTERVAL

Globally, all confidence intervals are quite narrow, which means that our horizon of forecast (1 day) is suitable. Perhaps more surprisingly, the sizes of these intervals are very similar with both strategies. The Serial strategy learns with more data (2 long subsamples) but its forecast horizon is farther (48 time units); the Parallel strategy learns with fewer data (48 short subsamples) but its forecast horizon in nearer (1 time unit). It seems that the influence of these two factors (sample size, time horizon) counterbalance each other.

In Figure 9 below, we report several statistical indicators that aim to assess the performance of our two strategies on the test day (01/03/2017). We also report the performance of a more naïve ARIMA without the preliminary STL step (Auto-ARIMA).

01/03/2017	Auto ARIMA	Parallel-Arima	Serial-Arima
me	-0,087805	-0,063501	-0,070050
mse	0,039518	0,032706	0,027412
rmse	0,198792	0,180848	0,165565
mae	0,176228	0,136221	0,132106
mpe	-1,711384	-0,593876	-1,261640
mape	1,859567	1,063075	1,455615
smape	0,790289	0,749267	0,628413
u1	0,344109	0,285608	0,280470
u2	9,294281	5,500710	8,533391

FIGURE 9: RESULTS OF STATISTICAL ERRORS

Most indicators are similar among the three methods. Some notable exceptions are the MAE (mean absolute error), the MPE (mean percentage error) and the MAPE (mean absolute percentage error). Both our strategies perform better than the naive one according to the MAE, and our Parallel method performs better than the others according to the MPE and the MAPE. It seems that our innovative strategy has real qualities and deserves interest.

6 Conclusion and Perspective

In this article, we have proposed to deal with occupancy forecasting in a Smart Building context. Occupancy forecasting allows a smart control of HVAC devices in order to save energy and optimise comfort. We presented a forecasting strategy of occupancy mainly based on four steps. First, from direct data measurements (CO2, PIR, Hr, Temp), we define an occupancy probability based on MLR classification. Then, we remove the seasonal component of the time series of occupancy through the STL-method. The third step predicts the temporal signal (occupancy) with two ARIMA strategies: one is the "Day Sequence" (Serial) and the other is the "Daily Time Slice" (Parallel). Finally, we add the seasonal component back.

The cautious reader will have noticed that in Figure 7 , the forecasts are negative between 8 and 11 units of time. This cannot represent a suitable probability, and is due to the fact that ARIMA works with unconstrained real values. We plan to solve this problem by using the ARIMA strategies on the log-odds instead of the probabilities. Globally, both ARIMA strategies give suitable results with low uncertainties. The "Daily Time Slice" forecast is more dynamic than the "Day Sequence" one, but has similar uncertainties, at least for a 1-day horizon. It is well known that as the forecast horizon increases, the confidence intervals' size tends to rise. Our two strategies might exhibit a difference in the speed of this size increase. This question will be addressed in future works.

Bibliographie

[1] Mora, Niccolò , Guido Matrella and Paolo Ciampolini, «Cloud-Based Behavioral Monitoring in Smart Homes,» *Sensors (Basel, Switzerland),* Vols. %1 sur %218,6 1951, n° %1doi:10.3390/s18061951, 15 Jun. 2018.

[2] Simin Ahmadi-Karvigha, Ali Ghahramania, Burcin Becerik-Gerberb, Lucio Soi-belmanc, «Real-time activity recognition for energy efficiency in buildings,» *Ap-plied Energy 211,* p. 146–160., 2018.

[3] Jens Lundström, Eric Järpe, Antanas Verikas, «Detecting and exploring deviating behaviour of smart home residents,» *Expert Systems With Applications 55,* p. 429–440, 2016.

[4] Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, Marco Ortolani, «User activity recognition for energy saving in smart homes,» *Pervasive and Mobile Computing 16,* p. 156–170, 2015.

[5] Homay Danaei Mehr, Huseyin Polat and Aydin Cetin, «Resident Activity Recognition in Smart Homes by Using Artificial Neural Networks,» *2016 4th International Istanbul Smart Grid Congress and Fair (ICSG), Istanbul, Turkey,* n° %1DOI: 10.1109/SGCF.2016.7492428, 20-21 April 2016.

[6] Niall Twomey, Tom Diethe, Ian Craddock, Peter Flach, «Unsupervised learning of sensor topologies for improving activity recognition in smart environments,» *Neurocomputing,* vol. Volume 234, n° %1ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2016.12.049, pp. 93-106, 2017.

[7] Anthony Fleury, Michel Vacher, Norbert Noury, «SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results,» *IEEE Transactions on Information Technology in Biomedicine*, Vols. %1 sur %2Volume: 14, Issue: 2, March 2010.

[8] Muhammad Fahim, Iram Fatima, Sungyoung Lee, Young-Koo Lee, «Activity Recognition Based on SVM Kernel Fusion in Smart Home,» *Part of the Lecture Notes in Electrical Engineering book series (LNEE, volume 203), Computer Science and its Applications*, vol. 203, pp. 283-290, Octobre 2012.

[9] Vanus, J., Belesova, J., Martinek, R. et al., «Monitoring of the daily living activities in smart home care,» *Human-centric Computing and Information Sciences (2017)*, vol. 7, n° %1https://doi.org/10.1186/s13673-017-0113-6, 30 December 2017.

[10] Jiho Park, Kiyoung Jang, Sung-Bong Yang, «Deep neural networks for activity recognition with multi-sensor data in a smart home,» *Internet of Things (WF-IoT) 2018 IEEE 4th World Forum on*, pp. 155-160, 2018.

[11] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2e éd., Newyork: Springer, 2009, pp. 119-135.

[12] Brownlee, Jason, «A Gentle Introduction to SARIMA for Time Series Forecasting in Python,» 17 08 2018. [En ligne]. Available: https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/. [Accès le 06 2019].

[13] Nau, Robert, «Stationarity and differencing,» Forecasting home page, 09 2014. [En ligne]. Available: https://people.duke.edu/~rnau/411diff.htm. [Accès le 06 2019].

[14] Srivastava, Tavish, «A Complete Tutorial on Time Series Modeling in R,» Analytics Vidhya, 16 12 2015. [En ligne]. Available: https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/. [Accès le 06 2019].

[15] Robert B. Cleveland, William S. Cleveland, McRae, J. E., & Terpenning, I. J., «STL: A seasonal-trend decomposition procedure based on Loess.,» *Journal of Official Statistics*, vol. 6, n° %11, pp. 3-33, 1990.

Engineering Data for Business Forecasting

Reverse adaptation of Time Series Data

Forecasting and Similarity

and a contribution to

Forecasting Sporadic Time Series

Author

Klaus Spicher

(retired from TH OWL Lemgo, Germany)

Keywords: Forecasting, Outlier, Missing Data, REVINDA, Similarity, Sporadic Time Series

Abstract: **Section 2** describes two innovative ideas/conceptions, enabling the development of new forecasting methods. One related method (algorithm) based on the described ideas is presented in **Section 3**. **Section 4** deals with the development of a new heuristic approach enabling (beyond Croston-/ and WSS-Methods) forecasting the number of sporadic events along the forecast period and a related metric measuring the result.

1. Introduction

The actual development of IT offers the revision of traditional forecasting approaches. Big Data, Predictive Analytics, Machine Learning (ML) and AI are hyping the actual discussion. - Nevertheless the actual paper follows traditional approaches based on new (innovative) ideas for creating supportive results for practitioners based on heuristic approaches.

2. Ideas and Conceptions

2.1 Reverse Adaptation of Time Series Data

The comparison of Forecasts and Actuals along a Time Series results in deviations. Let's assume at time t the forecast error or, e.g. the average forecast error along the last k observations exceeds an acceptable limit. What can be done to re-adjust the forecast procedure aiming at again improving the next forecasts?

The answer in this paper: Modifying the original history values (i.e. creating a modified set of history values, we call it "virtual history"), for which the forecast minimizes the absolute forecast error for the latest (badly predicted) forecast or minimizes the average forecast error along the last k observations. (The target function for optimization can be arbitrarily selected.) - The solution of this problem seems to be impossible as long as we look at most (all?) traditional forecasting methods. So, in order to follow this idea, a new forecasting approach has to be designed allowing for "reverse adaptation of time series data".
The method REVINDA (Reverse Index Data Transcription), allowing for target-oriented modification of historical data will be described in Section 3.1. A big retail-chain (Print Media) is using this method weekly for their online-branch since more than 5 years successfully, forecasting online orders on a daily basis for 3 weeks ahead.

2.2 Forecasting and Similarity

The idea utilizing methods of digital image processing and especially Similarity Metrics in forecasting seems to be straight forward, but to the knowledge of the author has not been applied up to now. So, the idea presented here, is dealing with similarity of time series. Experiments with many different Similarity Metrics in image processing resulted in applying the "overlapping bar charts" (in German language: "Histogramm-Schnitt") for forecasting. The definition of Similarity is given by

$$S(H_1; H_2) = \sum_i \min(H_1(x_i); H_2(x_i)) \quad \text{mit } 0 \leq S \leq 1. \qquad (2.1)$$

$(H_1; H_2)$ Represent 2 discrete density functions. Replacing the density functions H_1 and H_2 by scaled time series for 2 history periods, the similarity resp. distance of the time series can be determined from equation (2.1). Potential applications become relevant for

(1) Identification of **Missing Data** (E.g. identification whether a time series of length N with k gaps has to be treated as sporadic time series or as a complete time series with k Missing Data). The critical value of k can be specified due to a pre-set Similarity Level S.
(2) The same principle holds for **Outliers**. The replacement value for an outlier can be specified to a given (intended) similarity S between the original and the modified time series. There is no comparable method to the knowledge of the author.

(3) Similarity can be used as **holistic measure of forecasting "accuracy"**. In addition, Similarity can be used as an indicator for (unidentified) inherent 'structural conformity' along the two history periods, supporting the selection of forecast methods.
(4) Based on the concept of similarity the **forecasting method** "METRIX" has been developed. In case the history periods $H_{-2}(x_i)$ and $H_{-1}(x_i)$ show similarities

above 60% the forecast quality at least competes with traditional forecast methods. - The details of the issues discussed above will be presented in Section 3.2.

2.3 Forecasting Sporadic Time Series - the SIMFAC-Approach

A new methodology is under development extending the standard solutions of Croston-/ and WSS-method. As an intermediate project-report the findings up to now are offered for presentation. - The actual state of development will be given in Section 4.

3. Results from Ideas and Conceptions

3.1 Reverse History Value Adaptation: "REVINDA" Forecasting Method

The conception of reverse value adaptation can be read from the displays below.

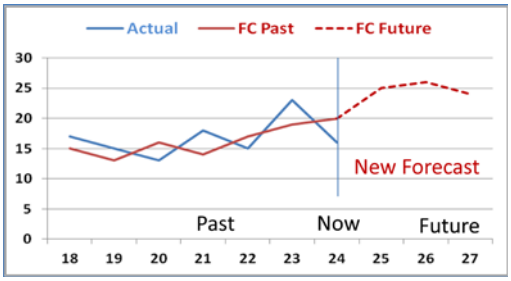


Exhibit 1

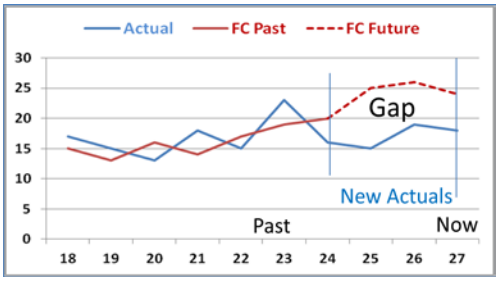


Exhibit 2

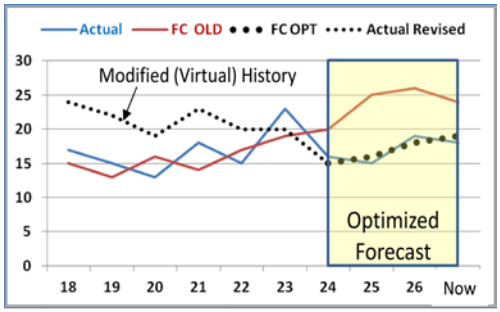


Exhibit 3

The gap between Forecast and Actuals (Exhibit 2) will be reduced by creating a virtual history (Exhibit 3). The virtual history is not calculated explicitly. It is represented by a set of REVINDA parameters, which are optimized due to the selected error reducing target function. The next forecasts will apply the virtual history.

For optimization the ECXEL-SOLVER-Function has been applied. Creating a virtual history can be applied in case of unacceptable forecast error size or in line with each forecast step. – In case of volatile time series the application of gentle smoothing methods is recommended.

REVINDA Method – the Algorithm

REVINDA requires 2 complete history periods H_{-2} and H_{-1}. The forecast period is denoted by H_0. The history values are $y_{-2}(t_i)$ and $y_{-1}(t_i)$ respectively. Based on the selection of reference functions $K_{-2}(t_i)$ and $K_{-1}(t_i)$ the individual history values are transformed to so called "Period-Index-Values" (PI).

$$PI_{-2}(t_i) = \frac{y_{-2}(t_i)}{K_{-2}(t_i)}; \quad PI_{-1}(t_i) = \frac{y_{-1}(t_i)}{K_{-1}(t_i)} \tag{3.1}$$

A second category of Index-Values are calculated – "Sequential Index-Values" (SI) for k (indicating the history periods) from 1 up to 3 periods ahead and both history periods j.

$$SI_j^{(k)}(t_i) = \frac{y_j(t_{i+k})}{y_j(t_i)} \tag{3.2}$$

The forecasts k steps ahead are calculated according to:

$$\hat{F}^{(k)}(t_i) = c^{(k)} * PI_0(t_i) + d^{(k)} * SI^{(k)}(t_i) \tag{3.3}$$

For k = 1 to 3 and $\forall$ i and

$$PI_0(t_i) = a * PI_{-2}(t_i) + b * PI_{-1}(t_i) \tag{3.4}$$

$$SI^{(k)}(t_i) = a^{(k)} * SI_{-2}^{(k)}(t_i) + b^{(k)} * SI_{-1}^{(k)}(t_i) \tag{3.5}$$

Initialization:

a) Selection of Reference-Function $K_{-j}(t_i)$
The forecast quality does not depend on the selection of the reference functions. For simplicity reasons the average of $y_j(t_i)$ represents $K_{-j}(t_i)$ for j = 2, 1. But also e.g. polynomial trends can be used. – For planning reasons $PI_0(t_i)$ can be modified by adding an incremental percentage (x %). In this case $PI_0(t_i)$ $\forall$ i show the individual planning values along H_0 using a = b = 0,5 or other combinations for weighing the history periods. Optimizations will modify the parameters a and b.

b) The initial values for all parameters a, b, c and d (weighing factors) are equal (= 0,5).

As mentioned in Section 2.1 REVINDA is being used in Print Media retail online business since more than 5 years on a weekly basis. There is no booking of sales on Sundays. Comparisons with other forecasting methods (e.g. Holt / Winter) reveal at least competitive quality or better, depending on the consistency of the inherent drivers/structure. High Similarity indicates high consistency. An example of the weekly output is given in Exhibit 4. The dotted red line represents the daily forecast.

Forecast - Werte (Std. Profil)						
9	10	11	12	KW		8.865
3.194	3.068	3.439	3.088	Sa		IST KW12
0	0	0	0	So		4.336
18.227	17.506	19.623	17.620	Mo		0
10.106	9.705	10.879	9.769	Di		19.677
9.406	9.034	10.126	9.093	Mi		11.507
8.718	8.373	9.386	8.428	Do		8.921
8.531	8.193	9.184	8.247	Fr		7.688
11,9%	**8,6%**	**6,6%**	**12,3%**	9,8%		9.923
58.182	55.878	62.638	56.243	FC:IST		62.052

Exhibit 4

3.2 Similarity-based Findings

In this section results from applying Similarity Conceptions on complete and Sporadic Time Series are presented.

3.2.1 Missing Data

Similarity Theory allows for the identification of the critical number k' of gaps in time series, specifying the transition from complete time series with k' missing values and sporadic time series with k > k'. k' results from simulation. Randomly selected values are removed one by one from the time series and the related similarities are calculated accordingly. When the related similarity falls below a pre-set similarity level S, the critical k' is identified. – Interpretation: A time series of length N = 40 will be tagged as a complete time series with missing data in case k' ≤ 2 values are missing.

Limits of Missing Data										
N	10	20	30	40	50	60	70	80	90	100
k'		Similarity Level = 90%								
1	-	1	1	1	1	1	1	1	1	1
2	-	-	-	2	2	2	2	2	2	2
3	-	-	-	-	-	-	3	3	3	3
4	-	-	-	-	-	-	-	4	4	4

Exhibit 5

3.2.2 Outliers

How to identify (not arbitrarily selected) replacement values of outliers of a complete time series of length N represents presumably an unsolved problem. The conception of Similarity offers a simple solution. Removing a potential outlier Ω = Max (y_i) ∀ i, completely from a time series results in two time series. The related similarity can be calculated applying formula (2.1). Under the condition of pre-setting a required similarity S (e.g. S = 90%) the replacement value R can be calculated.

$$R = (S-1) * \sum(y_i) + \Omega. \qquad (3.6)$$

Looking at sporadic time series of length N: In case the k gaps represent "Blanks", i.e. the values do not exist, equation (3.6) can be applied. If the gaps are interpreted as zeros, i.e. taking the accord of zeros into account, then

$$R = (S-1) * \sum_i(y_i) + \Omega - (1-S) * k/N\,. \qquad (3.7)$$

In case another value in the new time series again presumably represents an outlier, the procedure can be repeated. Applying the application of equation (3.7) twice, results in $S = S_1 * S_2$ with S representing the similarity between the original and the two times modified time series.

3.2.3 Forecasting Accuracy

In this paper forecasting is seen as the extrapolation of inherent, structure(s). Level, trend, seasonality and long-term cycles represent visible/identifiable structures offer-

ing different options for estimation and thus developing traditional forecasting methods. - Similarity between Forcast and Actual represents a holistic measure of "accuracy", in some way comparable with the measure of accuracy proposed by probabilistic forecasting. In addition the similarity between H_{-2} and H_{-1} can be seen as a measure for the consistancy of the inherent drivers of the processes behind the time series. For the author another measure for inherent structures is Permutation Entropy (PE). Maybe it's worth analyzing whether similarity and PE are linked. Another area for future research might be comparing Theil's U and related similarities (and PE).

3.2.4 Similarity-based Forecasting Approach.

The similarity concept enables forecasting for the complete forecasting horizon. So, in case looking at monthly data – given the monthly data for 2 years history – a forecast can be made for the complete next year on a monthly basis.

The scaled monthly time series data are supposed representing a frequency distribution. This distribution represents the relative monthly profile of the time series. For the two history data sets the similarity S (according to equation (2.1)) measures the structural conformity in terms of the "distance" = 1-S.

Instead of the Minimum in equation (2.1) we define different "distances S*" by using either the Maximum or the Average.

E.g. $S^*(H_1; H_2) = \sum_i \text{ave}(H_1(x_i); H_2(x_i))$ mit $0 \leq S$. (3.8)

Estimating or fixing the (planned) volume of next year in terms of equal monthly values and applying the monthly profile provide a first monthly forecast for next year. Then, based on the actual monthly values (Jan, Feb, … etc.) the year-end estimation will be adapted accordingly. This simple procedure is called METRIX-Approach.

The resemblance between REVINDA and METRIX is visible. Both approaches utilize the structural in-data conformity – but with different procedures. METRIX was developed for simplification of REVINDA.

The comparison of the numerical accuracy (MAD, MAPE, …) of both methods do not show significant differences.

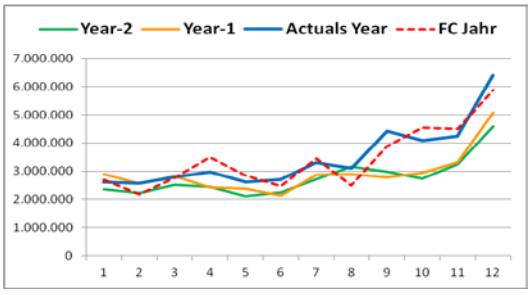


Exhibit 6 shows an example from a a German Print Media Company representing online sales with X-mas seasonality. In spite of the simplicity of the METRIX approach, the results show acceptable quality.

Exhibit 6

4. Forecasting Sporadic Time Series – SIMFAC Approach for Spare Parts

Due to specific attributes (Volatility and Average Gap Size) sporadic time series can be categorized into Erratic, Lumpy, Smooth, and Intermittent time series according to relevant literature. The category of "Isolated Data" is mentioned in addition, because in practice many time series of monthly data with zero, one or just two "events" (i.e. values >0) per year occur.

The SIMFAC approach (like REVINDA and METRIX) is based on (simple) data engineering without any additional assumptions. **The development of SIMFAC has not been finished yet.** But first results show a promising performance extending the results of Croston-Method [1] and WSS-Method [2] because SIMFAC estimates the number of events along the forecast period and year-end volume. Extensive literature studies did not reveal comparable findings. In addition a new compound metric will be introduced for measuring the performance of the new SIMFAC results. The quality of preliminary results encourages the author adding SIMFAC for being discussed at the Conference.

The final version of the SIMFAC Method (forecasting sporadic time series) will consist of the methodologies for each category of sporadic time series – estimating the number of events, the timing of events, the total volume for the forecast period and compound measures (metrics) for assessing the forecast quality.

Again, we start with a history of two years monthly sporadic data. The pool of available data for testing consists of 300 time series (monthly values for 5 years) from a German High Tech Engineering company and 200 time (for 3 years) series from a different Chinese High Tech company.

4.1 The SIMFAC approach

The final design of the method consists of 4 steps.

(1) Data engineering for identifying the number of events along the forecast period, which again is one year on a monthly basis. The basic algorithm will be presented below.
(2) Identifying category-specific modifications for improving forecast quality – e.g. selection of approximation functions including variations of extrapolation, moving slopes for improving the estimate of events, …
(3) Designing metrics for compound measuring forecast quality. Forecast quality consist (1) of the estimated the number of events, (2) the number of timely matches during the forecast period and (3) the total volume for the whole forecast period.

(4) Analysis of the data pools (little Big Data) for identifying additional supportive attributes relevant for forecasting demand of spare parts – including the comparison of the different data pools.

The actual state of research: Steps 1 is ready for application. Significant progress for step 2 has been achieved.

4.2 The SIMFAC Algorithm

Starting with two historical sporadic time series V_1(t) and $V_2(t)$ each of length N (t = 1, …,N). For some t values of V_1(t) = 0 applies, while obtaining for the other t values $V_1\ (t) > 0$. The same holds for $V_2\ (t)$ and $V_{1+2}\ (t)$ with length 2N. Then

$$z_{1,2,1+2}(t) = \begin{cases} 0, \text{ if } V(t)>0 \\ \\ 1, \text{ if } V(t)=0 \end{cases} \quad \text{identify the gaps in history data.} \qquad (4.1)$$

The forecast period is 1 year (N=12 months). N can be selected arbitrarily. In case the individual steps are valid for all 3 data sets the indices are omitted.

The Algorithm:

(1) Determination of the empirical distribution functions $Z(t)$ of counted Zeros for all 3 functions V(t). $Z(t) = \sum_t z(t)$ for t = 1 to N, respectively t = 1 to 2N.
(2) Approximation of Z(t) (e.g. linear, polynomial, sigmoidal, …). The Approximation Function is called Z*(t). Restriction: Z*(t) has to be monotonous.
(3) Extrapolation/extension of Z*(t) along the complete forecasting period.
(4) Determination of the integer parts of Z* (t_i) for all t_i of the forecasting period denoted by $ZI^*\ (t_i)$. E.g. {25, 26, 26, 26, 27, 28, 28, 29, …}.
(5) Transformation of $ZI^*\ (t_i)$ into a binary sequence called $ZI^{**}\ (t_i)$.
If $ZI^*(t-1) = ZI^*(t)$, then $ZI^{**}(t) = 1$, else $ZI^{**}(t) = 0$. The result according to the example under (4) ends in { 0, 1, 1, 0, 0, 1, 0, …}
(6) The number and timing of events result from the intersections of the extrapolated approximation function $Z^*\ (t_i)$ with the horizontal integers of the vertical axis.

Obviously the **slopes** of the approximation functions (e.g. trend lines) determine the number and the timing of estimated events along the forecast period. Applying the algorithm on histories H_{-2} and H_{-1} (each with length N) and the combined History (with length 2N) results in three different forecasts. – The forecasts of the number of events – at this level of method development – still represents a rough estimate, because this approach can be applied for all categories of sporadic time series. The forecast of the timing of the events cannot be regarded as a meaningful "forecast". Matches will happen incidentally.

It seems to be worth mentioning that for erratic and lumpy time series the quality of number of events is relevant, while for smooth and intermittent time series the quality of the related volumes dominates.

The remaining development issues are under research.

5. Summary

The paper summarizes research activities and findings of the author over some years. All results are based just on "Data Engineering" without any theoretical assumptions. Key objective is creating added value to business. All methods presented in this paper just utilize the available data.

Surprisingly, applying Similarity Metrics of digital image processing seems to represent an innovative approach. Especially the results regarding Outliers of sporadic time series will contribute to improving related forecasting. The SIMFAC project will follow up this aspect.

6. Literature

[1] **Croston**, J. : *Forcasting and Stock Control for Intermittent Demands*; Operational Quarterly, 3 (1972); pp 289-303

[2] **Willemain**, T., **Smart**, S., and **Schwarz**. H.: *A new Approach to forecasting intermittent demand for service parts inventory.* International Journal of Forecasting; 20 (2004); pp. 375-387

About the Author: Prof. em. Dr. Klaus Spicher,
58640 Iserlohn, In den Hülsen 7 - spicher@gmx.de
Tel.: +49 2371 154477; +49 171 7979 361
Germany


Klaus Spicher
Education: RWTH Aachen – Mathematics; Dr. rer. nat. [Statistics/Quality]
SPRINGORUM-Medal (Best of Year 1966)
Management History: 7 years Unilever; 3 years Grohe;
7 years SME Owner & CEO Eco-House Construction
10 years Advisory Board Member of a public Internet-Company
Consulting History: Co-Founder of German Logistical Society BVL (1978)
> 20 years SPL Consulting in Germany (Logistics, SCM, ...)
Teaching 10 years - Professor Logistics, SCM (OWL, Lemgo, Germany)
Teaching – Consulting – Coaching in China
2002 – 2014 EU-Expert in CEVTC-Project (China Europe Vocational Training Centre)
Founding Chinese Company run by Chinese Management (Wuhan)
Teaching in different Universities (Wuhan, Shanghai)
International Conference Keynote Speaker (ICLEM, GLSC, OCSF, ...)
Coaching post-graduates for the Chinese Academy of Sciences (IIEE)
Consulting / Coaching Blue Chips – Huawei, China Tobacco, Haier, AVIC ...
Awards „Supply Chain Consulting Medal" (Huawei – 2014)
Appointment „DeTao-Master" International Expert SCM (DeTao – 2015)
„Operational Excellence Award" for 6 years services (Huawei – 2016)

EVALUATING THE EFFECTIVENESS OF TRANSPORTATION INFORMATION PROVISION IN THE SHARING ECONOMY CONTEXT

Joshua Paundra[1,*], Jan van Dalen[1], Laurens Rook[2], Wolfgang Ketter[1,3]

[1]Rotterdam School of Management, Erasmus University, Rotterdam, the Netherlands
{Paundra, jdalen}@rsm.nl
[2]Faculty Technology, Policy and Management, Delft University of Technology, Delft, the Netherlands
{l.rook}@tudelft.nl
[3]Faculty of Management, Economics, and Social Sciences, University of Cologne, Cologne, Germany
{ketter}@wiso.uni-koeln.de

Abstract. Information communication and technology (ICT) based transportation applications have facilitated the provision of transportation information related to environmental impact, trip price, and traffic situations to commuters. ICT advancement is also central in the rise of sharing economy transportation services, such as ridesharing. These developments raise the possibility of encouraging people to consider alternative transportation options other than private vehicles. This study considers the use of information provision as a soft policy intervention in the presence of sharing economy transportation options. Based on an experiment, we showed that providing information on environment, price, and traffic enabled people to respectively consider the more environmentally friendly, cheaper, and lower travel time options. These effects, in part, depend on people's individual psychological ownership. Providing relevant transportation information seemed to be a promising soft policy approach.

Keywords: information provision, nudging, environment, price, traffic, individual psychological ownership, sharing economy

1 Introduction

The United Nations [1] expect urban population to increase by 2.5 billion in the next three decades. This urbanization trend will continue to put pressure on urban mobility. To this end, research has identified private vehicles as the main cause of urban traffic congestion and pollution. To cope with the increasing urban mobility demand, policy makers have considered the use of policy intervention, which aims to make people shift from private vehicles to alternative transportation modes in order to increase the efficiency of the existing system. However, policy interventions may be unwelcomed by commuters if these interventions directly penalize car driving [2,3]. The use of "soft' policy intervention— that aims to change the demand for a particular mode of transport,

adfa, p. 1, 2011.

especially private vehicle, without restricting use of any of the available options— is thus preferred.

A commonly used soft policy strategy is intervention through information provision. In recent years, this strategy becomes relevant given the role of advanced information and communication technology (ICT) in assisting commuters in their daily commute. ICT advancement brings about two changes to the transportation systems. First, commuters can obtain information about the trip characteristics prior to their travel [4]. ICT-based travel applications, such as Google Map, have been used by commuters to arrange and adjust their trips. Thus, these applications play a crucial role in influencing commuting decisions. Second, ICT advancement brings new mobility services based on sharing economy principles, such as the app-based ridesharing service ([5]). Ridesharing services has been linked to societal benefits, including traffic congestion reduction [6] and a decrease in drunk-driving accidents [7]. The availability of these mobility services increases the range of travel options, which benefits commuters as they can choose transportation options that fit with their preferences. We argue that these two developments are central in the discussion of the effectiveness of transportation information provision as soft policy intervention.

The objective of this study is to investigate the effectiveness of different types of information provision in influencing people's decisions to take private car, public transportation, or sharing economy transportation services. To do so, several aspects of information provision as an intervention strategy are considered. We investigate the provision of three different types of information, namely, environmental, price, and traffic information, which has been known to people's preference to take alternative transportation options [8,9]. While transportation information provision has been widely reviewed (e.g. [10,11,12]), research on the effectiveness of various types of information provision in affecting people's transportation mode decisions in the sharing economy context remains sparse. Next, we examine the quality of the provided information, particularly for price and traffic information. Extant studies indicate that low-quality traffic information would impact commuters' decisions [13,14]. We also consider people's individual psychological disposition. Information provision can only effectively be used to nudge people to certain transportation options if people are open to the possibility of using these options. Paundra et al. [15] found evidence on the moderating role of individual psychological ownership in people's evaluation of alternative transportation options in relation to private vehicles.

The remainder of this study is organized as follows. Section 2 describes the design of the experiment. Section 3 presents the results of our analysis. Finally, the study concludes with the discussion of the findings.

2 Method

2.1 Participants

A total of 315 undergraduate students from a Dutch university participated in an online experiment. Among these participants, six did not complete the experiment, and a further 25 participants were excluded, because they had zero standard deviation in

their response to the individual psychological ownership construct, which contains an item with reverse scale. The final data set contained observations about 284 participants (109 women and 175 men; *M*age = 21.36 years, SD = 2.45).

2.2 Design

The experiment was a 2 (environmental information: no, yes) x 2 (price: high range, low range) x 2 (traffic: low, high) between-subjects which was introduced in stepwise manner over the forty rounds of mode decisions, in which four transportation modes were distinguished: private car, public transportation (PT), exclusive ridesharing (RSE), and pooled ridesharing (RSP). Individual psychological ownership was added to this design as a covariate. The forty rounds were divided into four phases that introduced the three manipulations of the experiment in stepwise manner in the experiment. Environmental information for each transportation mode was displayed from round six onwards, price information for each transportation mode choice from round eleven onwards, and traffic information from round 26 onwards.

2.3 Procedure

Participants were introduced to the experiment as follows: they would need to make a trip during rush hour for an appointment at 8:30 a.m. Our participants were students, so we chose their home university, located in the city of Rotterdam, as the point of arrival in order to enhance the external validity of the experiment. The points of departure were selected at five random locations in the periphery of the city of Rotterdam, each approximately ten kilometers away in all directions from the point of arrival. Participants received these points of departure in random order throughout the experiment in order to ensure that their specific understanding of a particular location did not influence their decisions. They were informed that four different transportation options available to them: (1) their own car, (2) public transportation, (3) exclusive ridesharing, and (4) pooled ridesharing. They were asked to select their preferred transportation mode for the trip from these modality options forty times. The basic characteristics of these modes of transportation were provided to them; see Table 1.

When participants moved from one phase of the experiment to another, additional information was provided prior to making subsequent choices. In the first phase, only information on the departure time and the time taken for each transportation mode were provided. In phase two, participants, who received environmental information manipulation, were informed that it presently was possible to check the environmental information of the trip. This was also done for the price of each transportation mode in phase three, and for traffic information in phase four. Figure 1 indicates the flow of the experiment over the forty rounds. In each round, participants chose their preferred option, and for that round, received the result of the trip as a feedback. Participants received feedback about their trip, involving the time taken for the trip they had just made, whether they would run late, and, depending on the phase, the environmental impact, and the cost of the trip. A cumulative summary of the results was also provided.

Table 1. Basic characteristics and explanation of each mode of transportation

Characteristics	Private car	Public transportation	Ride sharing (Exclusive)	Ride sharing (Pooled)
Time taken	**Fast** but prone to significant delay due to traffic condition. Parking might take sometimes especially in rush hour	**Slow** but less prone to traffic due to priority / separate lane	**Fast** but prone to significant delay due to traffic condition. No parking is required Waiting time might take sometimes	**Fast** but prone to significant delay due to traffic condition. No parking is required Slight detour or wait for the other traveller will be necessary
Privacy	**Very High** There will only be you	**Low** There will be many travellers	**High** There will be you and the driver	**Medium** There will be you, the driver, and one other traveller in the car
Comfort	**High**	**Low** During peak hours public transportation is crowded	**High**	**High**

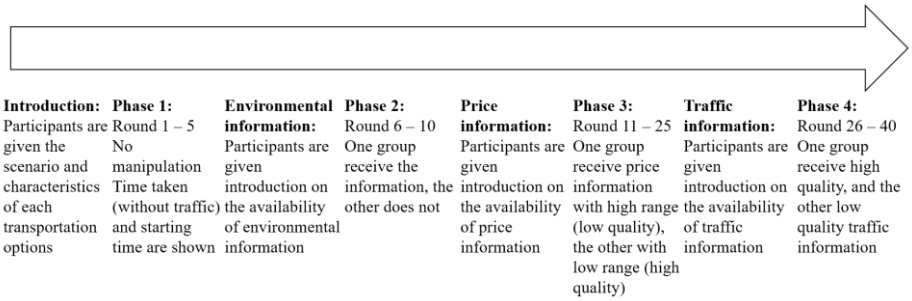


Fig. 1. The flow of the forty rounds of mode choice decision

Table 2. Simulation of traffic condition

Traffic condition at *t*-1	Choice at *t*-1	Traffic at time t due to choice at *t*-1	Choice at *t*	Traffic at time *t* due to choice at *t*	Total traffic time at *t*
Yellow indicator (10- 15 minutes)	Car/RSE	increase traffic indicator to red	Car/RSE/ RSP	Increase traffic by 1-3 minutes	15 to 20 + 1 to 3 = *16 to 13 minutes*
	Car/RSE	increase traffic indicator to red	Car/RSE/ RSP	Increase traffic by 1-3 minutes	15 to 20 + 1 to 3 = *16 to 23 minutes*
	RSP/PT	Decrease traffic indicator to green	Car/RSE/ RSP	Increase traffic by 1-3 minutes	5 to 10 + 1-3 = *6 to 13 minutes*

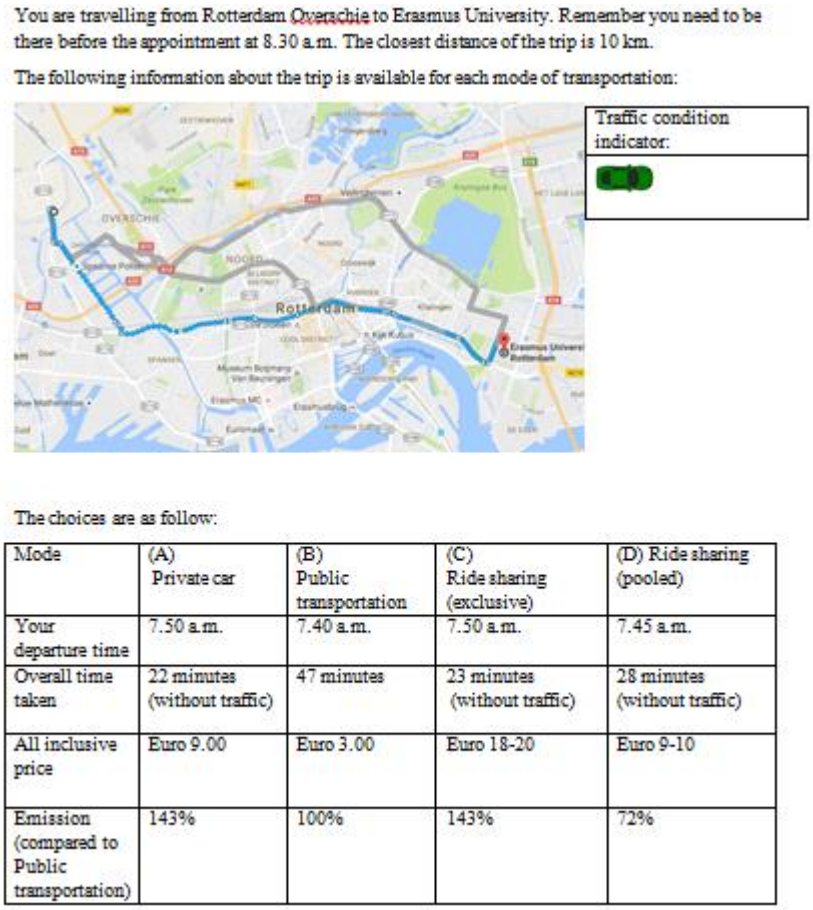
To simulate the traffic conditions in the experiment so that it mirrors the real behavior that taking single occupancy vehicle would increases while taking public transportation or sharing a ride would decrease traffic congestion, we consider the following scenario: opting for private car and exclusive ridesharing would increase the traffic condition in the next round, whereas opting for public transportation and pooled ridesharing would decrease the traffic condition in the next round. We used this approach to eliminate the need to run parallel experiments among a group of participants. The traffic condition was calculated for all forty rounds, but only in phase four, a traffic information was provided to participants. Also, participants' choices affected the current traffic only by a small amount (1-3 minutes additional time from free flow time taken), since participant's choices actually would add to the current traffic level. Note, though, that when participants chose public transportation, traffic condition did not impact participants' travel time. The explanation of the traffic condition in this experiment is shown in Table 2. An example of the information provided in the final phase (phase 4) is in Figure 2.

After completing the transportation information provision experiment, participants were asked to fill out a questionnaire that contained several psychological measures and demographic questions.

2.4 Manipulations

Environmental information. In the second phase of the experiment, participants were randomly assigned to two different groups: with or without environmental information. In the environmental information group, participants were provided with information about the emissions of transportation mode relative to that of public transportation. We offered our participants a percentage value rather than nominal value of emitted CO2 to make the comparison more salient to them. The emission data were collected from 9292.nl [16], a public transportation information system in the Netherlands (the country

of the study). Private car and exclusive ridesharing were assumed to have emission values equal to a medium size car, corresponding to 143% of public transportation emissions. For pooled ridesharing, this value was divided between the two passengers, leading to emission equals to 72% of public transportation. No information about environmental impact was given to the group without environmental display.



Mode	(A) Private car	(B) Public transportation	(C) Ride sharing (exclusive)	(D) Ride sharing (pooled)
Your departure time	7.50 a.m.	7.40 a.m.	7.50 a.m.	7.45 a.m.
Overall time taken	22 minutes (without traffic)	47 minutes	23 minutes (without traffic)	28 minutes (without traffic)
All inclusive price	Euro 9.00	Euro 3.00	Euro 18-20	Euro 9-10
Emission (compared to Public transportation)	143%	100%	143%	72%

Fig. 2. An example of information displayed to participants in Phase 4

Price information. Price information was given to participants from round 11 onwards. Thus, no price information was shown in round 1 to 10. Displayed price were different for each transportation mode and based on the data from the Dutch Touring Club (ANWB) for private cars, on the data from 9292.nl [16] for public transportation, and on the data from Uber.com [17] for exclusive as well as pooled ridesharing, respectively. The prices were Euro 9.00 for private car, Euro 3.00 for public transportation, approximately Euro 19.00 for exclusive ridesharing, and approximately Euro 9.50 for pooled ridesharing. In the third phase of the experiment, participants were randomly

assigned to two different groups: low or high price range of ridesharing services. The price of ridesharing services is typically shown in terms of a range. Following Li et al. [18], we provided different price ranges to participants. In the low price range group, the ridesharing options had a smaller price range compared to the high price range group. Specifically, the low price range had a 10% price range, varying between Euro 18.00 to Euro 20.00 for exclusive ridesharing, and Euro 9.00 to Euro 10.00 for pooled ridesharing. In the high price range group, the price range was 30% between Euro 16.00 to Euro 22.00 for exclusive ridesharing, and Euro 8.00 to Euro 11.00 for pooled ridesharing.

Traffic information. Traffic information was provided to participants from round 26 onwards. Thus, no traffic information was shown in round 1 to 25. We borrowed the approach of showing an icon to indicate traffic from Bouman et al. [19]. Traffic information was provided as a single green car, two yellow cars, and three red cars. These indicators represented different estimates of traffic density. A single green car was associated with 5-10 minutes of additional travel time; two yellow cars were associated with 10-15 minutes of additional travel time; three red cars were associated with 15-20 minutes of additional travel time due to traffic. Participants were informed about these indicators at the end of round 25, before they were asked to continue with their mode choices. The quality of the information provided to participants depend on whether participants were in high or low traffic quality information group. In the low traffic quality information group, the pre-trip traffic information provided to participants was randomly selected from the three traffic states and, hence, did not provide any details regarding the traffic situation in that particular round. Meanwhile, in the high traffic quality information group, the pre-trip traffic information provided to participants was a perfect match to the traffic state shown by the traffic indicator of that particular round.

2.5 Individual psychological ownership

We measured the individual psychological ownership level of our participants using a scale based on van Dyne and Pierce [20]. As the scope of the study only concerned individual psychological ownership, following Paundra et al. [15], we took four out of the seven original items that were used to measure individual feelings of psychological ownership, and modified these items to fit the vehicle ownership context. When the original item was formulated as “I feel a very high degree of personal ownership for this organization”, for the present research we modified that item into “I feel a very high degree of personal ownership for this vehicle” (1 = strongly disagree, 7 = strongly agree). The other items that we used were rewritten as follows: “This is MY vehicle”, “I sense that this is MY vehicle”, and “It is hard for me to think about this vehicle as MINE” (reversed). The Cronbach’s α for this four-item scale was 0.91.

3 Results

We evaluate the impact of environmental, price, and traffic information as well as psychological ownership on people's decision to take a particular transportation option by means of a Multinomial logit Bayesian regression with random effects. We included the lagged choices as participant-specific effects as for each participant, their previous choice (round t-1) have influence on the traffic level they would face in the current round (round t). The advantage of using the Bayesian approach is that we can incorporate and evaluate the group-level coefficients [21]. This article use R's brms package that implement Hamiltonian Monte Carlo, which is advantageous as it converges more quickly. For this analysis, we consider the use of weakly informative priors. Table 3 presents the results.

The results suggest that environmental information has a positive influence on participants' likelihood to opt for public transportation (estimate = 0.59, est. error = 0.13, 95% CI [0.34; 0.83]), and pooled ridesharing (estimate = 3.47, est. error = 0.30, 95% CI [2.86; 4.07]). The estimate of price information also showed that participants favored public transportation when they know that this transportation is cheaper than private vehicle (estimate = 1.36, est. error = 0.19, 95% CI [1.00; 1.73] and estimate = 1.40, est. error = 0.19, 95% CI [1.03; 1.79], respectively). Furthermore, there is negative estimate of likelihood of participants to opt for exclusive ridesharing in low range (estimate = -0.90, est. error = 0.28, 95% CI [-1.45; -0.36]), and in high range price information groups (estimate = -0.99, est. error = 0.29, 95% CI [-1.57; -0.44). Similarly, negative estimates are observed for pooled ridesharing in low range (estimates = -0.89, est. error = 0.24, 95% CI [-1.35; -0.38]) and in high price range manipulation groups (estimate = -0.86, est. error = 0.25, 95% CI [-1.32; -0.35]). The estimates for high and low price range manipulation groups did not seem to differ for the two ridesharing services. Meanwhile, traffic information provision seemed to negatively impact the likelihood of choosing public transportation in both low (estimate = -0.65, est. error = 0.17, 95% CI [-0.98; -0.30]) and high quality level of the information (estimate = -0.88, est. error = 0.16, 95% CI [-1.20; -0.55]). For exclusive ridesharing, only high quality traffic information has negative impact (estimate = -0.80, est. error = 0.31, 95% CI [-1.40; -0.21], while for pooled ridesharing, quality traffic information did not seem to have an impact.

When considering people's individual level of psychological ownership, it can be seen that the more psychologically attached people were to their vehicles, the less likely they would choose alternative transportation modes, especially public transportation (estimate = -0.18, est. error = 0.06, 95% CI [-0.31; -0.05]). Individual psychological ownership also moderated the influence of low quality traffic information in the likelihood of opting for public transportation (estimate = 0.22, est. error = 0.07, 95% CI [0.09; 0.35]), and pooled ridesharing (estimate = 0.38, est. error = 0.17, 95% CI [0.05; 0.71]), as well as the impact of environmental information provision for exclusive ridesharing (estimate = 0.39 est. error = 0.17, 95% CI [0.05; 0.73]).

Table 3. Random effects multinomial logit model

	Public transportation			Exclusive Ridesharing			Pooled Ridesharing		
	Esti-mate	Est. Error	CI	Esti-mate	Est. Error	CI	Esti-mate	Est. Error	CI
Intercept	-0.10	0.16	[-0.42; 0.20]	-3.26	0.26	[-3.80; -2.78]	-4.57	0.32	[-5.21; -3.97]
Environment	0.59	0.13	[0.34; 0.83]	0.05	0.27	[-0.47; 0.58]	3.47	0.30	[2.86; 4.07]
Price1	1.36	0.19	[1.00; 1.73]	-0.90	0.28	[-1.45; -0.36]	-0.89	0.24	[-1.35; -0.38]
Price2	1.40	0.19	[1.03; 1.79]	-0.99	0.29	[-1.57; -0.44]	-0.86	0.25	[-1.32; -0.35]
Traffic1	-0.65	0.17	[-0.98; -0.30]	-0.35	0.31	[-0.95; 0.24]	-0.29	0.24	[-0.77; 0.17]
Traffic2	-0.88	0.16	[-1.20; -0.55]	-0.80	0.31	[-1.40; -0.21]	0.24	0.19	[-0.14; 0.61]
PO	-0.18	0.06	[-0.31; -0.05]	-0.11	0.13	[-0.37; 0.15]	-0.08	0.17	[-0.42; 0.27]
Env*PO	0.03	0.08	[-0.12; 0.18]	0.39	0.17	[0.05; 0.73]	0.13	0.17	[-0.21; 0.47]
Price1*PO	-0.02	0.07	[-0.16; 0.11]	-0.36	0.19	[-0.73; 0.01]	-0.18	0.13	[-0.45; 0.07]
Price2*PO	-0.07	0.07	[-0.20; 0.07]	0.12	0.22	[-0.29; 0.55]	0.07	0.13	[-0.45; 0.33]
Traffic1*PO	0.22	0.07	[0.09; 0.35]	0.09	0.25	[-0.38; 0.58]	0.38	0.17	[0.05; 0.71]
Traffic2*PO	0.08	0.06	[-0.03; 0.19]	-0.13	0.21	[-0.57; 0.27]	-0.03	0.10	[-0.22; 0.16]
SD_Intercept (Rounds)	0.38	0.06	[0.29; 0.51]	0.21	0.12	[0.01; 0.47]	0.32	0.09	[0.16; 0.50]
Lagged PT	2.12	0.13	[1.87; 2.38]	0.46	0.35	[0.02; 1.25]	0.80	0.28	[0.28; 1.38]
Lagged RSE	0.39	0.29	[0.02; 1.06]	3.06	0.54	[2.10; 4.18]	1.83	0.46	[0.91; 2.78]
Lagged RSP	0.78	0.22	[0.34; 1.20]	1.05	0.70	[0.04; 2.56]	2.36	0.33	[1.73; 2.05]
SD_Intercept (Participants)	1.18	0.08	[1.03; 1.34]	2.05	0.22	[1.64; 2.52]	2.40	0.26	[1.92; 2.94]

Note: Lagged dependent variables are random effects at participant level; Price1 refers to low range, and Price2 to high range. Traffic1 refers to low quality and Traffic2 to high quality. PO is individual psychological ownership; CI is the 95% confidence interval; Prior distributions are Normal (0,6) for respective choices and t (3,0,10) for the random intercepts; The model is run with 4 chains each with 2000 iterations (first 1000 is a warmup);

4 Conclusion

In this study, the effectiveness of information provision as a soft policy intervention that nudge commuters away from their private vehicles to alternative transportation modes such as public transportation or ridesharing services was investigated. We observed that providing information regarding environment, price, and traffic made people turn towards pooled ridesharing service or public transportation instead of their own car. Interestingly, we also found that the quality of traffic information have varying impact to commuters. For instance, high quality traffic information enabled people to use private cars than alternative transportation. The quality of ridesharing service price information did not seem to have an impact. In addition, and consistent with prior research [15], individual psychological ownership moderated the influence of information provision on commuters' decisions.

References

1. The United Nations, https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html
2. Eriksson, L., Garvill, J., Nordlund, A. M.: Acceptability of single and combined transport policy measures: The importance of environmental and policy specific beliefs. Transportation Research Part A: Policy and Practice 42, 1117-1128 (2008).
3. Geng, J., Long, R., Chen, H., Li, Q.: Urban residents' response to and evaluation of low-carbon travel policies: Evidence from a survey of five eastern cities in China. Journal of environmental management 217, 47-55 (2018).
4. Klein, I., Ben-Elia, E.: Emergence of cooperation in congested road networks using ICT and future and emerging technologies: A game-based review. Transportation Research Part C: Emerging Technologies 72, 10-28 (2016).
5. Zha, L., Yin, Y., Yang, H.. Economic analysis of ride-sourcing markets. Transportation Research Part C: Emerging Technologies 71, 249-266 (2016).
6. Li, Z., Hong, Y., Zhang, Z.: An empirical analysis of on-demand ride-sharing and traffic congestion. In 50th Hawaii International Conference on System Sciences, Hawaii, pp 4-13 (2017).
7. Greenwood, B. N., Wattal, S. Show me the way to go home: An empirical investigation of ride-sharing and alcohol related motor vehicle fatalities. MIS Quarterly 41, 163-187 (2017).
8. Gaker, D., Vautin, D., Vij, A., Walker, J.L.: The power and value of green in promoting sustainable transport behavior. Environmental Research Letters 6, 034010 (2011).

9. Avineri, E., Waygood, E.O.D.: Applying valence framing to enhance the effect of information on transport-related carbon dioxide emissions. Transportation Research Part A: Policy and Practice 48, 31-38 (2013).
10. Cairns, S., Sloman, L., Newson, C., Anable, J., Kirkbride, A., Goodwin, P.: Smarter choices: Assessing the potential to achieve traffic reduction using 'soft measures'. Transport Reviews, 28, 593-618 (2008).
11. Möser, G., Bamberg, S.: The effectiveness of soft transport policy measures: A critical assessment and meta-analysis of empirical evidence. Journal of Environmental Psychology, 28, 10-26 (2008).
12. Brög, W., Erl, E., Ker, I., Ryle, J., Wall, R.: Evaluation of voluntary travel behaviour change: Experiences from three continents. Transport Policy, 16, 281-292 (2009).
13. Avineri, E., Prashker, J.N.: The impact of travel time information on travelers' learning under uncertainty. Transportation 33, 393-408 (2006).
14. Ben-Elia, E., Di Pace, R., Bifulco, G. N., Shiftan, Y.: The impact of travel information's accuracy on route-choice. Transportation Research Part C: Emerging Technologies 26, 146-159 (2013).
15. Paundra, J., Rook, L., van Dalen, J., Ketter, W.: Preferences for car sharing services: Effects of instrumental attributes and psychological ownership. Journal of Environmental Psychology 53, 121-130 (2017).
16. 9292.nl, https://9292.nl/en/
17. Uber.com, https://www.uber.com/en-NL/ride/
18. Li, T., Kauffman, R.J., Van Heck, E., Vervest, P., Dellaert, B.G.: Consumer informedness and firm information strategy. Information Systems Research 25, 345-363 (2014).
19. Bouman, P.C., Kroon, L., Vervest, P., Maróti, G.: Capacity, information and minority games in public transport. Transportation Research Part C: Emerging Technologies 70, 157-170 (2016).
20. Van Dyne, L., Pierce, J. L.: Psychological ownership and feelings of possession: Three field studies predicting employee attitudes and organizational citizenship behavior. Journal of Organizational Behavior 25, 439-459 (2004).
21. Bürkner, P. C.: brms: An R package for Bayesian multilevel models using Stan. Journal of Statistical Software, 80, 1-28 (2017).

The influence of local terrain variations on spectral analysis of insolation time-series in Sierra Nevada (Granada province, southern Spain)

J. Sánchez-Morales[1], E. Pardo-Igúzquiza[2] and F. J. Rodríguez-Tovar[1]

[1] Universidad de Granada, Avd. Fuentenueva s/n, 18071 Granada (Spain)
[2] Instituto Geológico y Minero de España, Ríos Rosas 23, 28003 Madrid (Spain)
josesanmor@correo.ugr.es; e.pardo@igme.es; fjrtovar@ugr.es

Abstract. The widespread availability of marine and terrestrial climate proxies recording global events allows the identification of past and present climatic cycles. Basically, these proxies can be understood as different ecological and geological processes ruled in first instance by orbital phenomena, transforming energy inputs (usually from the Sun) into noisy signals or time-series. Characterizing past insolation by means of orbital solutions helps to improve the understanding of the global climatic context inherent to these systems. However, we want to raise the attention on a type of noise, which is the effect produced by the terrain on blocking the Sun light in mountainous areas. This article focuses on the effects of different orography settings on the local insolation calculations and ultimately, it shows how these alterations can produce different spectral signals.

Keywords: Power Spectrum, orbital cycles, Milankovitch band.

1 Introduction

The Milankovitch cycles theory [1] explains the advances and retreats of Earth's ice ages as the consequence of having different insolation rates at a latitude of about 65 degrees North. These insolation changes respond to quasi-periodic variations of Earth's orbital parameters and known today as Milankovitch cycles: precession, obliquity and eccentricity. Almost forty years later the theory was tested [2] and the echoes of that pioneer work remain valid for the most part. A good summary on the state-of-the-art on this issue can be found [3]. In a nutshell, the length of the Milankovitch cycles in the geological record is well referred in many scientific papers [4]: Precession has a length of 19 and 24 ky (extremes at 14 and 28 ky), obliquity about 41 ky (extremes at 28 and 54 ky) and eccentricity of 100 and 400 ky. Precession is the combined movement of the axial precession and the apsidal precession; the first one (clockwise) is the absolute movement of the Earth's rotation axis describing a cone in space (one cycle takes about 26 ky) and the second one (counterclockwise) is the

precession of the Earth's orbit or the precession of the line connecting the apsides of Sun and Earth (one cycle takes about 112 ky). Obliquity is the variation (tilt) of the Earth's axis of rotation and its current value is about 23.44° but it can oscillate between roughly 22.1° and 24.5°. Eccentricity is the change on the shape of Earth's orbit from almost circular to highly elliptical. Berger in 1978 [5][6] developed an astronomical solution (Ber78) that resolves the referred three orbital parameters and is valid +/- 1e6 years back. Another well-established model was developed by Laskar et al. in 2004 [7] and later refined [8]. The Ber78 model has been chosen for this article and it takes as inputs solely the latitude of the study region as well as the age or epoch to be assessed, in which 1950 is year 0 in the model. The mathematical and/or astronomical basis behind these models is highly detailed and complex, and it is beyond the scope of this article.

Orbital forcing plays a huge role in the amount of insolation that any region on Earth receives. It is widely known for instance that in the northern hemisphere, the Sun's elevation in the sky is higher during Summer and lower during Winter. Also, the Sun's azimuth (the horizontal component of the relative direction of the Sun) at sunset is due East in both the Spring and Autumn equinoxes, but moves further North in Summer (the furthest occurs on the Summer solstice) and moves further South in Winter (the furthest occurs on the Winter solstice). All these phenomena produce longer days in the Summer and shorter days in the Winter and introduce variations in the amount of daily insolation. However, in mountain regions the presence of complex orography settings can have a profound impact on the rates of annual insolation, and in the derived power spectra of the analyzed insolation time-series too, as we present in this work.

2 Methodology

This study uses a GIS group of techniques known as 'visibility analysis', as well as the spectral analysis to detect the variation of orbital cycles in the interval 0-100 ky. The aim is to introduce elevation data as noise in the signal and see how the spectral signal reacts to it.

First, a high-resolution digital elevation model (DEM) is needed for the analysis, as different observation points must be first allocated across the study area. These locations should have different orographic settings. For this purpose, mountain regions are ideal as they have greater terrain variability as opposed to flat areas. The first step in the visibility analysis is to calculate the skyline for each location. The skyline is the polyline that separates the terrain from the sky. It is in fact the furthest point an observer could see from that location. The skyline can be represented as a skyline graph which is a 360 degrees visualization, or polar graph, in which for each horizontal angle or azimuth, an elevation value or zenith is provided. The zenith angle can vary from 0 to 90 degrees. The azimuth and zenith values extracted from the polar graph,

can be used to determine when and where the Sun is visible in the horizon from that location.

The second step of the methodology process is to calculate the position of the Sun in the sky for each day through the year. There are many computer programs and/or libraries available [9][10][11] that can perform accurate solar calculations. In this study we have used the NOAA Solar Position calculator [11] which is able to provide with accurate times, azimuth and zenith values of the Sun for any given location and date. The procedure consists of counting the total amount of hours and minutes in which the Sun is visible in the sky for each location without considering the visibility analysis, and then doing the same but by considering the effect of the terrain on blocking the Sun. By this way, we can obtain what we have called the 'visibility ratio'; one per day and 365 values for the whole year. We have selected the year 2,000AD for the calculations and use the same ratios for the rest of the epochs. The made assumption is that the terrain configuration has not changed considerably in the study area for the latest 100 ky.

The third step is to calculate the insolation values for all locations using Ber78 as astronomical solution [12][13], and for the period 0-100 ky with a sample interval of 0.1 ky. We first obtain an idealized insolation curve considering clear skies and no terrain. Then, for the 365 days of each run year we multiply each insolation value by the corresponding visibility ratio. The model Ber78 calculates the insolation starting at the Vernal Equinox (day 0), so the visibility ratios sequence starts on that date. It is very important to understand that one measurable effect of precession is effectively moving the date in which the equinoxes and/or solstices occur in the calendar, but the position of the Sun in the sky is spatially consistent for repeated astronomical positions. The Sun is always at its maximum zenith in the Summer solstice in the NH, equinoxes will always have equal day and night times, etc. For this reason it is very helpful to visualize the 3D movement of the Earth in the past [12].

Finally, we have used the spectral analysis as the statistical technique to quantify the precession and obliquity associated frequencies in Sierra Nevada. The power spectrum estimator has been the smoothed Lomb-Scargle periodogram which in this case has worked with even time-series. The computer program SLOMBS [14] has been used and it has two main features; it evaluates the statistical significance of the peaks by the Monte Carlo permutation test as neighbouring frequencies are highly correlated, and it can adjust the statistical significance by smoothing the periodogram. In this second case, linear smoothing with 3 terms was applied to the raw periodogram. Four output files are generated by the program: The Lomb-Scargle spectrum, the achieved confidence level spectrum, the mean spectrum of permutations and the phase spectrum.

The input parameter values necessary for running the program have been optimized for dealing with insolation values for the last 100 ky, and the associated Milankovitch cycles. Thus, the six inputs parameters have been: '0.0001' for the highest frequency

to evaluate (>10 ky cycles detection), ‘200’ for the number of frequencies in the interval, ‘2,000’ for the number of permutations, ‘75,654’ for the integer random number, linear smoothing as enabled and ‘3’ for the number of smoothing terms.

3 Case Study

The Spanish Geographic Institute (IGN) has DEM coverage at 5m per pixel for the whole country [15]. Nine different sunlight conditions have been selected across the study area. The locations and their coordinates are shown (Table 1).

Table 1. Coordinates and elevation of studied locations within the study area.

Location	Coordinates (Lat / Lon)	Elevation (m.s.l)
P01	37.060 / -3.315	2,893.62
P02	37.160 / -3.510	841.16
P03	37.043 / -3.460	1,901.35
P04	37.143 / -3.208	1,660.49
P05	37.032 / -3.249	1,652.41
P06	37.053 / -3.312	3,470.42
P07	37.090 / -3.328	1,751.71
P08	37.011 / -3.371	1,877.58
P09	37.060 / -3.368	3,067.64

The extent of the visibility polygons 6 and 9 is well beyond the study area (Fig. 1), because no terrain barriers exist in the vicinity for certain azimuth angles. For these two, the polygons have been clipped for practical reasons as the Sunlight is the same.

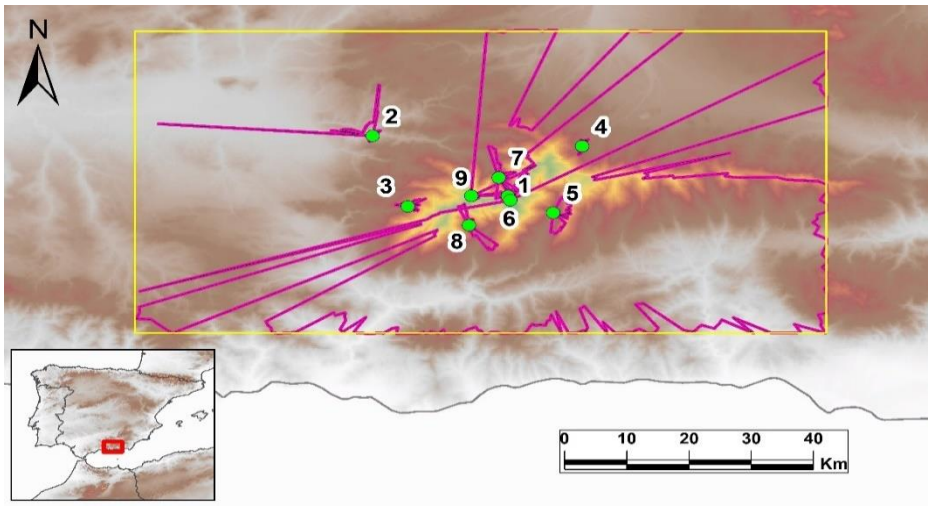
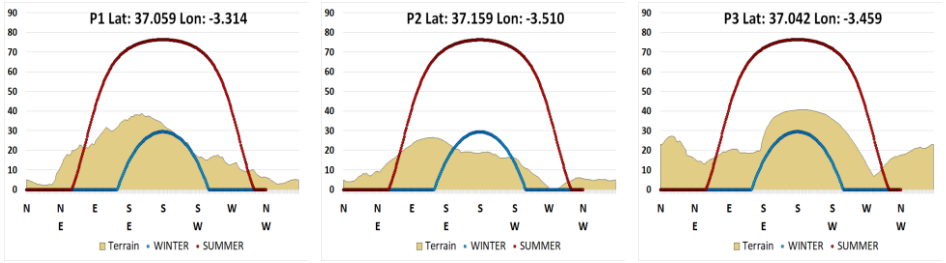


Fig. 1. Study area and selected locations around the Sierra Nevada mountain massif.

The highest and lowest Sun paths for all 9 locations in one year is shown (Fig. 2). On each graph three elements have been plotted; the terrain elevation (brown fill), the lowest insolation given by the Sun on the Winter solstice (blue line) and the highest insolation given by the Sun on the Summer solstice (red line). The Sun is only visible in the horizon when it is above the terrain elevation, otherwise the observer remains in the shade. The y-axis represents the Sun zenith (0-90º) and the x-axis represents all cardinal points (0-360º). As an example, in the Summer solstice and for the latitude of the study area, the sunrise occurs near the North-East and the sunset occurs near the South-West.



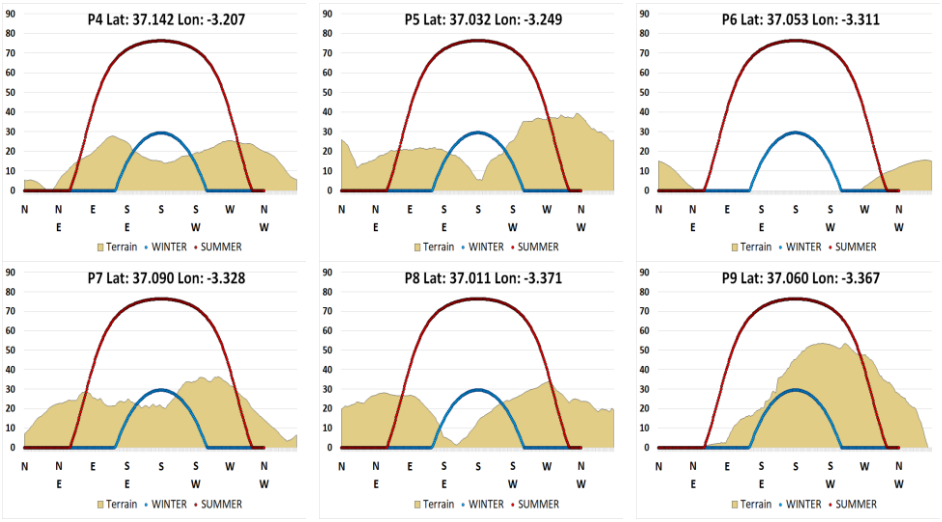


Fig. 2. Sun position in the sky in the Summer solstice (red line) and Winter solstice (blue line) of the year 2,000 for all studied locations, including the terrain-visibility analysis in the background, indicating above which specific azimuth and elevation, the Sun can be spotted in the horizon.

If we compute the number of hours in which the Sun is visible in the horizon for all studied locations without considering the terrain elevations, we get a sinusoidal curve (Fig. 3a). As all locations have almost identical latitude values, the number of daily hours is almost identical too. However, by introducing the visibility analysis in the process the count of daily hours of direct sunlight turns to be unique for each location as the terrain elevation is also unique (Fig. 3b). The amount of daylight decreases particularly in the Winter when the Sun is lower in the horizon. Some locations (P01, P03 and P09) have 0 hours of direct Sunlight during those days, and others (P06) are barely affected.

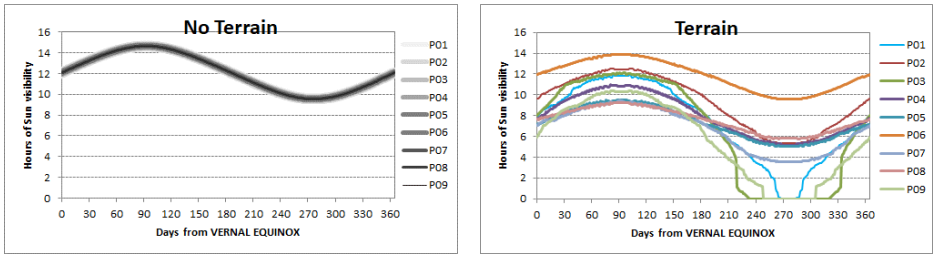


Fig. 3. Count of daily hours of direct sunlight through the year 2,000 for all locations. Fig. 3a (left) without considering the local terrain elevations. Fig. 3b (right) considering the local terrain elevations. Count always starts on the 20th of March (present Vernal Equinox).

The decrease of daylight hours due to the presence of terrain elevations could be represented as daily ratios: the daylight hours considering terrain are divided by the daylight hours without considering terrain; 1 would mean Sun is visible at all times, and 0 would mean that the Sun is completely blocked by the terrain (Fig. 4a).

Due to obliquity, the Earth's axis tilt can oscillate between around 22.1 and 24.5º, being its current value at 23.43º and decreasing. In our methodology we haven't updated the visibility ratios, and we have considered the same ratios derived from year 2,000 back to all past time-series. The potential effect of obliquity on visibility ratios is similar as to change the latitude for that location. Using the maximum, the current and the minimum Earth's tilt value we have assessed the potential impact of obliquity on visibility ratios for location P01 (Fig. 4b).

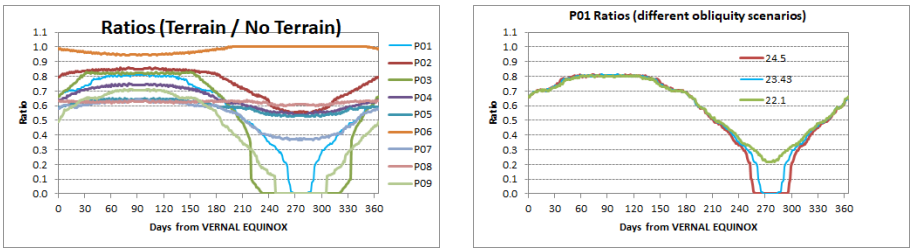


Fig. 4. Daily ratios of direct sunlight through the year 2,000. Fig. 4a (left) for all locations considering the local terrain elevations. Fig. 4b (right) for location P01 only at different obliquity scenarios. Count starts on the 20th of March (present Vernal Equinox).

As we can see (Fig. 4b) the current Earth's tilt produces theoretically an effect very similar to the maximum obliquity value. Again, the differences are more evident in Winter when the Sun is nearer the horizon and the terrain influence is higher.

Moving into the past climate scenario, we have calculated the insolation for each location and for the latest 100 ky, using the Ber78 model and 0.1 ky as sample interval. Therefore, the series length is 1,000 records. In the first run, no terrain has been considered. The results (Fig. 5) show almost identical insolation curves, as the only input to the model that changes every run is the fixed latitude for each location.

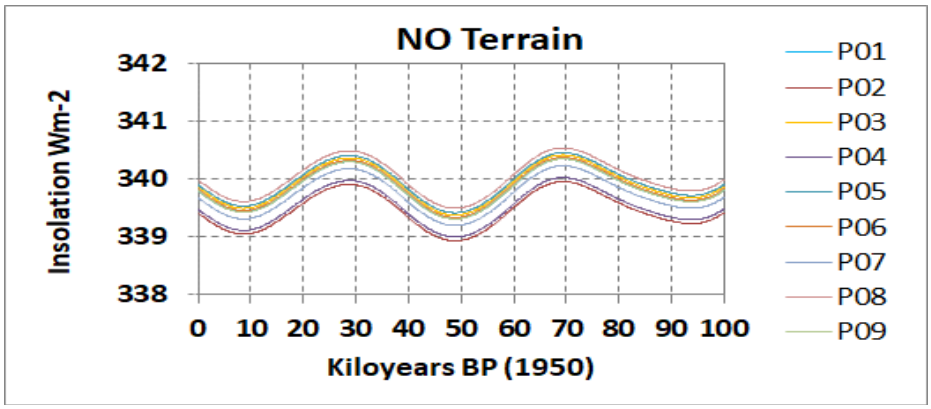


Fig. 5. Insolation for the latest 100 ky considering no terrain and clear skies for all locations and using Ber78 solution.

To introduce the terrain elevations into the analysis, the direct Sunlight ratios on each location derived from the visibility analysis have been used. As the Ber78 model starts calculating the insolation on the Vernal Equinox (day 0), the ratio value starting on the 20th of March has been set up as first value into the series. All the insolation curves with and without the visibility analysis have been obtained (Fig. 6).

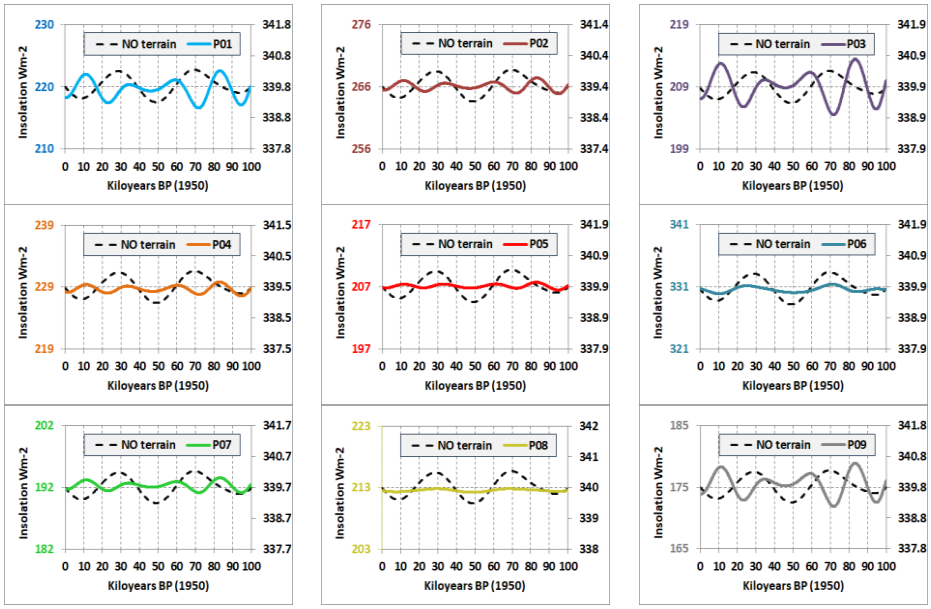


Fig. 6. Insolation for the latest 100 ky considering terrain and clear skies for all locations and using Ber78 solution

4 Spectral analysis

The spectral analysis initially has been conducted without considering the terrain analysis. The power spectra of all long-term insolation series (100 ky) have been calculated, as a comparative (Fig. 7) and for each location separately (Fig. 8).

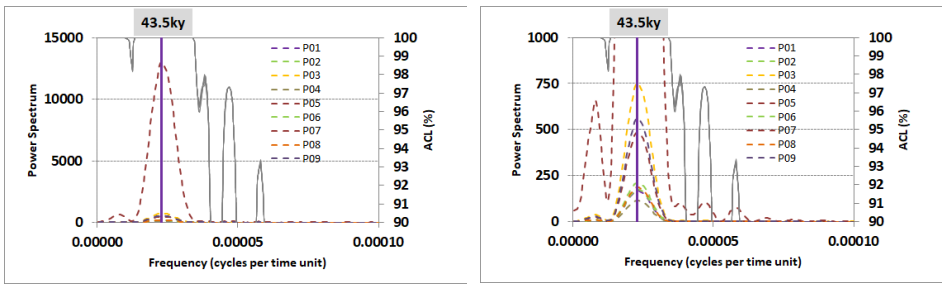


Fig. 7. All locations' combined power spectra of the insolation time-series considering no terrain and clear skies using Ber78 solution.

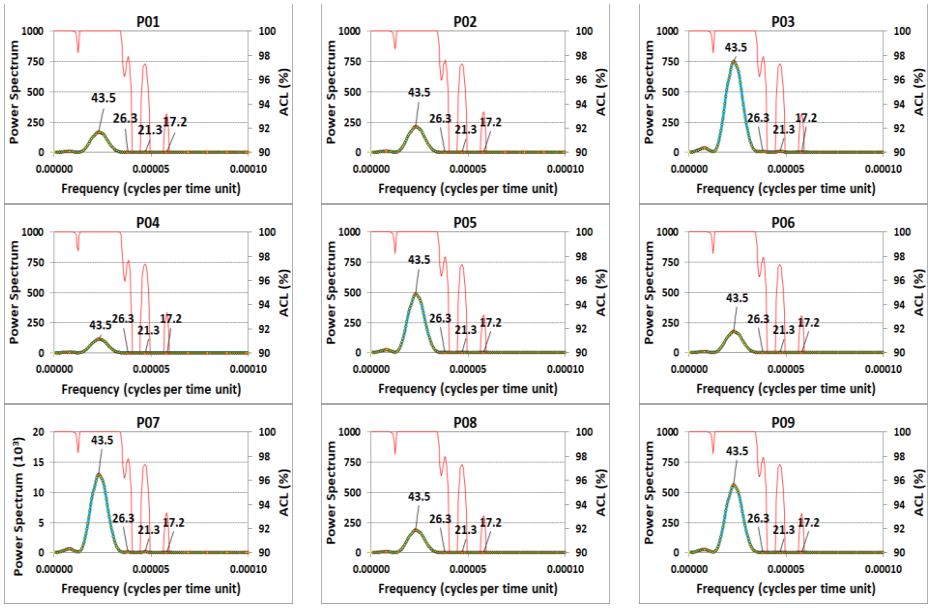


Fig. 8. Individual power spectrum of the insolation time-series for each location, considering no terrain and clear skies using Ber78 solution.

The obliquity cycle has been detected as the most significant spectral peak with an average value of 43.5 ky. Other cycles have been detected at 26.3, 21.3 and 17.2 ky.

After having assessed the spectral analysis without the terrain variations, now we have introduced that element into the analysis. In theory, the more terrain variations exist for a given location the more noise is introduced into the insolation series. The power spectra of all these new series have been calculated (Fig. 9) and plotted as individual graphs (Fig. 10).

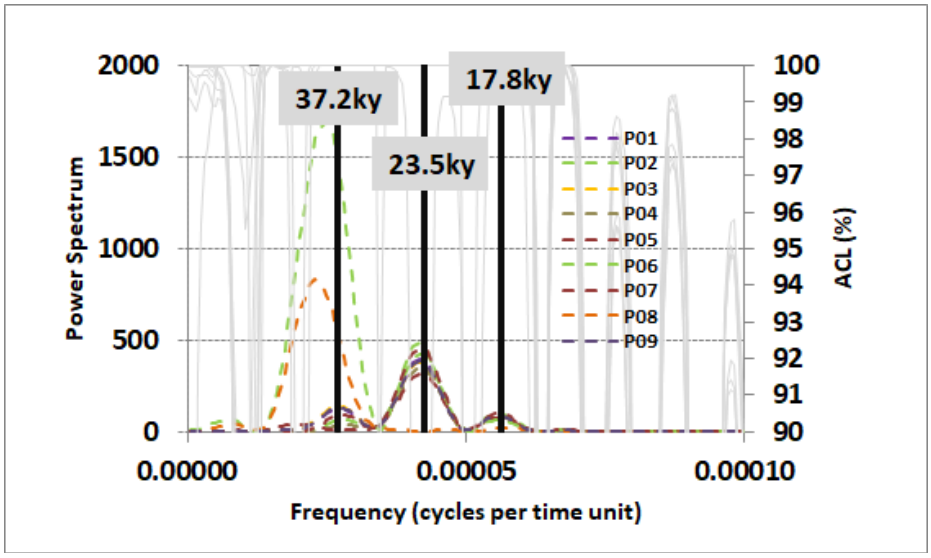
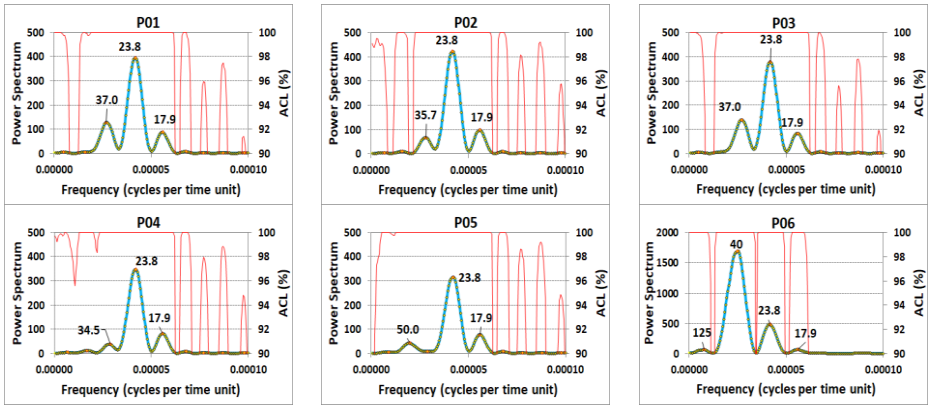


Fig. 9. Power spectra of all insolation time-series considering terrain and clear skies for all locations and using Ber78 solution.



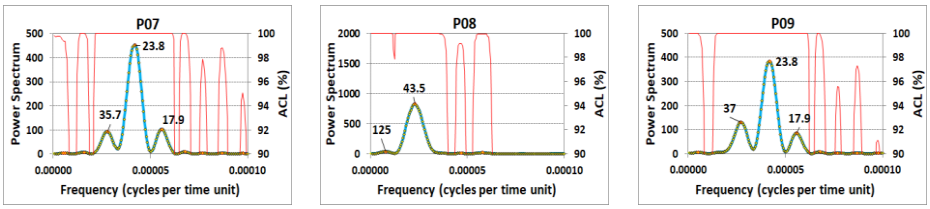


Fig. 10. Each individual power spectrum for all insolation time-series considering terrain and clear skies using Ber78 solution.

The insolation signal in the power spectra has been modulated by creating one central peak around 23.5 ky, and 2 secondary peaks at around 37.2 ky and 17.8 ky in most of the series. Locations P06 and P08 are less affected by this process, and they keep the original peaks at around 40 and 43.5 ky respectively.

5 Interpretation

For the case study, the introduction of terrain noise into the insolation series, produces from a climatic point of view the same effect (in the northern hemisphere) as if the studied region was further North. Indeed, there are less insolation hours than in the scenario without considering terrain because the terrain blocks the Sun rays, especially at the lowest Sun elevations, i.e. at around Sunrise and Sunrise times. In occasions, the terrain completely blocks the Sun path for a whole day if the elevations are very high. This effect is more noticeable in Winter when the Sun elevation is lower and the daylength is shorter.

The astronomical model Ber78 has turned out to be highly dependent on the obliquity cycle (43.5 ky) as the spectral analysis without terrain noise shows. When multiplying the daily insolation by a factor or visibility ratio, between 0 and 1, we are in fact using a signal filtering. We could consider this type of filtering as a continuous one as the overall power spectra has indeed the same frequency components. However, the accumulated effect over the years of the terrain has been acting as a low-pass filter. Thus, high frequencies have been more attenuated (obliquity) than the low frequencies (precession). A summary of all cycles and their variations is presented (Table 2).

Table 2. Comparative between all detected cycles above 90% of ACL, using insolation model Ber78 with (no terrain) and without (terrain) visibility analysis in all locations. Main spectral peaks according to their value of power spectrum have been marked in bold.

Location	No Terrain (Ky)	Terrain (Ky)
P01	125.0, **43.5**, 26.3, 21.3, 17.2, 14.5, 12.7, 11.2, 10.1	200.0, 62.5, **37.0**, **23.8**, **17.9**, 14.7, 13.0, 11.5, 10.2
P02	125.0, **43.5**, 26.3, 21.3, 17.2, 14.5, 12.7, 11.2, 10.1	142.9, 62.5, **35.7**, **23.8**, **17.9**, 14.7, 13.0, 11.5,

		10.2
P03	125.0, **43.5**, 26.3, 21.3, 17.2	200.0, **37.0**, **23.8**, **17.9**, 14.7, 13.0, 11.5, 10.2
P04	125.0, **43.5**, 26.3, 21.3, 17.2, 14.5, 12.7, 11.2, 10.1	142.9, 58.8, **34.5**, **23.8**, **17.9**, 14.7, 13.0, 11.5, 10.2
P05	125.0, **43.5**, 26.3, 21.3, 17.2	111.1, **50.0**, 34.5, **23.8**, **17.9**, 14.7, 12.8, 11.4, 10.2
P06	125.0, **43.5**, 26.3, 21.3, 17.2	125.0, **40.0**, **23.8**, **17.9**
P07	125.0, **43.5**, 26.3, 21.3, 17.2, 14.5, 12.7, 11.2, 10.1	200.0, 166.7, 62.5, **35.7**, **23.8**, **17.9**, 14.7, 13.0, 11.5, 10.2
P08	125.0, **43.5**, 26.3, 21.3, 17.2	125.0, **43.5**, 21.3, 17.2
P09	125.0, **43.5**, 26.3, 21.3, 17.2	200.0, **37.0**, **23.8**, **17.9**, 14.7, 13.0, 11.5, 10.2

6 Conclusions

The combination of spectral analysis and GIS techniques has been proved to be valid for assessing the theoretical past insolation inputs in many locations of the Sierra Nevada area (Granada province, southern Spain). The noise introduced by the decrease of direct Sunlight or terrain elevation noise, can modify the spectral signal of insolation time series derived from astronomical models. These set of techniques could be improved by further integration of spectral analysis and GIS techniques, and they could help in the process of past climate reconstructions of local areas in which a more detailed analysis is needed.

Acknowledgement

Funding for this research was provided by Projects CGL2015-71510-R and CGL2015-66835-P (Secretaría de Estado de I+D+I, Spain), Research Group RNM-178 (Junta de Andalucía), and Scientific Excellence Unit UCE-2016-05 (Universidad de Granada).

References

1. Milankovitch, M.: Kanon der Erdbestrahlung und seine Anwendung aufdas Eiszeitenproblem, 633 pp., Ed. Sp. Acad. Royale Serbe, Belgrade (English translation: Canon of Insolation and Ice Age Problem, Israel program for Scientific Translation; published for the U.S. Department of Commerce and the National Science Foundation), 1941.

2. Hays, J.D., Imbrie, J. and Shackleton, N.J.: Variations in the Earth's orbit: pacemaker of the ice ages. Science 194(4270), 1121-1132 (1976).
3. Maslin, M.: In retrospect: Forty years of linking orbits to ice ages. Nature 540(7632), 208 (2016).
4. Rodríguez-Tovar, F.J.: Orbital climate cycles in the fossil record: From semidiurnal to million-year biotic responses. Annual review of Earth and Planetary Sciences 42, 69–102 (2014).
5. Berger, A. L.: Long-term variations of daily insolation and Quaternary climatic changes. J. Atmos. Sci. 35, 2362–2367 (1978).
6. Berger, A. L.: A Simple Algorithm to Compute Long Term Variations of Daily Insolation, Institut D'Astronomie et de Géophysique, Université Catholique de Louvain, Louvain-la Neuve 18, (1978).
7. Laskar, J., Robutel, P., Joutel, F., Gastineau, M., Correia, A. C. M., and Levrard, B.: A Long-term numerical solution for the insolation quantities of the Earth. Astronomy & Astrophysics 428, 261–285 (2004).
8. Laskar, J., Fienga, A., Gastineau, M. and Manche, H.: La2010: A new orbital solution for the long-term motion of the Earth. Astronomy & Astrophysics 532, p.A89 (2011).
9. Stellarium Astronomy Software, https://stellarium.org/Stellarium.
10. R package 'insol', http://www.meteoexploration.com/R/insol/.
11. NOAA solar calculations tools, http://www.esrl.noaa.gov/gmd/grad/solcalc/calcdetails.html.
12. Kostadinov, T.S. and Gilb, R.: Earth Orbit v2. 1: a 3-D visualization and analysis model of Earth's orbit, Milankovitch cycles and insolation. Geoscientific Model Development, 7(3), 1051-1068 (2014).
13. R package 'palinsol', https://bitbucket.org/mcrucifix/insol.
14. Pardo-Igúzquiza, E., Rodríguez-Tovar, F.J.: Spectral and cross-spectral analysis of uneven time series with the smoothed Lomb–Scargle periodogram and Monte Carlo evaluation of statistical significance. Computers & Geosciences 49, 207–216 (2012).
15. "Instituto Geográfico Nacional" (IGN), www.ign.es.

Linking high-resolution marine data sets and the field of time series analysis – The long-term observational records from Helgoland and Sylt (North Sea)

Mirco Scharfe
Alfred-Wegener-Institut Helmholtz Zentrum für Polar- und Meeresforschung (Germany)

Abstract

Long-term observations are key to the detection and understanding of changes in the marine environment. The observations at the coastal stations Helgoland and Sylt, starting in year 1962 and year 1973, represent a unique combination of time series length, temporal resolution, and level of detail (range of parameters). Some of these time series, as water temperature at Helgoland Roads, cover around 13500 daily measurements in the past 57 years. Our continuous monitoring documents climatic change, such as the stronger than global warming trend at Helgoland Roads (0.034°C/yr, 1962-2018), and clearly shows the strong trends in nutrient loadings in the past decades. Moreover, time series of hydrographic parameters, nutrients and food-web components serve to develop new hypotheses, as a basis for experimental and model approaches to investigate anthropogenic and naturally driven long-term changes in the marine ecosystem. Our time series provide excellent properties to gain new knowledge by interdisciplinary approaches, e.g. by linking to approaches of non-linear time series analysis as well as approaches to derive causality from observations alone. I will show an overview of the different data series, including their main statistical features and will provide examples on how these long-term observations have contributed to the analysis of change in the North Sea. I will discuss approaches to further develop the explanatory power of the data records, e.g. by using time series techniques to derive new information content inherent to the time series. One important goal related to this is to gain more knowledge about the past and future response of the North Sea system to climate variability. Such findings will contribute to the improvement of information base for scientific use and the decision-making by society, politics, and authorities.

The Prediction Analysis of Zero Inflated Poisson Autoregression Model for the Number of Claims in General Insurance

Utriweni Mukhaiyar, Adilan Widyawan Mahdiyasa, Sapto Wahyu Indratno, and Maudy Gabrielle Meischke

Statistics Research Group, Faculty of Mathematics and Natural Sciences,
Institut Teknologi Bandung,
St. Ganesha 10 Bandung, 40132, West Java, Indonesia
{utriweni,adilan,sapto}@math.itb.ac.id
maudymeischke@ymail.com
http://www.math.itb.ac.id

Abstract. The number of claims happen in a fixed interval time, is highly possible to be a rare event. It make the series data has many zeros frequency, and the variance data is much higher than its mean, which called as over dispersion. Commonly, to model the probability distribution of frequency data, the poisson distribution is favorable. However for over-dispersed model, it no longer appropriate. The Zero Inflated Poisson (ZIP) Autoregression be the strong candidate to solve it. This model offer prediction of upcoming count data through its probability distribution. Here, this prediction method is equipped with the analysis of cumulative distribution function behaviours which assumed to follow beta distribution. Through this approach, the at most upcoming count data can be predicted as a single number instead of its probabilty distribution. For case study, the number of general insurance happen in Jakarta City is used.

Keywords: count data, time series, overdispersion, zero inflated poisson, prediction

Very Short Term Time-Series Forecasting of Solar Irradiance Without Exogenous Inputs*

Christian A. Hans and Elin Klages**

Control Systems Group, Technische Universität Berlin, Germany

Abstract. This paper compares different forecasting methods and models to predict average values of solar irradiance with a sampling time of 15 min over a prediction horizon of up to 3 h. The methods considered only require historic solar irradiance values, the current time and geographical location, i.e., no exogenous inputs are used. Nearest neighbor regression (NNR) and autoregressive integrated moving average (ARIMA) models are tested using different hyperparameters, e.g., the number of lags, or the size of the training data set, and data from different locations and seasons. The hyperparameters and their effect on the forecast quality are analyzed to identify properties which are likely to lead to good forecasts. Using these properties, a reduced search space is derived to identify good forecasting models much faster.

1 Introduction

In the last years, the capacity of globally installed photovoltaic generators has continuously increased [1]. With this growth, the intermittent nature of their infeed has become a major challenge in the operation of electric grids [2]. One important way to address this challenge is to use accurate forecasts to predict the infeed of photovoltaic power plants with a high temporal resolution [3].

Various studies on short term forecasting of photovoltaic power plant infeed [4–8] and solar irradiance [9–11] have been published. Autoregressive integrated moving average (ARIMA) models have been widely adopted in this context [4, 7, 9]. In the last decades forecasting methods that use techniques from artificial intelligence have become more prominent. Most of them use artificial neural networks [4, 8–11]. Others employ support vector [6, 8] and nearest neighbor regression [4, 5].

Despite additional effort that comes with the use of exogenous inputs, e.g., cloud cover or air temperature, only in [4, 7, 10] forecasts without exogenous inputs are considered. Moreover, in most publications [4–7, 10, 11] the forecasting models are obtained using only one data set. This makes it hard to draw general conclusions from the analysis that can be transferred to other locations or seasons of the year. Furthermore, in [4–6, 9–11] the model selection process is not explicitly discussed. Consequently, even though high forecast accuracies could be achieved for single data sets, due to the missing description of the selection processes, the findings cannot be directly used for different data sets. For example, in [11] the authors state that they used a search and in [9] that they tried different model structures to choose an artificial neural networks. However, they did not provide information on the search space. In [6], a search space for support vector regression forecasts is provided but no information on the selection criterion or the final model structure and parameters is given. The structure of an artificial neural networks forecasting model is examined in [10] using a sensitivity analysis. Unfortunately, the publication does not contain sufficient details to replicate the analysis for different data. To the knowledge of the authors only in [5] a search space is provided and the results are analyzed to gain information about suitable model structures. However, the analysis is only performed on data from one location. This makes it hard to draw conclusions that allow identify a reduced search space and in practice often leads to an exhaustive search for suitable models.

In this paper we derive a significantly reduced search space that exhibits a high probability of finding an accurate forecasting model. As potential forecasting methods, ARIMA and nearest neighbor regression (NNR) were considered due to their wide adoption. Furthermore, their hyperparameters, e.g., the number of autoregressive lags or neighbors, are mostly discrete which allows

* This work was partially supported by the German Federal Ministry for Economic Affairs and Energy (BMWi), Project No. 0324024A.

** Currently at the Institute of Modelling and Computation, Technische Universität Hamburg, Germany.

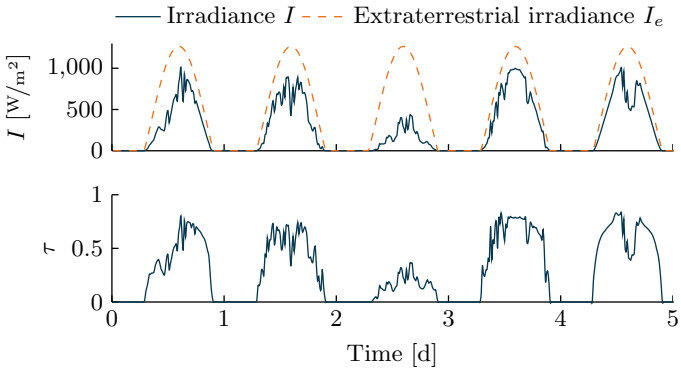


Fig. 1: Global horizontal irradiance, extraterrestrial irradiance and transmissivity over a duration of 5 d. The irradiance was taken from [14], the extraterrestrial irradiance was estimated using [15].

to form a discrete search space where points in a certain range can be explored. For each point, i.e., for each set of hyperparameters in this search space, forecasting models are trained and tested using data from different locations and seasons. Based on the resulting forecast accuracies, conclusions for a reduced search space are drawn. These include the choice of NNR over ARIMA, the number of training data points and autoregressive lags as well as handling of night data and use of transmissivity instead of irradiance. The selection process is explained in much detail allowing others to perform a similar search on different data.

For simplicity and robustness, this work focuses on forecasts of solar irradiance without exogenous inputs using only historic irradiance data, time and location. Motivated by the smallest intraday interval of energy trading in Germany a 15 min prediction step is chosen [12]. To cover a full charge or discharge of a medium size storage unit, a prediction horizon of 3 h is considered.

The remainder of this paper is organized as follows. Solar irradiance and data preprocessing are discussed in Section 2. In Section 3, basics on time-series forecasting are discussed. The forecasting methods considered in this work are presented in Section 4. Then, the model selection procedure is illustrated in Section 5. Finally, in Section 6 the results of the hyperparameters search are analyzed.

1.1 Preliminaries

Throughout this paper, real numbers are denoted by $\mathbb{R}$, nonnegative real numbers by $\mathbb{R}_0^+$, positive real numbers by $\mathbb{R}^+$, natural numbers by $\mathbb{N}$ and nonnegative integers by $\mathbb{N}_0$. The transpose of a matrix x is x^T and the Euclidean norm of a vector x is $\|x\|_2$. The sum over all elements of a set $\mathbb{K} \subset \mathbb{N}$ where each element $i \in \mathbb{K}$ is taken exactly once is denoted $\sum_{i \in \mathbb{K}} x_i$.

2 Solar Irradiance

The sunlight reaching the outer earth's atmosphere is called extraterrestrial solar radiation. It can be estimated from the energy emitted by and the position of the sun. The sunlight reaching a horizontal plane on earth per unit area at time t, is called global horizontal irradiance $I_t \in \mathbb{R}_0^+$. It includes direct normal irradiance, which originates directly from the sun, and diffuse irradiance, which includes scattered and ground reflected components, [13].

For forecasting, it can be beneficial to normalize the solar irradiance I_t by the extraterrestrial irradiance $I_{\mathrm{e},t} \in \mathbb{R}_0^+$ (see Fig. 1). This yields to transmissivity defined as $\tau_t = {}^{I_t}/_{I_{\mathrm{e},t}}$ [8]. In this work, the solar position algorithm from [15] is used to estimate the extraterrestrial irradiance $I_\mathrm{e}(t)$. It only requires the position of the respective surface to calculate the zenith angle $\zeta_t \in [0, 2\pi)$, i.e., the incidence angle of the sun light on a horizontal plane on the earth's surface [16]. Using ζ_t, the extraterrestrial irradiance can be determined by $I_{\mathrm{e},t} = \epsilon_t I_\mathrm{s} \cos(\zeta_t)$, where ϵ_t is a correction factor and $I_\mathrm{s} = 1360.8\,\mathrm{W/m^2}$ is the solar constant.

3 Time-Series Forecasting

A time series is a collection of $n \in \mathbb{N}$ chronologically ordered observations $y_1, y_2, \ldots, y_n$. In this collection, every element y_t refers to an observation performed at a time instant $t = 1, \ldots, n$. In this work, we focus on univariate forecasts, i.e., forecasts using only historic irradiance to forecast future irradiance. Thus, all elements $y_t \in \mathbb{R}$ for $t = 1, \ldots, n$ are real-valued scalars.

Broadly speaking, a forecasting method is a procedure to estimate future values $\hat{y}_{t+j|t}$ of a time series based on past values $Y_t \in \mathbb{R}^n$, i.e., $\hat{y}_{t+j|t} = f(Y_t)$. Here, $\hat{y}_{t+j|t}$ refers to the value at prediction step $j \in \mathbb{N}$ with last known value at time $t \in \mathbb{N}$. In case of univariate forecasting, Y_t only includes the present and past values of the time series being forecast, i.e., $Y_t = [y_t \; y_{t-1} \; \cdots \; y_{t-n+1}]^T$.

The vector Y_t can also be based on non-consecutive lags, e.g., to represent seasonal behavior. An example for such a vector is $Y_t = [y_t \; y_{t-s+1} \; y_{t-s}]^T$, where $s \in \mathbb{N}$ is the length of the season, e.g., one day. In this example, the present observation, y_t, the observation from the last day, y_{t-s+1}, at the same time as the predicted value, $\hat{y}_{t+1}$, and from the current time minus 24 h, y_{t-s}, are included.

3.1 Multi-Step Forecasts

To create multi-step ahead forecasts, a recursive strategy can be applied [17, Section 4.]. The forecasting model is repeatedly used for one step forecasts, where the elements of Y_t are adapted at each step until prediction step $J \in \mathbb{N}$ is reached. For example, if Y_t includes the last three data points, $Y_t = [y_t \; y_{t-1} \; y_{t-2}]^T$ then the procedure for $J = 12$ prediction steps would be

$$\hat{y}_{t+1|t} = f(Y_t) = f([y_t \quad y_{t-1} \quad y_{t-2}]^T), \tag{1a}$$

$$\hat{y}_{t+2|t} = f(\widehat{Y}_{t+1|t}) = f([\hat{y}_{t+1|t} \quad y_t \quad y_{t-1}]^T), \tag{1b}$$

$$\vdots$$

$$\hat{y}_{t+12|t} = f(\widehat{Y}_{t+11|t}) = f([\hat{y}_{t+11|t} \quad \hat{y}_{t+10|t} \quad \hat{y}_{t+9|t}]^T). \tag{1c}$$

For seasonal models, the vector $\widehat{Y}_{t+j|t} \in \mathbb{R}^n$ is equally adapted at each step in a similar manner.

3.2 Evaluation of Forecast Accuracy

Handling of Night Data. Forecasts of data points during the night, i.e., of zero irradiance values, were not included in the evaluation of the forecast accuracy. Therefore, observations with a zenith angle $\zeta_t \leq 90.83°$ were excluded based on [18]. The angle ζ_t was taken from the data sets if available. Otherwise it was estimated using [15].

Root Mean Square Error (RMSE). For each prediction step $j = 1, \ldots, J$ over prediction horizon J, the RMSE is calculated to analyze the error of each step separately. Assuming that $m \in \mathbb{N}$ forecasts were performed, the RMSE of prediction step j is

$$\mathrm{RMSE}_j = \sqrt{\tfrac{1}{m} \textstyle\sum_{t=1}^{m} \left(\hat{y}_{t+j|t} - y_{t+j}\right)^2}. \tag{2}$$

4 Forecasting Methods

This section presents the forecasting methods employed in this work. First, the persistence model used as reference is discussed. Then, ARIMA models and nearest neighbor regression are illustrated.

4.1 Persistence Models

The basic idea of the persistence model is that future values are assumed to be equal to known past values. In this work, the simple model, $\hat{y}_{t+1|t} = y_t$ is used. It is also possible to consider a seasonal persistence model using, e.g., data from the previous day, week or year. This can be described by $\hat{y}_{t+j|t} = y_{t+j-s}$ where s is the seasonal period.

4.2 Autoregressive Integrated Moving Average (ARIMA) Models

ARIMA models are widely used for time-series forecasting. Future values are estimated using a linear combination of previously observed values and forecasting errors. Also, differencing can be applied to obtain stationary data. An ARIMA model can be described by [19]

$$\Phi(B)\nabla^d y_k = \theta_0 + \Theta(B)e_t, \tag{3}$$

where B is the backwards shift operator with $B^m y_t = y_{t-m}$, and ∇ is the backwards difference with $\nabla^d y_t = (y_t - y_{t-1})^d$, $d \in \mathbb{N}_0$. Furthermore, $\Phi(B) = 1 - \phi_1 B^1 - \ldots - \phi_p B^p$ with $\phi_1, \ldots, \phi_p \in \mathbb{R}$, $p \in \mathbb{N}_0$ is the autoregressive part, and $\Theta(B) = 1 - \theta_1 B^1 - \ldots - \theta_q B^q$ with $\theta_1, \ldots, \theta_q \in \mathbb{R}$, $q \in \mathbb{N}_0$ the moving average part. Moreover, e_t is the difference between measured value and forecast.

To fit a model to data that exhibits seasonality, ARIMA models can be extended by seasonality of period s. These, so called seasonal ARIMA (SARIMA) models, can be described by

$$\Phi(B)\mathbf{\Phi}(B)\nabla^d \nabla_s^D y_t = \theta_0 + \Theta(B)\mathbf{\Theta}(B)e_t, \tag{4}$$

where $\mathbf{\Phi}(B) = 1 - \Phi_1 B^s - \ldots - \Phi_P B^{P\,s}$ with $\Phi_1, \ldots, \Phi_P \in \mathbb{R}$, $P \in \mathbb{N}_0$ is the seasonal autoregressive part and $\mathbf{\Theta}(B) = 1 - \Theta_1 B^s - \ldots - \Theta_Q B^{Q\,s}$ with $\Theta_1, \ldots, \Theta_Q \in \mathbb{R}$, $Q \in \mathbb{N}_0$ the seasonal moving average part. Moreover, $\nabla_s^D y_t = (y_t - y_{t-s})^D$, $D \in \mathbb{N}_0$ represents seasonal differencing.

For the implementation of the ARIMA models, the econometrics toolbox in MATLAB 2015b was used. Within the toolbox, maximum likelihood estimation is used to find the model parameters.

4.3 Nearest Neighbor Regression (NNR)

In what follows, based on [20] we introduce NNR. In NNR the pattern of historic data at time instant t, $Y_t \in \mathbb{R}^n$, is compared to previous patterns $Y_i \in \mathbb{R}^n$, $i = 0, \ldots, N-1$ in the reference sample $\mathbb{D} = \{(Y_0, y_1), (Y_1, y_2), \ldots, (Y_{N-1}, y_N)\}$, $N \in \mathbb{N}$ to forecast $\hat{y}_t$.

Although NNR allows to use arbitrary elements in Y_i, e.g., $Y_i = [y_{i-13}\ y_{i-111}\ y_{i-2678}]^T$, we follow the notion of autoregressive and seasonal autoregressive lags of (seasonal) ARIMA models to enable a comparison of both methods. For NNR this means that according to the number autoregressive lags p the vectors Y_i in $\mathbb{D}$ are formed. Thus, analog to ARIMA models the entries of the NNR reference sample with $p = 3$ autoregressive lags has the form $Y_i = [y_i\ y_{i-1}\ y_{i-2}]^T$.

The same holds for the seasonal autoregressive lags. In a similar fashion as in (4), the seasonal autoregressive part is multiplied with the autoregressive part. Consequently, a reference sample with $p = 3$ autoregressive lags, $P = 2$ seasonal autoregressive lags and a season of $s = 10$ results in

$$Y_i = [y_i \quad y_{i-1} \quad y_{i-2} \mid y_{i-9} \quad y_{i-10} \quad y_{i-11} \quad y_{i-12} \mid y_{i-19} \quad y_{i-20} \quad y_{i-21} \quad y_{i-22}]^T.$$

Thus, as in (4), the elements of the previous seasons that are associated with $\hat{y}_{i+1|i}$ (here y_{i-9} and y_{i-19}) are included in Y_i. This is due to the zero order term, i.e., the 1, in $\Phi(B)$ and $\mathbf{\Phi}(B)$.

A simple forecasting model can be obtained by combining the $k \in \mathbb{N}$ elements in $\mathbb{D}$ with the smallest distance to Y_t. Using the set of k nearest neighbors of Y_t, $\mathbb{K}(Y_t) \subseteq \mathbb{D}$, this can be written as

$$\hat{y}_{t+1|t} = f(Y_t, \mathbb{D}) = {}^1\!/_k \sum_{Y_i \in \mathbb{K}(Y_t)} y_{i+1}. \tag{5}$$

Note that the number of neighbors k can be fixed or determined by a maximum distance $\varepsilon \in \mathbb{R}^+$. With distance $d(Y_i, Y_t)$, the set of neighbors closer than ε to Y_t is $\mathbb{K}(Y_t) = \{Y_i \in \mathbb{D} \mid d(Y_i, Y_t) \leq \varepsilon\}$. Note that in this work, the Euclidean norm is used as distance, i.e., $d(Y_i, Y_t) = \|Y_i - Y_t\|_2$.

The model (5), can be modified using a weighted average. In this work, weights inverse to the distance between x_t and the neighbor x_i, have been considered, i.e.,

$$\hat{y}_{t+1|t} = f(Y_t, \mathbb{D}) = \begin{cases} {}^1\!/_k \sum_{Y_i \in \mathbb{K}(Y_t)} \frac{1}{d(Y_i, Y_t)} y_{i+1}, & \text{if } d(Y_i, Y_t) \neq 0\ \forall Y_i \in \mathbb{K}(Y_t), \\ y_{l+1}, & \text{if } \exists Y_l \in \mathbb{K}(Y_t) \text{ with } d(Y_t, Y_l) = 0. \end{cases} \tag{6}$$

Although, (5) requires no training, the model and the selected reference sample will be referred to as trained model for simplicity. For implementation, the Statistical Learning Toolbox for MATLAB [21] was used. Note that in this work, approximated nearest neighbors which is faster than the exact nearest neighbors algorithm is used as no significant difference in the forecast accuracy between the two could be observed.

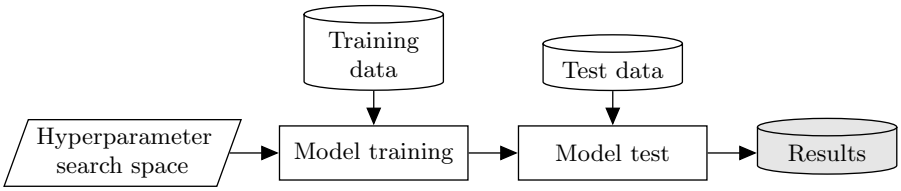


Fig. 2: Scheme used to obtain forecast accuracy data for different models.

5 Hyperparameter Search

We aim to compare the forecast accuracy for different forecasting model structures and different sets of historic data. Therefore, an exhaustive search was performed to evaluate what models approximate the test data best. In our search, different model structures are trained using different sets of historic data. To identify model structure and historic data, hyperparameters are used. Moving along Fig. 2, in this section first training and test data are discussed. Then, the hyperparameter search space is posed. Finally, training and test of the models is illustrated.

5.1 Training and Test Data

For training and test data sets [14, 22, 23] were used. From each data set three different seasons were selected to include different climatic situations in the hyperparameter search. For every location and every season, i.e., for 9 data sets, the observations were divided into training data (up to 2 months) and independent test data (1 week) that was not used for training. The test data was chosen to always follow directly the data used for training. As shown in Fig. 3, the nine weeks of test data include different climatic situations, e.g., a sunny week in December from the National Renewable Energy Laboratory (NREL) Clark Station, a cloudy week in March from the Atmospheric Radiation Measurement (ARM) Facility, and various weeks with sunny, foggy and cloudy days.

Note that all data sets [14, 22, 23] have a temporal resolution of 1 min. The 15 min average values $y_1, \ldots, y_N$ were derived from the original time series $y'_1, \ldots, y'_{15N}$ as $y_t = {}^1\!/_{15} \sum_{q=1}^{15} y'_{q+15(t-1)}$.

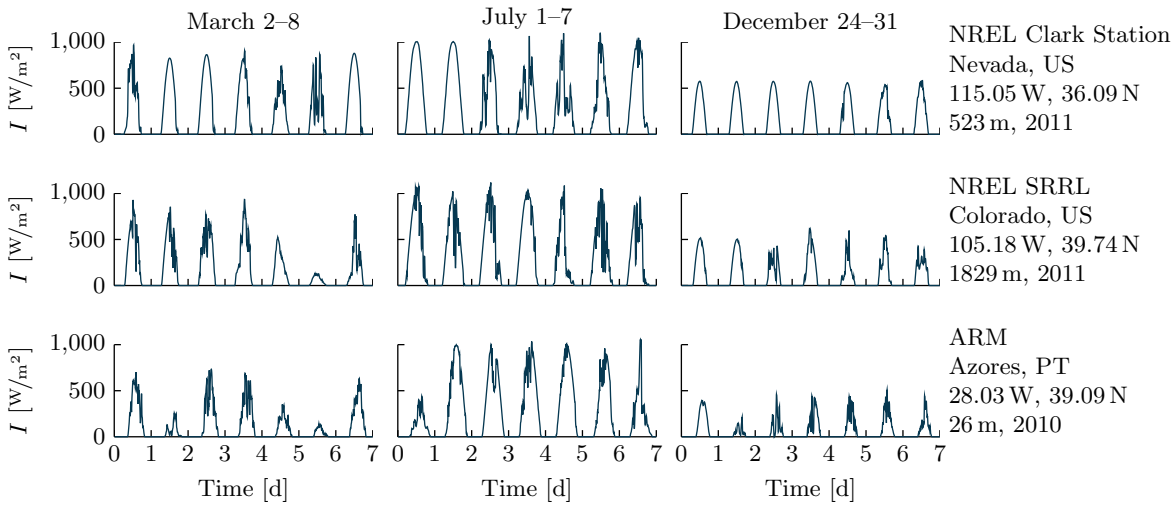


Fig. 3: Measured solar irradiance data used to test the trained models.

Table 1: Sets of hyperparameters for different forecasting methods considered in search.

(a) ARIMA models.

Autoregressive lags p	$0, 1, \ldots, 10$
Moving average lags q	$0, 1, \ldots, 10$
Differencing d	$0, 1, 2$

(b) NNR.

Autoregressive lags p	$1, 2, \ldots, 20$
Weight	uniform, inverse to distance
Number of neighbors k	$1, 2, \ldots, 20$
Threshold[3] ε	$0.01, 0.05, 0.1, 0.5, 1$

(c) SARIMA models.

Autoregressive lags p	$0, 1, 3$
Moving average lags q	$0, 1, 3$
Seasonal autoregressive lags P	$0, 1, 2, 3$
Seasonal moving average lags Q	$0, 1, 2, 3$
Differencing d	$0, 1$
Seasonal differencing D	$0, 1$

(d) SNNR.

Autoregressive lags p	$1, 2, \ldots, 11$
Seasonal autoregressive lags P	$1, 2, \ldots, 7$
Weight	uniform, inverse to distance
Number of neighbors k	$1, 2, \ldots, 20$
Threshold[3] ε	$0.01, 0.05, 0.1, 0.5, 1$

5.2 Hyperparameters

Data Hyperparameters. The hyperparameters that concern the training data are independent of the forecasting method and the model structure. We considered modifications of training data by 1. pre- and postprocessing, 2. handling of night data ans 3. number of data points used.

1. Regarding the pre- and postprocessing, two approaches were considered. (a) The time series of irradiance, I_t, is directly used to train the models without further preprocessing. (b) Transmissivity $\tau_t = {}^{I_t}/_{I_{\mathrm{e},t}}$ is derived from I_t and extraterrestrial irradiance $I_{\mathrm{e},t}$. Then, the models are trained to forecast $\hat{\tau}_{t+j|t}$. Using $\hat{\tau}_{t+j|t}$ the irradiance forecast is then calculated via $\hat{I}_{t+j|t} = I_{\mathrm{e},t}\,\hat{\tau}_{t+j|t}$.

2. Three cases related to the handling of data points at night were considered: (a) include all data points from day and night, (b) only include data points between 5 am and 8 pm local time, and (c) only include data points between sunrise and sunset, i.e., observations with $\zeta_t > 90.83°$ (see Section 3.2). As it can be cumbersome to implement models with seasonality for a varying number of data points per day, case (3) was not considered for seasonal models.

3. Regarding the number of training data points, 1, 3, 7, 14, 30, and 60 days of data were considered. For seasonal models, the data was chosen to include at least 14 days.

Model Hyperparameters. For every forecasting method, the model structure can be uniquely described using model specific hyperparameters. Note the difference to model parameters of a trained model. For example, the number of autoregressive lags p is a hyperparameter. The coefficients of the autoregressive part, $\phi_1, \ldots, \phi_p$, of a trained ARIMA model are model parameters. The subsets of hyperparameters considered in the search for are shown in Table 1. Note that in Tables 1(b) and 1(d) either the number of neighbors k or the maximal distance to the neighbors ε is used. Further note that the season s was chosen to be 1 d for SARIMA and seasonal NNR (SNNR).

5.3 Model Training

For each combination of hyperparameters described in Section 5.2, we trained nine models, i.e., one for each location and each season (see Section 5.1). This lead to more than 250,000 trained forecasting models with different hyperparameters. Note that some combinations of hyperparameters did not result in suitable models. Especially for ARIMA, no stable model could be derived for certain hyperparameters. Furthermore, for SARIMA 3.3 % of the hyperparameters were excluded, mostly due to very long training procedures that rendered them unsuitable for practical use.

[3] Multiplied by $1360.8\,{}^{\mathrm{W}}/_{\mathrm{m}^2}$ for irradiance.

5.4 Model Test

In the model test, the performance of the trained models was evaluated. Therefore, every trained model was tested by forecasting irradiance of the unknown test data (see Fig. 3) that followed the training data. Due to zero irradiance, data points during the night were excluded from the evaluation based on the zenith angle ζ_t (see Section 3.2). For each of the data point in the test data, a 12 step ahead, i.e., 3 h, prediction was obtained. Using the forecast values and the test data, the RMSE_j (see Section 3.2) was then calculated for $j = 1, \ldots, 12$ and stored. In the next section, we will analyze the resulting RMSE_j for every combination of hyperparameters, location and season to identify models that are likely to provide good forecasts.

6 Analysis

In this section, we analyze the results of the hyperparameter search in Section 5. Our goal is to draw conclusions that help to reduce the space of future hyperparameter searches. More precisely, we aim to find forecasting methods and ranges of hyperparameters that increase the probability of finding a suitable model for unseen data. Before starting the analysis, some remarks are posed.

Remark 1. As reference, we use the persistence model on transmissivity as it outperformed the persistence model on irradiance for all test data sets. In the reference model historic transmissivity is derived using $\tau_t = {}^{I_t}/_{I_{\mathrm{e},t}}$. Then a persistence forecast is performed, i.e., we set $\hat{\tau}_{t+j|t} = \tau_t$ for $j = 1, \ldots, J$. Finally, the irradiance forecast is determined as $\hat{I}_{t+j|t} = \hat{\tau}_{t+j|t}\, I_{\mathrm{e},t+j}$ for $j = 1, \ldots, J$.

Remark 2. The style of the box plots used throughout this work is illustrated in in Fig. 4. Here, m marks the median. The box around the median contains all data from the 25th (q_1) to the 75th (q_3) percentile. The left whisker marks the lowest occurring value within $q_1 - 1.5(q_3 - q_1)$ and the right whisker marks the highest occurring value within $q_3 + 1.5(q_3 - q_1)$. Due to numerous outliers, only every 100th outlier is shown. To assure that the forecast with the highest accuracy is considered the lowest outlier is always included.

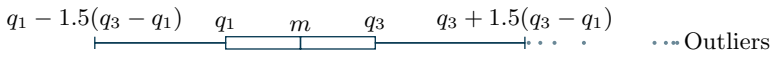


Fig. 4: Box plot with median m, quartiles q_1 and q_3, whiskers and outliers.

Remark 3. Most distributions of the forecast accuracy discussed in this section are similar for the different test data sets shown in Fig. 3. Therefore, most distributions were combined in one plot that includes the values of all locations.

Remark 4. Throughout this paper, the forecast at prediction steps $j = 1, 4, 12$ are analyzed to approximate the evolution of the forecast accuracy over prediction horizon $J = 12$. For the 15 min sampling time considered, they correspond to a forecast 15 min, 1 h and 3 h ahead.

6.1 Forecasting Method

We first compare the forecasting methods ARIMA and NNR along with the persistence model on transmissivity that acts as a reference (see Remark 1). In Fig. 5, box plots illustrating the distributions of forecast accuracies of all models included in the search are shown.

The box plots in Fig. 5 show that for predictions 15 min ahead, almost no model outperforms the persistence model. For a prediction step of 1 h some models outperform the persistence forecaster. Furthermore, the RMSE of forecasts performed with NNR varies less than the RMSE of ARIMA forecasts. This indicates a higher probability of finding a good forecasting model using NNR. For

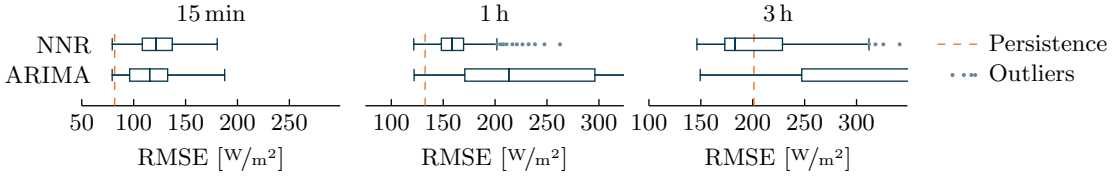


Fig. 5: RMSE for 15 min, 1 h and 3 h ahead predictions with ARIMA and NNR models.

predictions of 3 h ahead, the potential improvement over the persistence model is significant for all methods. As already observed for 1 h ahead predictions, the RMSE of NNR varies less. Additionally, the 25th percentile is remarkably lower for NNR. The highest forecast accuracy achieved is very alike for NNR and ARIMA for all prediction steps. Still, NNR achieves much lower medians and lower 25th and 75th percentiles. Consequently, the probability of finding a suitable NNR model is much higher. Therefore, ARIMA and SARIMA models will be disregarded in the next steps of the analysis. Consequently, in what follows we only investigate how the search spaces for NNR and SNNR can be reduced to increase the probability of finding an accurate forecasting model.

6.2 Preprocessing of NNR and SNNR

As stated in Section 5.2, models using irradiance or the normalized transmissivity values were considered in the search. In Fig. 6 the forecast accuracy of both approaches is shown. It can be observed that the box plots for 15 min forecasts are very alike. Also, little difference can be seen for 1 h ahead predictions. However, for 3 h ahead predictions the transmissivity based forecasts often outperform the irradiance based forecasts. The only exception from this observation could be found for data from the NREL Clark Station during March 2–8 (see Fig. 7), where the best models use irradiance data. Still the lower whiskers and the medians in this case are very alike.

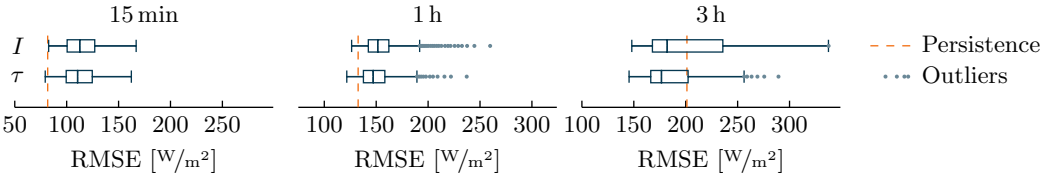


Fig. 6: RMSE of NNR and SNNR using different preprocessing for all locations and seasons.

The box plots indicate that using transmissivity yields a higher probability of finding an accurate model. Although for data set [22] a better irradiance model for the 3 h prediction could be found, the overall distribution in this case shows no significant additional difference to Fig. 6. As the single best model is hard to find, we are confident that using transmissivity yields a higher probability of finding a good model. Therefore, irradiance forecasts are excluded from the reduced search space.

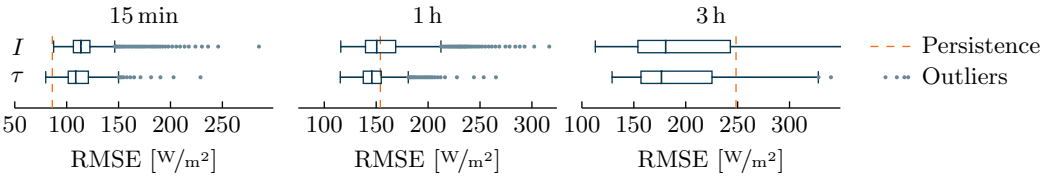


Fig. 7: RMSE of NNR and SNNR using different preprocessing at NREL Clark Station, March 2–8.

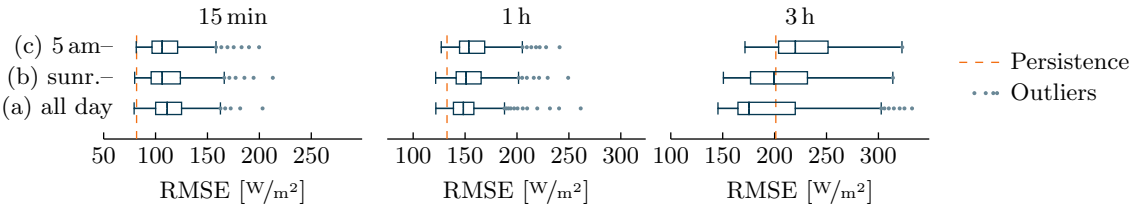


Fig. 8: RMSE of NNR and SNNR for different handling of night data: (a) entire day, (b) sunrise to sunset and (c) 5 am to 8 pm.

6.3 Handling of Night Data of NNR and SNNR

As stated in Section 5.4, data points during the night were not included in the evaluation of the forecast accuracy. However, in some cases they were used in the data of the reference sample to allow for smooth transitions between two days. As stated in Section 5.1, we considered training data that (a) includes the data of the entire day (and night), (b) only includes data points from sunrise to sunset and (c) only includes data points from 5 am to 8 pm.

In Fig. 8, the forecast accuracy for different handling of night data is shown. It can be observed that the forecast accuracy is similar for the 15 min forecast. For forecasts 1 h ahead, the difference between the approaches remains small. However, for forecasts 3 h ahead, using all data points for training, i.e., including night values, outperforms the other approaches. Therefore, all data points including night data are considered in the reduced search space.

6.4 Size of the Reference Sample of NNR and SNNR

The reference sample is chosen such that it includes that last values of historic data up to the most recent data point. A large reference sample, i.e., a set $\mathbb{D}$ with a high number of elements, therefore includes data that reaches further into that past than a small reference sample.

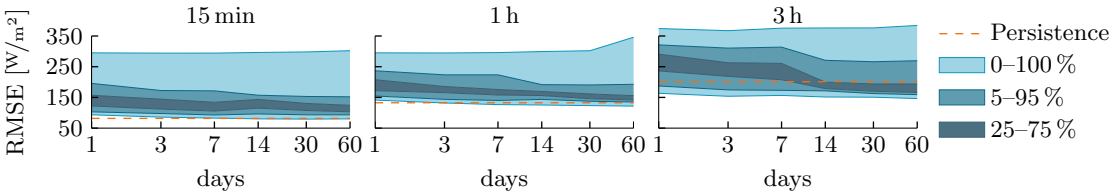


Fig. 9: RMSE of NNR and SNNR using different sizes of the reference sample.

In Fig. 9, the forecast accuracy using a reference sample of 1 d, 3 d, and 7 d as well as two weeks (14 d), three weeks (21 d), one month (30 d) and two months (60 d) is shown. The plots for all prediction steps shows a small decrease of the 25th and the 75th percentile as well as the lowest RMSE for an increasing number of elements in the reference sample. Considering the largest RMSE of each prediction step, the inverse effect can be observed. In general little difference can be observed in the forecast accuracy, when varying the size of the reference sample. However, as the forecast accuracy increases using more data, a reference sample of 60 d is used in the reduced search space.

6.5 Autoregressive (AR) Lags of NNR and SNNR

This section focuses on the AR lags, i.e., the data points of the historic data used in the elements of the reference sample. We differentiate between AR and seasonal AR lags. The former refer to 15 min steps prior to the first forecast time instant. The latter refer to 24 h steps, used to include

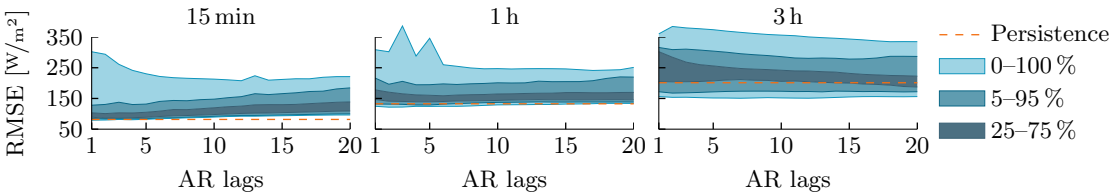


Fig. 10: RMSE of NNR using different AR lags.

the days prior to the forecast. It can be observed in Fig. 10 that for all prediction steps the models with the highest forecast accuracy has a small number of AR lags. However, no clear tendency can be identified that supports choosing a particular number of AR lags.

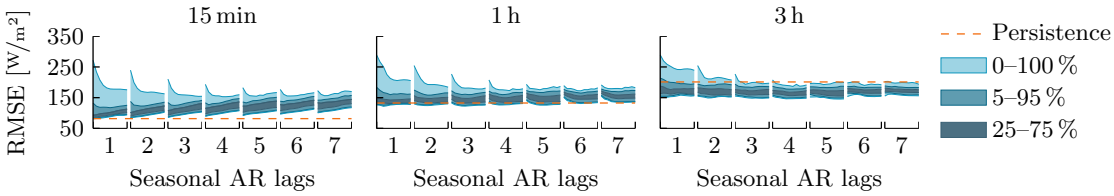


Fig. 11: RMSE of SNNR using different seasonal AR lags. For every seasonal AR lag, the non-seasonal AR lags increase from 1 on the left side of the plot to 11 on the right side of the small subplot.

Fig. 11 shows the RMSE for different numbers of seasonal AR lags. Here, for every seasonal AR lag, the number of AR lags was increased from 1 (left) to 11 (right). It can be observed that for 15 min forecasts the RMSE slightly increases with the number of seasonal AR lags. Furthermore, for every seasonal AR lag, the RMSE also increases with increasing number of non-seasonal AR lags. In contrast, for the 3 h forecasts, the RMSE slightly decreases with the number of seasonal lags. Comparing Figs. 10 and 11, shows that the variance decreases for the 1 h and 3 h forecasts with increasing number of seasonal lags. This results in a higher probability of finding a good model for a higher number of seasonal AR lags. Unfortunately, the 15 min forecasts do not show the same effect. Still, as the 15 min forecast barely outperform the persistence model it seems reasonable focus on the improvement for larger horizons and consider only seasonal models in the reduced search space.

6.6 Weights of NNR and SNNR

As stated in Section 5.2 uniform weights and weights inverse to the distance to the neighbors were considered. In the following, we analyze the effect of the distance to neighbors on the distribution of the RMSE. As can be seen in Fig. 12, the box plots hardly differ for all prediction steps. This indicates that there is no strong correlation between forecast accuracy and weights. However, because of an easier implementation, uniform weights were chosen for the reduced search space.

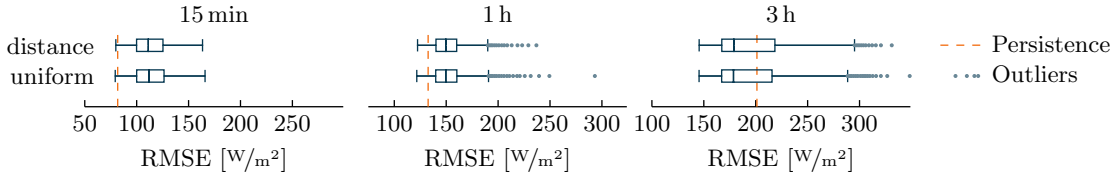


Fig. 12: RMSE using uniform weights and weights inverse to the distance of the neighbor.

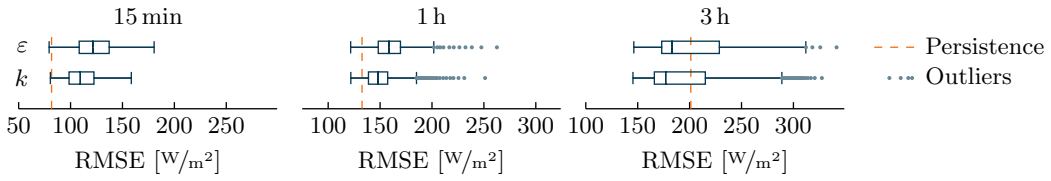


Fig. 13: RMSE using a threshold ε or a fixed number of neighbors k to define the neighborhood $\mathbb{K}$.

6.7 Definition of the Neighborhood of NNR and SNNR

As stated in Section 4.3, the neighborhood can be chosen to be a fixed number of neighbors, k, or a maximum distance, ε. The comparison in Fig. 13 shows that the highest accuracy achieved with both hyperparameters is very alike for all prediction steps. However, the 25th and 75th percentile, the upper whisker and the median are lower for models using a fixed number of neighbors k. Therefore, in the reduced search space, only models using a fixed number of neighbors k are considered.

6.8 Number of Neighbors k of NNR and SNNR

In Fig. 14, the forecast accuracy of all k nearest neighbor models included in the search are shown. It can be observed, that the forecast accuracy mostly changes using 1 to 5 neighbors. The distribution of the RMSE becomes narrower for most cases with an increasing k. Furthermore, the lowest RMSE decreases slightly for $k \geq 10$. Therefore, models using between 10 and 20 neighbors are considered the reduced search space.

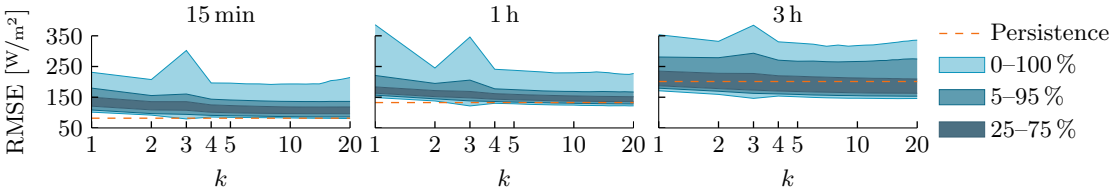


Fig. 14: RMSE of NNR and SNNR using a different number of neighbors k.

6.9 Summary

In summary, the following conclusions can be drawn from the analysis.

1. The probability of finding a sufficiently accurate model to forecast irradiance is much higher for NNR and SNNR than for ARIMA and SARIMA models (see Section 6.1).
2. Data pre- and postprocessing of NNR and SNNR models:
   (a) Use transmissivity instead of irradiance (Section 6.2).
   (b) Include night data in the reference sample (Section 6.3).
   (c) Use data from the past 60 d in the reference sample (Section 6.4).
3. Hyperparameters of NNR and SNNR models:
   (a) Seasonal models are beneficial (Section 6.5).
   (b) Favor uniform weights over weights inverse to distance (Section 6.6).
   (c) Choose the neighbors using a fixed number k instead of a maximum distance (Section 6.7).
   (d) Search for models with 10 to 20 neighbors (Section 6.8).

Based on these findings the hyperparameter search space, i.e., the number of potential models could be reduced from more than 250,000 to less than 1,000. The search space for suitable k nearest neighbor (kNN) forecasters now only includes the number of autoregressive lags with range $1, 2, \ldots, 11$, number of seasonal autoregressive lags with range $1, 2, \ldots, 7$ and number of neighbors with range $10, 11, \ldots, 20$.

7 Conclusion

A comparison of ARIMA and NNR for short term solar forecasts showed that it is more likely to find good forecasting models using NNR. Based on this finding, we derived a reduced search space of models that are likely to provide a good NNR forecasting model on unseen data.

Future work concerns an extension of the current approach to include exogenous data. Additionally, other artificial intelligence based methods, e.g., neural networks and support vector regression, are planned to be investigated.

Bibliography

[1] REN21 Secretariat: Renewables 2017 global status report. Technical report, REN21 (2017)
[2] Denholm, P., Hand, M.: Grid flexibility and storage required to achieve very high penetration of variable renewable electricity. Energy Policy **39**(3) (2011) 1817–1830
[3] West, S.R., Rowe, D., Sayeef, S., Berry, A.: Short-term irradiance forecasting using skycams: Motivation and development. Sol. Energy **110** (2014) 188–207
[4] Pedro, H.T., Coimbra, C.F.: Assessment of forecasting techniques for solar power production with no exogenous inputs. Sol. Energy **86**(7) (2012) 2017–2028
[5] Pedro, H.T., Coimbra, C.F.: Nearest-neighbor methodology for prediction of intra-hour global horizontal and direct normal irradiances. Renewable Energy **80** (2015) 770–782
[6] Shi, J., Lee, W.J., Liu, Y., Yang, Y., Wang, P.: Forecasting power output of photovoltaic systems based on weather classification and support vector machines. IEEE Trans. Ind. Appl. **48**(3) (2012) 1064–1069
[7] Bacher, P., Madsen, H., Nielsen, H.A.: Online short-term solar power forecasting. Sol. Energy **83**(10) (2009) 1772–1783
[8] Zeng, J., Qiao, W.: Short-term solar power prediction using a support vector machine. Renewable Energy **52** (2013) 118–127
[9] Reikard, G.: Predicting solar radiation at high resolutions: A comparison of time series forecasts. Sol. Energy **83**(3) (2009) 342–349
[10] Sfetsos, A., Coonick, A.: Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques. Sol. Energy **68**(2) (2000) 169–178
[11] Mellit, A., Pavan, A.M.: A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy. Sol. Energy **84**(5) (2010) 807–821
[12] EPEX SPOT SE: 15-minute intraday call auction (2016) URL: `https://www.epexspot.com/document/29113/` (accessed Apr. 24, 2018).
[13] ISO 9488:1999: Solar Energy – Vocabulary. International Organization for Standardization, Geneva, CH (1999)
[14] ARM: Climate research facility, surface meteorology system (MET). Eastern North Atlantic Facility ARM Data Archive: Oak Ridge, US (2010) (accessed Jul. 14, 2011).
[15] Reda, I., Andreas, A.: Solar position algorithm for solar radiation applications. Sol. Energy **76**(5) (2004) 577–589
[16] Quaschning, V.: Regenerative Energiesysteme. München, Hanser (2009)
[17] Bontempi, G., Taieb, S.B., Le Borgne, Y.A.: Machine learning strategies for time series forecasting. In: European Business Intelligence Summer School, Springer (2012) 62–77
[18] Meeus, J.H.: Astronomical algorithms. Willmann-Bell, Inc. (1991)
[19] Box, G.E.P., Jenkins, G.M., Reinsel, G.C.: Time Series Analysis: Forecasting and Control. Prentice- Hall, Inc. (1994)
[20] Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician **46**(3) (1992) 175–185
[21] Lin, D.: Statistical learning toolbox (2006)
[22] Andreas, A., Stoffel, T.: Nevada power: Clark station; Las Vegas, Nevada, US (data). Technical Report DA-5500-56508, NREL, Golden, US (2006)
[23] Andreas, A., Stoffel, T.: NREL solar radiation research laboratory (SRRL): Baseline measurement system (BMS); Golden, Colorado, US (data). Technical Report DA-5500-56488, NREL, Golden, US (1981)

The effect of Daylight Saving Time on Spanish Electrical Consumption

Eduardo Caro[1], Jesús Juan[1], Jesús Rupérez[2], Carlos Rodríguez[2], Ana Rodríguez[2], Juan José Abellán[2]

[1] *Universidad Politécnica de Madrid*, Madrid, Spain
[2] *Red Eléctrica de España*, Spain

(Corresponding authors: eduardo.caro@upm.es jesus.juan@upm.es)

Abstract. In this work, a detailed simulation-based analysis is conducted to assess the impact of adopting Daylight Saving Time (DST) on the consumption in the Spanish electric power system. To carry out this study, it has been used the short-term electric load forecasting software currently used by the Spanish transmission system operator, simulating the load in case of removing the DST in Spain. Obtained results denote that the DST may have a positive impact on the reduction of the electric energy demand.

Keywords: electricity demand forecasting, daylight saving clock change, Spanish electric energy system.

1 Introduction

Daylight Saving Time (DST) is the practice adopted by many countries worldwide of advancing clocks during summer months (usually from March until October) so that evening sunlight has a longer duration, while sacrificing normal sunrise times. Consequently, the DST is a measure to improve the use of available daylight during the summer months, which results in a change in energy consumption.

Technical literature is rich in references concerning the effect of daylight-saving time change over the electric consumption [1], [2]. The problem has been analyzed in several countries and regions, such as Great Britain [3], Indiana [4], Ontario [5], Chile [6], Turkey [7], Southern Norway and Sweden [8], Jordan [9], Kuwait [10], Australia [11], Argentina [12], among others.

Most of the above works indicate that the implementation of DST provokes a small reduction of the electric energy consumption [1], [2]. In some studies, this effect has been quantified: in Jordan, the load decreases 0.2% in general (saving lighting energy, but increasing for heating and cooling purposes) [9]; in Great Britain, in Chile and in Turkey, the reduction is estimated around 0.3% [3], 0.55% [6], and 0.7% [7], respectively. A higher reduction is reported in Southern Norway and Sweden, indicating a decrease at least 1% in both countries [8]. In Ontario, for the evening period, this reduction has been estimated to be 1.5%, approximately [5]. On the other hand, other studies indicate that the effect on total consumption is negligible, but it has a significant impact on the redistributional effect among hours; this is the case of Australia [10].

adfa, p. 1, 2019.

Finally, other works indicates that this reduction is not clear, or even mixed. This is the case of Argentina, observing an increment of total electric demand between 0.4% and 0.6%, but a decrease in the peak consumption between 2.4% and 2.9% [12].

Even some analysis denote that the DST implementation provokes an energy consumption increase. This is the case of Kuwait, reporting an increment of 0.07% [10]. In Indiana, it has been estimated an increase of 2-4% in the fall season, leading to a 1% of increment considering the whole year.

To the best of Author's knowledge, not so many works have been focused on the Spanish case. In this work, the impact of adopting Daylight Saving Time on the consumption in the Spanish electric power system is assessed, based on a detailed simulation-based analysis. The simulation has been performed using the short-term electric demand forecaster currently used by *Red Eléctrica de España, REE* (the Spanish transmission system operator) [13], estimating the most-likely electric consumption in case of removing the DST in Spain. Obtained results indicate that the DST may have a positive impact on the reduction of the electric energy demand.

The purpose of this paper is twofold. First, the sunlight effect is implemented in the short-term electric load forecasting software. Second, the daylight-saving clock change's influence on the Spanish electric consumption has been analyzed.

2 Estimating the DST effects on the electric load

In this section, the short-term electric load forecasting software is modified in order to consider the sunlight effect. Then a simulation is performed, comparing the case of (i) considering the daylight-saving clock change, which is the real case, and (ii) disregarding the DST effect.

2.1 Procedure

In order to perform the simulation, the load forecasting model must be slightly modified first, to consider the daylight effect in a more realistic way. This procedure comprises two steps: first, the sunset/sunrise times must be computed for Spain. Second, the daylight duration information must be included in the model as an exogenous variable (regressor).

Step 1) Computation of sunset and sunrise times

To obtain the exact time of sunrise and sunset hours, we have made use of the Excel file created by the Department "*Earth System Research Laboratory*" (web www.esrl.noaa.gov) pertaining to the agency "*National Oceanic and Atmospheric Administration*" [10]. This datafile computes the sunrise/sunset moments given any geographical location determined by its latitude and longitude.

	A	B	C	D	U	V	W	X	Y	Z	AA
1	NOAA Solar Calculations - Change any of the highlighted cells to get solar position data for that location and time-of-day for a year.			Date	var y	Eq of Time (minutes)	HA Sunrise (deg)	Solar Noon (LST)	Sunrise Time (LST)	Sunset Time (LST)	Sunlight Duration (minutes)
2	Latitude (+ to N)	40,46		01/01/2018	0,04	-3,54	70,07	13:18:20	8:38:05	17:58:36	560,52259
3	Longitude (+ to E)	-3,7		02/01/2018	0,04	-4,00	70,16	13:18:48	8:38:10	17:59:26	561,26569
4	Time Zone (+ to E)	1		03/01/2018	0,04	-4,47	70,26	13:19:16	8:38:14	18:00:18	562,07166
5	Local Time (hrs)	12:00:00		04/01/2018	0,04	-4,92	70,37	13:19:43	8:38:15	18:01:11	562,93972
6	Year	2018		05/01/2018	0,04	-5,37	70,48	13:20:10	8:38:14	18:02:06	563,86907

Fig. 1. Extract from the Excel file for the computation of sunrise and sunset time, from *Earth System Research Laboratory.*

In this study, three Spanish cities are considered: Madrid (located in the central zone of the mainland), Barcelona (located in the Western region of the country) and Santiago de Compostela (located in the Eastern region of Spain). The sunrise and sunset times for the aforementioned three cities are provided in Fig. 2, for all de days of the year, considering UTC time.

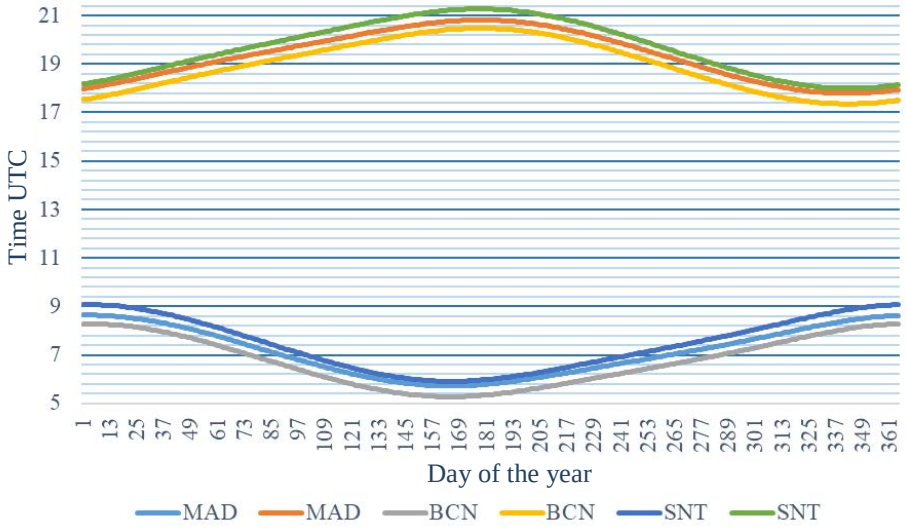


Fig. 2. Sunrise and sunset times for Madrid (MAD), Barcelona (BCN) and Santiago (SNT)

As it can be observed in Fig. 2, daylength varies throughout the year, and there is a significant difference of sunrise/sunset times for the three selected cities: almost 45 minutes of difference between Barcelona and Santiago de Compostela. It should be noted that the curves in Fig. 2 are always valid, no matter which year is considered.

In order to validate the previous values, we have accessed to the webpage of the "*Spanish National Astronomical Observatory - National Geographic Institute*", from the Spanish Ministry of Development. In the web [11], a text file can be automatically generated containing the sunrise and sunset times for a specific year of any of the

Spanish regions, considering local time. Fig. 3 provides the sunrise ("Ort" column) and sunset ("Ocas" column) local time for Madrid during the year 2018.

MADRID — SALIDA Y PUESTA DE SOL PARA 2018 — Observatorio Astronómico Nacional
Latitud y longitud: 40 24 35, - 3 41 11 — Instituto Geográfico Nacional
Año 2018 — Hora oficial en la península y Baleares — Ministerio de Fomento, España

Día	Enero		Febrero		Marzo		Abril		Mayo		Junio		Julio		Agosto		Septiem.		Octubre		Noviemb.		Diciemb.	
	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas	Ort	Ocas
	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m
1	838	1759	824	1833	749	1906	759	2039	714	2110	647	2139	648	2149	712	2130	742	2047	811	1957	744	1812	818	1749
2	838	1800	823	1834	747	1907	757	2040	713	2111	646	2139	649	2149	713	2129	743	2045	812	1956	745	1811	819	1749
3	838	1801	822	1835	746	1908	756	2041	712	2112	646	2140	649	2149	714	2127	744	2044	813	1954	747	1809	820	1748
4	838	1802	821	1837	744	1909	754	2042	711	2113	646	2141	650	2148	715	2126	745	2042	814	1952	748	1808	821	1748
5	838	1802	820	1838	743	1911	752	2043	709	2114	645	2141	650	2148	716	2125	746	2040	815	1951	749	1807	822	1748
6	838	1803	819	1839	741	1912	751	2044	708	2115	645	2142	651	2148	717	2124	747	2039	816	1949	750	1806	823	1748
7	838	1804	818	1840	739	1913	749	2045	707	2116	645	2143	652	2147	718	2123	748	2037	817	1947	751	1805	824	1748
8	838	1805	817	1841	738	1914	748	2046	706	2117	645	2143	652	2147	719	2122	749	2036	818	1946	752	1804	825	1748
9	838	1806	816	1843	736	1915	746	2047	705	2118	644	2144	653	2147	720	2120	750	2034	819	1944	754	1803	826	1748
10	837	1807	815	1844	735	1916	744	2048	704	2119	644	2144	654	2146	721	2119	751	2032	820	1943	755	1802	827	1748
11	837	1808	813	1845	733	1917	743	2049	703	2120	644	2145	654	2146	722	2118	751	2031	821	1941	756	1801	828	1748
12	837	1809	812	1846	731	1918	741	2050	702	2121	644	2145	655	2145	723	2116	752	2029	822	1940	757	1800	828	1748
13	837	1811	811	1847	730	1919	740	2051	701	2122	644	2146	656	2145	723	2115	753	2027	823	1938	758	1759	829	1749
14	836	1812	810	1849	728	1920	738	2053	700	2123	644	2146	656	2144	724	2114	754	2026	824	1937	759	1758	830	1749
15	836	1813	808	1850	727	1921	737	2054	659	2124	644	2147	657	2144	725	2112	755	2024	825	1935	801	1758	831	1749
16	835	1814	807	1851	725	1922	735	2055	658	2125	644	2147	658	2143	726	2111	756	2022	826	1934	802	1757	831	1749
17	835	1815	806	1852	723	1923	734	2056	657	2126	644	2147	659	2143	727	2110	757	2021	828	1932	803	1756	832	1750
18	835	1816	804	1853	722	1925	732	2057	656	2127	644	2148	700	2142	728	2108	758	2019	829	1931	804	1755	833	1750
19	834	1817	803	1855	720	1926	731	2058	655	2128	644	2148	700	2141	729	2107	759	2017	830	1929	805	1755	833	1750
20	833	1818	802	1856	718	1927	729	2059	654	2129	644	2148	701	2141	730	2105	800	2016	831	1928	806	1754	834	1751
21	833	1820	800	1857	717	1928	728	2100	653	2130	645	2148	702	2140	731	2104	801	2014	832	1926	808	1753	834	1751
22	832	1821	759	1858	715	1929	726	2101	653	2131	645	2149	703	2139	732	2102	802	2012	833	1925	809	1753	835	1752
23	832	1822	758	1859	714	1930	725	2102	652	2131	645	2149	704	2138	733	2101	803	2011	834	1923	810	1752	835	1752
24	831	1823	756	1900	712	1931	724	2103	651	2132	645	2149	705	2137	734	2059	804	2009	835	1922	811	1752	836	1753
25	830	1824	755	1902	810	2032	722	2104	651	2133	646	2149	706	2136	735	2058	805	2007	836	1921	812	1751	836	1754
26	829	1826	753	1903	809	2033	721	2105	650	2134	646	2149	706	2136	736	2056	806	2005	837	1919	813	1751	836	1754
27	829	1827	752	1904	807	2034	719	2106	649	2135	646	2149	707	2135	737	2055	807	2004	839	1918	814	1750	837	1755
28	828	1828	750	1905	805	2035	718	2107	649	2136	647	2149	708	2134	738	2053	808	2002	740	1817	815	1750	837	1756
29	827	1829			804	2036	717	2108	648	2136	647	2149	709	2133	739	2052	809	2001	741	1816	816	1750	837	1756
30	826	1830			802	2037	716	2109	648	2137	648	2149	710	2132	740	2050	810	1959	742	1814	817	1749	837	1757
31	825	1832			800	2038			647	2138			711	2131	741	2049			743	1813			838	1758
	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m	h m

Fig. 3. Extract from the text file containing the sunrise and sunset times, from the *Spanish National Astronomical Observatory* [1]

Since this database uses local time, and considering that the daylight-saving change day varies depending on the year, the text files downloaded from [11] are only valid for the specific year considered. In Fig. 3 it can be observed that the daylight saving changes occur in March 25th and October 28th, causing one hour difference of the sunrise/sunset time comparing with the previous day.

Step 2) Implementation of the daytime regressor

The short-term electric load forecaster has been modified to include the daytime information. According to the previous plot, depending on the day of the year, the set of hours 6-7-8 am and 6-7-8 pm may have sunlight or not. In other words, in Spain, there is always sunlight from 9 am to 5 pm, no matter the period of the year. Likewise, from 9 pm to 5 pm, there are no sunlight in any day of the year. However, the rest of the hours, depending on the period of the year, may have sunlight or not.

A set of 24 dummy variables (one for each hour) has been created, modeling the daytime effect: $l_{h,d} \in [0, 1]$, where indexes *h* and *d* indicate the hour and the day. For each hour *h*, the parameter $l_{h,d}$ is set to one if the *h*-th hour for the *d*-th day has sunlight, $l_{h,d} = 0$ otherwise. Fig. 4 provides the values for the parameter $l_{h,d}$ for a whole year

1

Web page: www.fomento.gob.es/salidapuestasol/2018/Madrid-2018.txt (accessed: 2019 June)

and for hours comprised between 6 am to 8 pm, for the UTC time and local time cases. Parameter $l_{h,d}$ is included in the forecasting model as an exogenous variable (regressor).

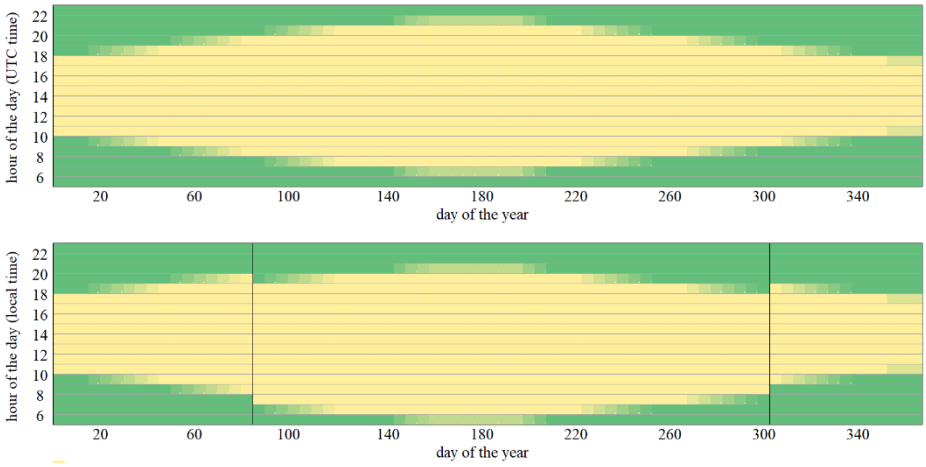


Fig. 4. Values for the parameters $l_{h,d}$ for hours $h \in [6, 22]$, for UTC and local times. Yellow and green color indicate $l_{h,d} = 1$ and $l_{h,d} = 0$, respectively.

3 Case Study

Once the daytime information is included in the model, and considering the actual implementation of the DST effect in the algorithm [13], the Spanish load short-term forecasting software can be used to simulate the effect of considering/disregarding the Daylight Saving Time. In the following subsections the cases of March and October are studied, for year 2017.

Considering that the predicting model has been designed and created for short-term forecasts (from one to ten days ahead), this study analyzes the effect of DST on the local period close the clock-change day.

3.1 DST effect on March

Considering that the clock-change day took place on March 26th, 2017, the Spanish load forecasting model has been used to predict the load behavior in case of: (i) considering the DST effect and (ii) disregarding intentionally this effect. Both cases have been simulated at 0.00 hours on March 26th, 2017, generating forecasts from one to ten days ahead.

Figs. 5 and 6 provides the forecasted load values for cases considering the DST (labeled as '*DST*') and neglecting the DST (labeled as '*no DST*'), and the observed load. The difference between curves '*DST*' and '*no DST*' corresponds to the effect of neglecting the daylight saving time.

The effect of removing the DST can be observed in Fig. 6: electric consumption during 8 pm and 9 pm increases significantly.

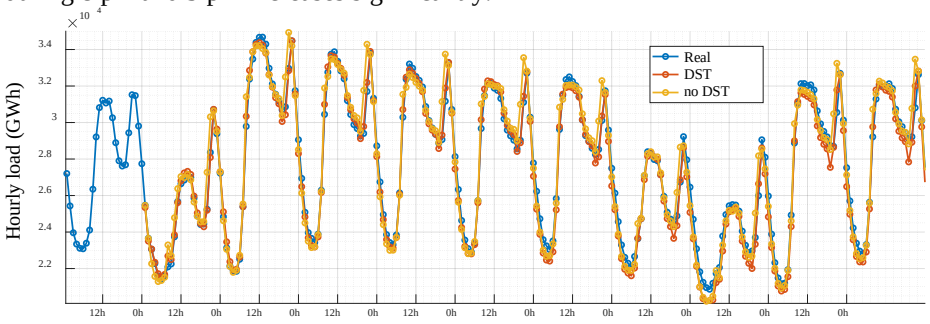


Fig. 5. Observed and forecasted electric load for the period: 25/03/2017 - 05/04/2017.

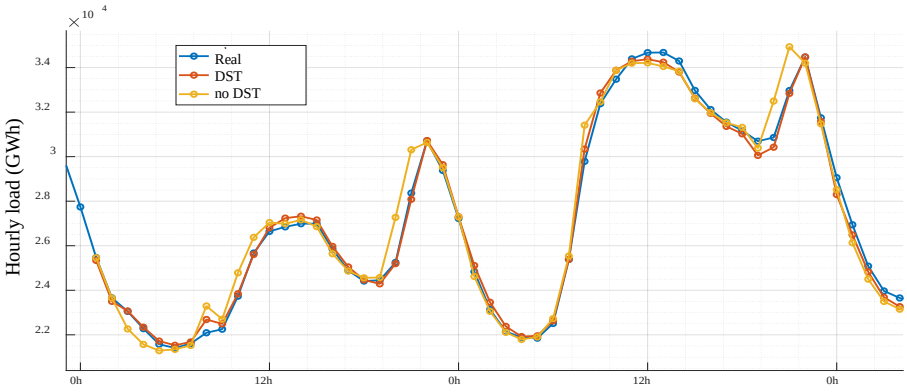


Fig. 6. Observed and forecasted electric load for the DST on March 2017.

From the daily consumption perspective, Table 1 provides the forecasted daily electric load for both cases (fourth and fifth columns), and the increment of daily demand in case of neglecting DST (sixth column). It can be observed that the DST reduces the electricity consumption around 0.65-1.06 % for the following days after the clock-change day.

Table 1. Forecasted daily load considering/neglecting the DST: March 2017

			Daily electric load (GWh)		Increment
MARCH			DST	no DST	DST vs no DST
Sunday	29/10/2017	d+1	603,0	606,9	0,64%
Monday	30/10/2017	d+2	711,2	715,2	0,55%
Tuesday	31/10/2017	d+3	708,0	712,4	0,62%
Wednesday	01/11/2017	d+4	696,0	699,2	0,46%
Thursday	02/11/2017	d+5	687,9	694,4	0,95%
Friday	03/11/2017	d+6	675,6	685,7	1,49%
Saturday	04/11/2017	d+7	604,2	612,5	1,38%
Sunday	05/11/2017	d+8	559,0	566,8	1,40%
Monday	06/11/2017	d+9	663,5	673,8	1,55%
Tuesday	07/11/2017	d+10	680,4	690,9	1,54%
Average (first five days)			681,2	685,6	0,65%
Average (first ten days)			658,9	665,8	1,06%

Table 2 provides the effect of considering the DST on Spanish hourly load, for the local period of the clock-change day of March 2017. As it can be observed, there is an increment of 6-7% at 8.00-9.00 pm. Additionally, note that there is a redistribution of loads between 8.00 am and 9.00 am.

Table 2. Effect of considering/neglecting the DST on hourly load: March 2017

MARCH		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Sunday	d+1	0.5	0.6	-3.5	-3.5	-2.0	-0.9	-0.7	2.7	0.9	3.9	3.0	0.9	-0.9	-0.6	-1.1	-1.3	-0.7	0.4	1.2	8.2	7.9	-0.3	-0.5	0.0
Monday	d+2	-2.0	-1.7	-1.2	-0.6	-0.2	0.4	0.6	3.5	-1.2	0.0	-0.3	-0.5	-0.6	0.1	-0.1	0.1	0.5	0.9	1.1	6.8	6.4	-0.8	-0.3	0.8
Tuesday	d+3	-1.3	-1.3	-0.8	-0.5	0.0	0.5	0.2	2.9	-1.6	-0.5	-0.4	-0.2	-0.2	0.4	0.4	0.4	0.6	1.0	1.3	6.7	6.5	-0.5	-0.2	0.6
Wednesday	d+4	-1.0	-1.1	-1.0	-0.8	-0.5	0.0	-0.2	2.7	-1.7	-0.7	-0.7	-0.7	-0.5	0.2	0.6	0.4	0.3	0.6	1.1	6.3	5.8	-0.5	0.2	1.4
Thursday	d+5	-0.4	-0.5	-0.3	-0.1	0.2	0.7	0.4	2.9	-1.3	-0.5	-0.5	-0.3	-0.1	0.6	0.6	0.8	0.6	1.0	2.1	7.3	6.7	0.1	0.5	1.6
Friday	d+6	0.7	0.5	0.7	0.9	1.3	1.9	1.5	3.6	-0.9	0.4	0.3	0.5	0.7	1.2	1.1	1.0	0.7	1.0	2.2	7.0	6.4	0.4	1.1	1.9
Saturday	d+7	0.7	0.4	0.5	0.7	0.7	1.4	0.9	3.4	0.1	0.8	0.5	0.1	-0.1	0.5	0.4	0.7	0.9	1.3	2.7	7.5	6.7	0.4	0.9	1.5
Sunday	d+8	0.4	0.3	0.1	0.4	0.8	1.3	1.0	3.1	0.4	1.4	0.7	0.3	-0.3	0.3	0.7	0.9	1.1	1.5	3.1	7.3	5.7	-0.2	1.0	2.3
Monday	d+9	0.8	0.7	0.7	0.7	1.1	1.6	1.3	3.2	-0.6	0.5	0.7	0.7	0.7	1.1	1.2	1.3	1.3	1.8	3.5	6.4	4.6	0.0	1.3	2.4
Tuesday	d+10	0.8	0.7	0.7	0.7	1.1	1.6	1.3	3.2	-0.6	0.5	0.7	0.7	0.7	1.1	1.2	1.3	1.3	1.8	3.5	6.4	4.6	0.0	1.3	2.4
Average (first five days)		-0.8	-0.8	-1.4	-1.1	-0.5	0.2	0.1	2.9	-1.0	0.4	0.2	-0.2	-0.5	0.1	0.1	0.1	0.3	0.8	1.4	7.0	6.7	-0.4	-0.1	0.9
Average (first ten days)		-0.1	-0.1	-0.4	-0.2	0.2	0.9	0.6	3.1	-0.6	0.6	0.4	0.2	-0.1	0.5	0.5	0.6	0.6	1.1	2.2	7.0	6.1	-0.2	0.5	1.5

3.2 DST effect on October

As in the previous subsection, the Spanish load forecasting model has been used to predict the load behavior with/without the DST. During year 2017, the clock-change day took place on October 29th, 2017. Both cases have been simulated at 0.00 hours on October 29th, 2017, generating forecasts from one to ten days ahead.

Figs. 7 and 8 provides the forecasted load values for cases considering the DST (labeled as '*DST*') and neglecting the DST (labeled as '*no DST*'), and the observed load. The difference between curves '*DST*' and '*no DST*' corresponds to the effect not changing the clock on October 29th, 2017.

The effect of the clock change can be observed in Fig. 8: electric consumption during 8 pm increases significantly. It should be noted that the sunset time changes from 7.20 pm (28/10/2017) to 8.20 pm (28/10/2017). Consequently, public and private lighting electric consumption commences one hour before.

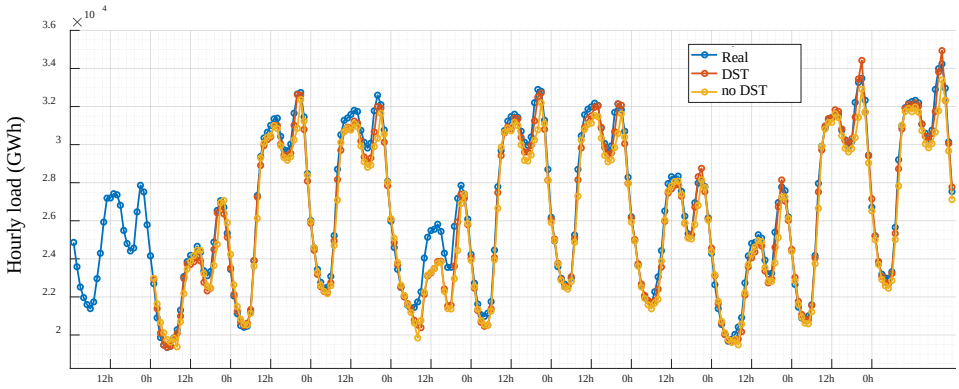


Fig. 7. Observed and forecasted electric load for the period: 27/10/2017 - 07/11/2017.

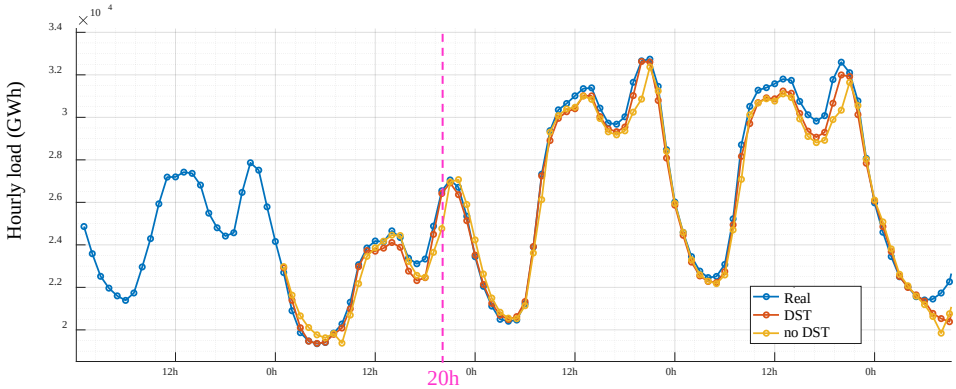


Fig. 8. Observed and forecasted electric load for the DST on October 2017.

From the daily consumption perspective, Table 3 provides the forecasted daily electric load for both cases (fourth and fifth columns), and the increment of daily demand in case of removing the clock change in October (sixth column). It can be observed that the clock change increases the electricity consumption around 0.65-1.06 % for the following days after the clock-change day.

Table 3. Forecasted load considering/neglecting the clock change: October 2017

			Daily electric load (GWh)		Increment
OCTOBER			DST	no DST	DST vs no DST
Sunday	29/10/2017	d+1	546,2	548,1	0,35%
Monday	30/10/2017	d+2	658,6	655,9	-0,41%
Tuesday	31/10/2017	d+3	671,6	667,1	-0,67%
Wednesday	01/11/2017	d+4	557,0	553,9	-0,55%
Thursday	02/11/2017	d+5	663,9	657,9	-0,90%
Friday	03/11/2017	d+6	679,5	672,1	-1,10%
Saturday	04/11/2017	d+7	607,1	603,3	-0,63%
Sunday	05/11/2017	d+8	554,0	552,7	-0,24%
Monday	06/11/2017	d+9	676,5	665,9	-1,57%
Tuesday	07/11/2017	d+10	698,9	688,0	-1,55%
Average (first five days)			619,4	616,6	-0,44%
Average (first ten days)			631,3	626,5	-0,73%

Table 4 provides the local effect of removing the clock change in October 2017 on the hourly demand. As it can be observed, there is a decrement -6% at 8.00 pm. Additionally, note that there is a redistribution of loads between 8.00 am and 9.00 am.

Table 4. Effect of clock change on hourly load: October 2017

OCTOBER		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Sunday	d+1	0.2	1.1	2.8	3.3	2.1	1.1	0.2	-3.6	-1.4	-3.5	-1.2	0.7	1.4	1.4	2.3	2.0	1.1	0.1	-3.5	-6.3	0.2	2.7	3.0	3.0
Monday	d+2	2.2	1.2	0.7	0.3	-0.5	-0.9	-1.3	-4.1	1.4	0.4	0.4	0.2	-0.1	-0.5	-0.3	-0.5	-0.5	-0.6	-2.5	-5.4	-0.7	1.5	1.1	0.3
Tuesday	d+3	0.6	0.5	0.3	0.1	-0.2	-0.7	-1.0	-3.8	1.4	-0.1	-0.1	-0.3	-0.4	-0.7	-0.9	-0.9	-0.9	-1.3	-2.5	-5.2	-0.9	1.4	0.7	0.1
Wednesday	d+4	0.9	0.5	0.5	0.4	-0.3	-0.8	-0.7	-3.4	1.8	0.5	0.2	0.2	0.0	-0.3	-0.2	-0.6	-0.6	-1.0	-2.8	-5.7	-1.8	0.7	0.2	-0.6
Thursday	d+5	0.3	0.3	0.5	0.3	-0.1	-0.5	-0.5	-3.0	1.5	0.1	-0.2	-0.4	-0.7	-1.2	-1.3	-1.5	-1.6	-1.7	-3.2	-5.3	-1.6	0.9	0.0	-0.7
Friday	d+6	-0.3	-0.2	-0.2	-0.1	-0.7	-1.1	-0.8	-3.1	1.4	-0.4	-0.9	-1.2	-1.3	-1.8	-1.9	-1.4	-1.2	-1.3	-2.7	-4.8	-1.4	1.1	-0.1	-0.9
Saturday	d+7	-0.3	-0.5	-0.7	-0.7	-1.0	-1.1	-0.6	-2.4	1.3	0.2	0.3	0.6	0.7	0.6	0.3	0.0	-0.2	-1.0	-3.4	-5.3	-2.1	0.8	-0.2	-0.8
Sunday	d+8	0.2	0.5	0.5	0.5	0.2	-0.1	0.5	-1.6	2.0	0.5	0.5	0.8	1.1	0.7	0.9	0.8	0.4	-0.6	-3.4	-6.0	-3.2	0.8	0.4	-0.5
Monday	d+9	-0.8	-0.7	-0.7	-0.7	-1.1	-1.6	-1.3	-3.2	0.6	-0.5	-0.7	-0.7	-0.7	-1.1	-1.2	-1.3	-1.3	-1.8	-3.4	-6.0	-4.4	0.0	-1.3	-2.4
Tuesday	d+10	-0.8	-0.7	-0.7	-0.7	-1.1	-1.6	-1.3	-3.2	0.6	-0.5	-0.7	-0.7	-0.7	-1.1	-1.2	-1.3	-1.3	-1.8	-3.4	-6.0	-4.4	0.0	-1.3	-2.4
Average (first five days		0.9	0.7	0.9	0.9	0.2	-0.4	-0.6	-3.6	0.9	-0.5	-0.2	0.1	0.0	-0.2	-0.1	-0.3	-0.5	-0.9	-2.9	-5.6	-1.0	1.5	1.0	0.4
Average (first ten days		0.2	0.2	0.3	0.3	-0.3	-0.7	-0.7	-3.1	1.1	-0.3	-0.2	-0.1	-0.1	-0.4	-0.3	-0.5	-0.6	-1.1	-3.1	-5.6	-2.0	1.0	0.3	-0.5

4 Conclusions

In this paper, the short-term Spanish load forecast model has been slightly modified to consider the daytime, and it has been used to simulate the local effect of disregarding the DST on March 2017, and the local effect of disregarding the change clock in October, 2017.

According to the performed numerical simulations, the DST change in March provokes a decrement of electric daily load consumption around 0.6-1.0% (decrement of 6-7% between 8.00 pm and 9.00 pm). On the other hand, the clock change in October causes an increment of daily demand about 0.4-0.7% (increment of 5% at 8 pm due to public and private lighting demand).

Future work will focus on expanding this study to more years and a broader period.

Acknowledgements

This work has been funded by *Red Eléctrica de España* (REE) as a R&D project.

References

1. Havranek, Tomas, Dominik Herman, and Zuzana Irsova. "Does daylight saving save electricity? A meta-analysis." *Energy Journal* 39.2 (2018): 35-61.
2. Aries, Myriam BC, and Guy R. Newsham. "Effect of daylight saving time on lighting energy use: A literature review." *Energy policy* 36.6 (2008): 1858-1866.
3. Hill, S. I., et al. "The impact on energy consumption of daylight saving clock changes." *Energy Policy* 38.9 (2010): 4955-4965.
4. Matthew J. Kotchen and Laura E. Grant "Does Daylight Saving Time Save Energy? Evidence from a Natural Experiment in Indiana", *The Review of Economics and Statistics* 2011 93:4, 1172-1185
5. Rivers, Nicholas. "Does daylight savings time save energy? Evidence from Ontario." *Environmental and resource economics*70.2 (2018): 517-543.
6. Verdejo, Humberto, et al. "Impact of daylight saving time on the Chilean residential consumption." *Energy Policy* 88 (2016): 456-464.
7. Karasu, Servet. "The effect of daylight saving time options on electricity consumption of Turkey." *Energy* 35.9 (2010): 3773-3782.
8. Mirza, Faisal Mehmood, and Olvar Bergland. "The impact of daylight saving time on electricity consumption: Evidence from Southern Norway and Sweden." *Energy Policy* 39.6 (2011): 3558-3571.
9. Momani, Mohammad Awad, Baharudin Yatim, and Mohd Alauddin Mohd Ali. "The impact of the daylight saving time on electricity consumption—A case study from Jordan." *Energy Policy* 37.5 (2009): 2042-2051.
10. Krarti, Moncef, and Ali Hajiah. "Analysis of impact of daylight time savings on energy use of buildings in Kuwait." *Energy Policy* 39.5 (2011): 2319-2329.
11. Choi, Seungmoon, Alistair Pellen, and Virginie Masson. "How does daylight saving time affect electricity demand? An answer using aggregate data from a natural experiment in Western Australia." *Energy Economics* 66 (2017): 247-260.
12. Hancevic, Pedro, and Diego Margulis. "Daylight saving time and energy consumption: The case of Argentina." (2016).
13. Caro, Eduardo; Juan, Jesús; Cara, Javier, "Estimating Periodically Correlated Models for Short-Term Electricity Load Forecasting". *Conference: 37th International Symposium on Forecasting*, Cairns, Australia, 2017
14. "*Earth System Research Laboratory*" pertaining to the agency "*National Oceanic and Atmospheric Administration*". Web page: www.esrl.noaa.gov (last access: 2019 June)
15. "*Observatorio Astronómico Nacional - Instituto Geográfico Nacional*", from the Spanish Ministry of Development (Ministerio de Fomento de España).
Web: www.fomento.gob.es/salidapuestasol (last access: 2019 June)

Wind Speed Forecasting Using Kernel Ridge Regression

Mohammad Amjad Alalami, Maher Maalouf, Tarek H.M. EL-Fouly

E-mail: Mohammad.a.alalami@gmail.com, maher.maalouf@ku.ac.ae, tarek.elfouly@ku.ac.ae

Khalifa University, Abu Dhabi, UAE

Abstract. Wind speed forecasting is considered a challenging task as wind energy data is highly variable. Therefore, powerful methods are required in the prediction of wind speed. In this paper, a kernel ridge regression model is proposed. The model performance accuracy, for wind speed forecasting, is compared with two reference prediction models, namely, the persistent model and the least squares model. For the least squares and kernel regression model, moving window cross-validation is used. Historical wind speed data from Canadian weather stations is used to validate the model performance for three different forecasting horizons (1-hour, 12-hours and 24-hours ahead). Results show that forecasts made with the kernel ridge regression produced the highest accuracy compared to the least squares model and the persistent model.

Keywords: Kernel Ridge Regression, Persistent model, Least Squares Model

INTRODUCTION

Nowadays, the interest in using renewable energy is increasing to mitigate the negative impact of conventional energy resources on the environment. Wind power is considered as one of the most rapidly growing renewable energy resource worldwide. For example, wind power generation in Spain accounts for more than 4% of its electricity [1].

Generating energy through wind depends mainly on its speed. Wind speed varies from one site to another depending on various factors. Therefore, wind power is intermittent in nature. This present a great challenge for power systems operators when large wind power installations are being integrated into their electricity network. Therefore, as the penetration of wind power through the power system increases, the system's

operations will be influenced such as generation dispatch and identifying generation reserve needs. This requires accurate forecasting of available wind generation. [2].

In order to predict wind speed, different models can be used. In this paper, a kernel ridge regression model is proposed. The model performance accuracy, for wind speed forecasting, is compared with two reference prediction models, namely, the persistent model and the least squares model.

PERSISTENCE MODEL

The persistent model is based on the theory that there is a high correlation between the present and future values of the wind speed. The model uses a simple technique to predict the wind speed of the next hour. Thus, the predicted wind speed of the next hour is the same value as the current observation. This can be modeled using the following generalized linear equation.

$$\hat{Y}_{t+b} = Y_t$$

Where $\hat{Y}_{t+b}$ the predicted wind speed at time t + b is, Y_t is wind speed observation at time t. This method is widely used by meteorologists as a reference to predict the next hour wind speed. On the other hand, the accuracy of this model reduces with the increase of prediction horizon [3].

LEAST SQUARES MODEL

The least squares (LS) model strive for reducing the squared errors between the predicted and the actual values as much as possible. Therefore, the model is then able to find the line of best fit for a given data set. The mathematical formula of the model is $y=X\beta+\varepsilon$. Where ($\varepsilon= \varepsilon1, \varepsilon2, \varepsilon3 \ldots \varepsilon n$) T is the error, given that the errors have a constant variance, normally distributed, and linearly independent. The β is estimated by minimizing the following equation: summation of (εi) from i=1 to n = $\varepsilon T\varepsilon=(y-X\beta)$ T $(y-X\beta)$. Assuming that the (XTX) is a non-singular matrix, the solution is: $\beta = (XTX)\text{-}1$ XT y [4].

KERNEL RIDGE REGRESSION

Unlike the LS model, the kernel ridge regression (KRR) model does not assume linearity of the data but includes a non-linear map ϕ (.), which plots the data to another dimensional space. The mapping ϕ (.) is unidentified and the solution is based on the dot product. The kernel function uses the dot product such that the K matrix is equal to K (xi, xj) = (ϕ (xi), ϕ (xj)). The KRR model is expressed by $y=K\alpha+\varepsilon$. As the KRR unknown vector is α and it is found through minimizing the following equation: f (α) =0.5(y-Kα) ^T *(y-Kα) +0.5*λ (α^T) Kα. Where λ is greater or equal to zero and it's considered as a regularization parameter. The KRR model can be expressed as: (K+λIn) α= y. Therefore, KRR is a powerful model when the data is assumed to be non-linear [5].

PROCEDURE

In order to examine each model, historical wind speed data from Canadian weather stations was used for wind speed, wind pressure, humidity, and wind direction. Also, different forecast time horizons (1 hour, 12 hours and 24 hours ahead) were examined to compare the accuracy of each model using five key performance indicators (KPIs)the mean square error (MSE), the root mean square error (RMSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the squared correlation coefficient (R^2). The KPIs were used to determine the best model as a predictor of the Canadian Wind Speed data.

For the KRR and LS models, moving window cross-validation was used. Also, the data is divided into two sets, the training, and testing sets. For example, when predicting the next hour wind speed, the training set is only 24 hours and the testing is 720 data points which are equivalent to 30 days. As each data point represents an hour, a day will consist of 24 consecutive data points. The independent variables for each model are different depending on the R^2 value. Only the variables that generate high R^2 are used. The results of the models are generated by MATLAB-R-2017B.

EXPERIMENTAL RESULTS AND DISCUSSION

This section presents the prediction results for the three models when tested to forecast wind speed for three different time horizons.

One Hour Time Horizon

When one-hour time horizon is used, the data updates its training and testing sets after predicting each hour. Table 1 shows the squared correlation coefficient value and input variables (independent variables) for each model when the time horizon is one. The KRR scores the highest value of R^2 compared to the LS and persistent models. Thus, the KRR predictions are closer to the actual value.

Table 1: Models Accuracy - Time Horizon = 1

	Input Variables	R^2
KRR	Wind speed, direction, pressure, and humidity	0.9633
Persistent	Wind speed	0.8745
LS	Wind speed	0.8598

The MSE, MAPE, RMSE, and MAE values of each hour is calculated for each model. The lower these values are, the more accurate the model is. Table 2 shows the average of all the 720 points using different models. The MAPE average is not calculated as some values are zero. The KRR generates the lowest averages of MSE, RMSE, and MAE. The second best model is the persistent model according to the averages of the measurements of error. In figure 1, the trend of these KPIs (the average value per day) is observed.

Table 2: Average error measure over the entire tested interval - Time Horizon = 1

	MSE	*RMSE*	*MAE*
KRR	8.6	2.9	1.3
Persistent	26.3	5.1	4.0
LS	29.3	5.4	3.7

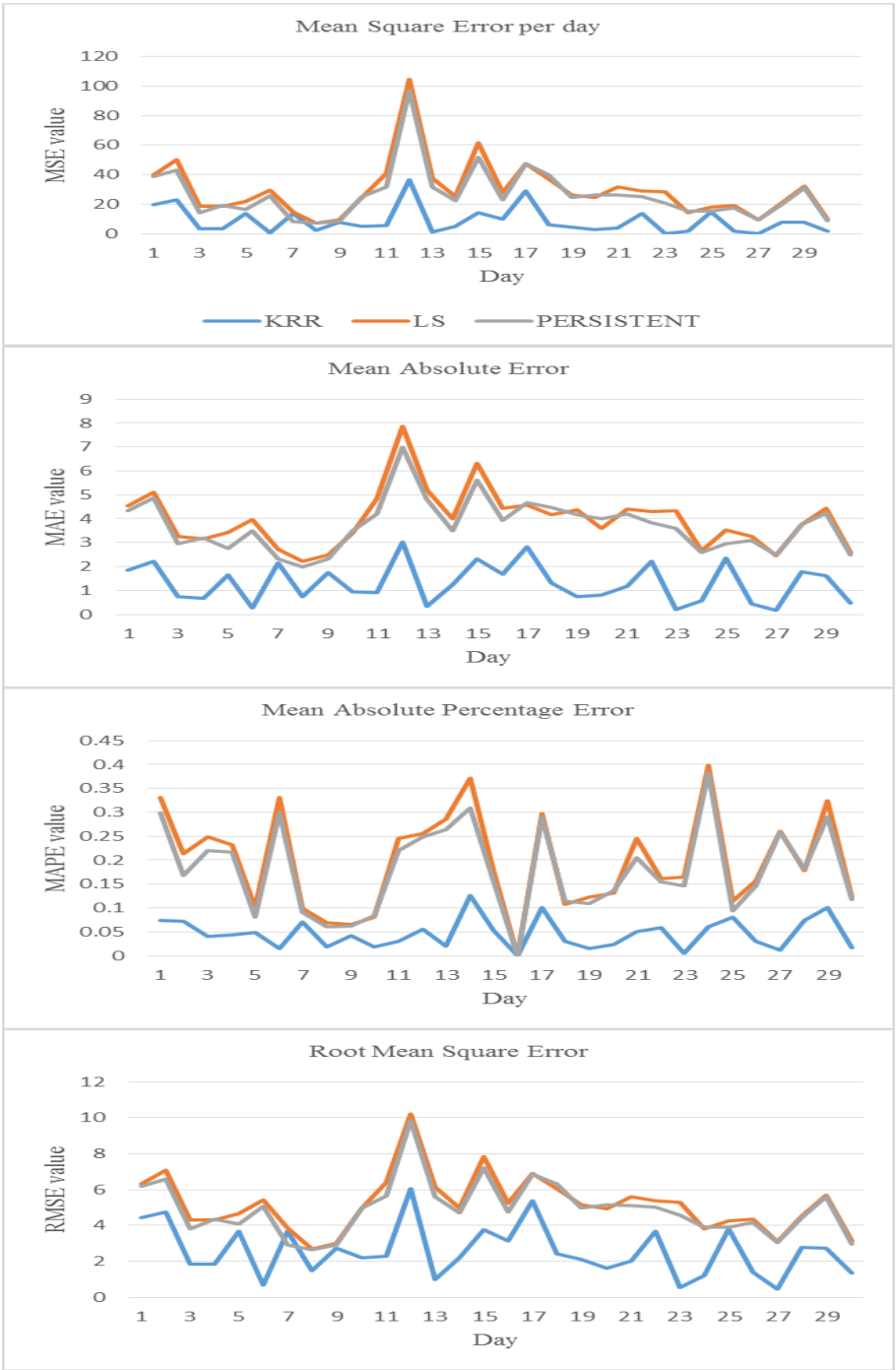


Fig. 1: Daily average error measure - time horizon = 1

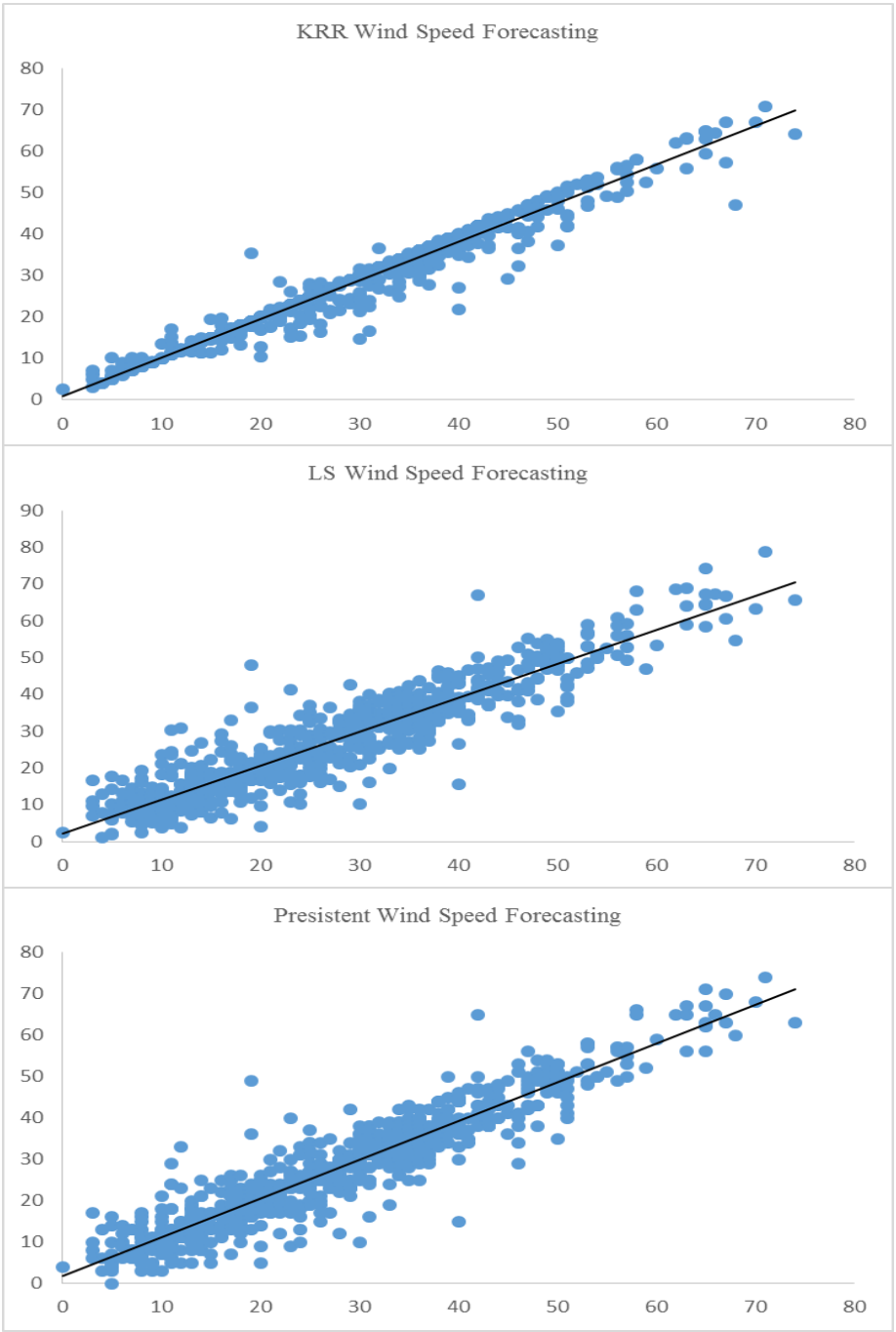


Fig. 2: Sample of the actual versus forecasted wind speed– time horizon = 1

Twelve Hours' Time Horizon

When twelve hours' time horizon is used, the data updates its training and testing sets after predicting every 12 hours. Also, the training set is increased from 24 to 80 data points only and the testing set still consists of 720 hours. Table 3 shows the squared correlation coefficient value and input variables for each model when the time horizon is twelve. The KRR model generates the highest R^2 value. The persistent model updates every twelve hours, the model predicts the wind speed a similar value for the following times t, t+1, t+2…t+11 (all the 12 values will equal to the value of the wind speed at t-1). For the KRR and LS models, the training set of 80 values seems to be insufficient. Increasing the training set increases the value of R^2. Due to the limited data set, only 80 values are trained. Therefore, the models did not produce high R^2 value and a bigger data set would be required to generate more accurate forecasts.

Table 3: Models Accuracy –Time Horizon = 12

	Input Variables	R^2
KRR	Wind speed, direction, pressure, and humidity	0.5271
Persistent	Wind speed	0.3988
LS	Wind speed	0.0371

The MSE, MAPE, RMSE, and MAE values of each hour is calculated for each model. As mentioned earlier, the lower these values are, the more accurate the model is. Also, the MAPE average is not calculated because some values are zero. Table 4 shows the average of all the 720 hours using the proposed models. The KRR model generates the lowest averages of MSE, RMSE, and MAE. The persistent model beats the LS model by generating more accurate results. In figure 2, the trend of these KPIs (the average value per day) is observed.

Table 4: Average error measure over the entire tested interval -Time Horizon = 12

	MSE	*RMSE*	*MAE*
KRR	102.9	10.1	7.6
Persistent	151.4	12.3	9.2
LS	206.7	14.4	11.6

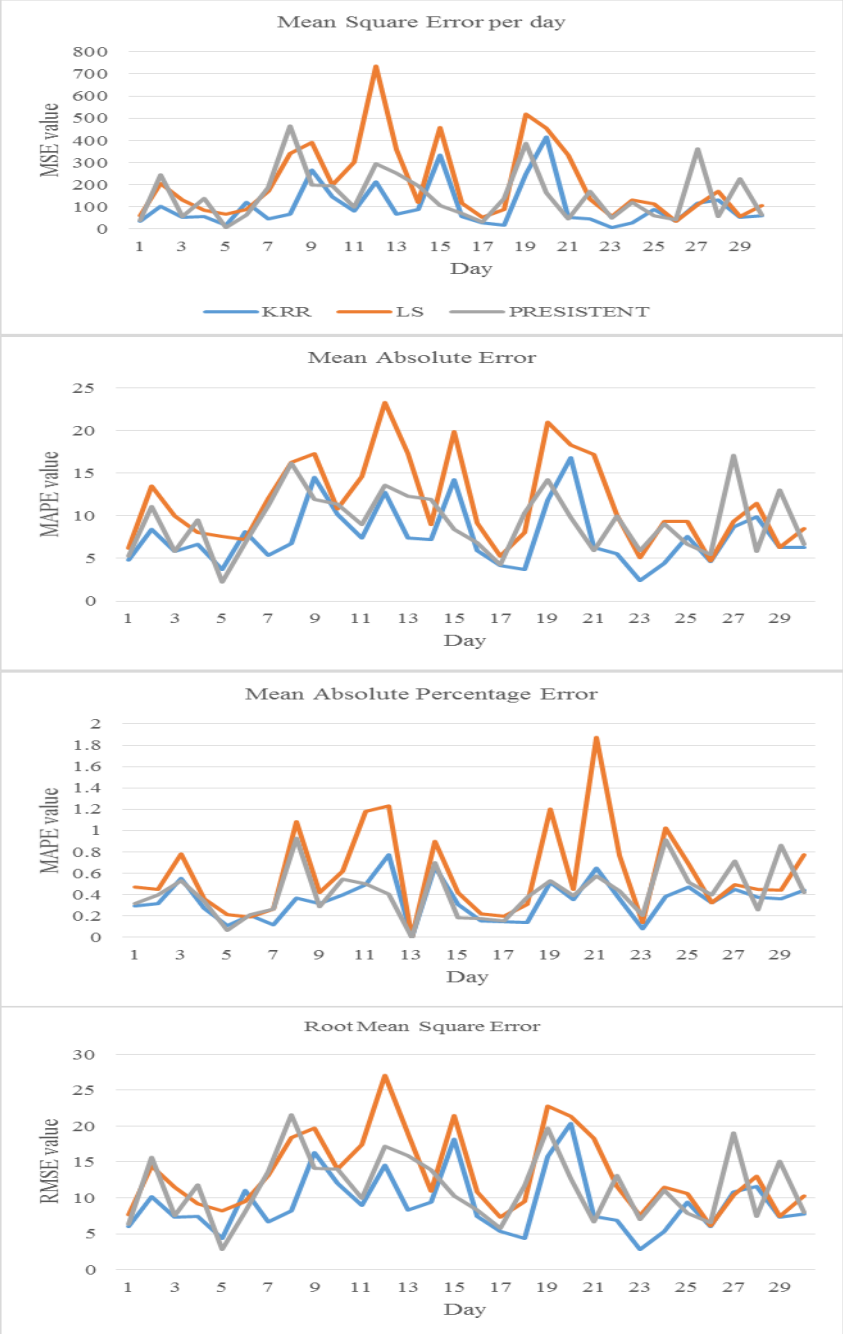


Fig. 3: Daily average error measure - time horizon = 12

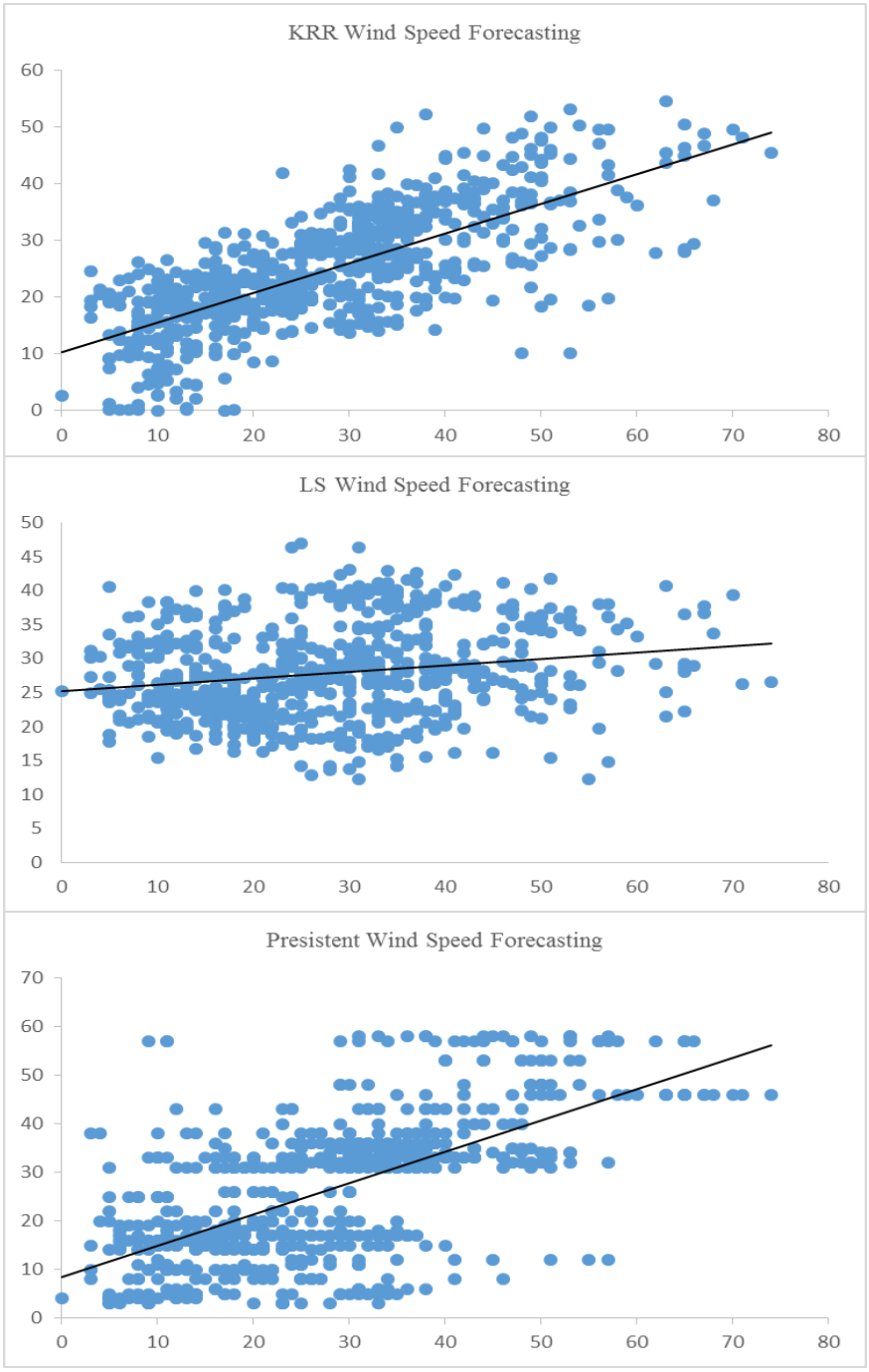


Fig. 4: The actual versus forecasted wind speed– time horizon = 12

Twenty-Four Hours' Time Horizon

When twenty-four hours' time horizon is used, the data updates its training and testing sets every 24 hours. Also, the training set is increased to 4800 data points and the testing set remains as 720 points. Table 5 shows the squared correlation coefficient value and input variables for each model when the time horizon is twenty-four. A similar trend compared to the twelve hours' time horizon can be observed. As the KRR model generates the highest R^2 value.

Table 5: Models Accuracy- Time Horizon = 24

	Input Variables	R^2
KRR	Wind speed, direction, pressure, and humidity	0.4363
Persistent	Wind speed	0.1809
LS	Wind speed	0.0238

The MSE, MAPE, RMSE, and MAE values of each hour is calculated for each model. The lower these values are, the more accurate the model is. Table 6 shows the average of all the 720 points using different models. As mentioned earlier, the MAPE average is not calculated because some values are zero. The KRR generates the lowest values of MSE, RMSE, and MAE. The persistent model generates better results compared to the LS model according to the given KPIs. In figure 3, the trend of these KPIs (the average value per day) is observed.

Table 6: Average error measure over the entire tested interval - Time Horizon = 24

	MSE	*RMSE*	*MAE*
KRR	34.7	5.9	4.6
Persistent	59.0	7.7	5.6
LS	96.2	9.8	8.5

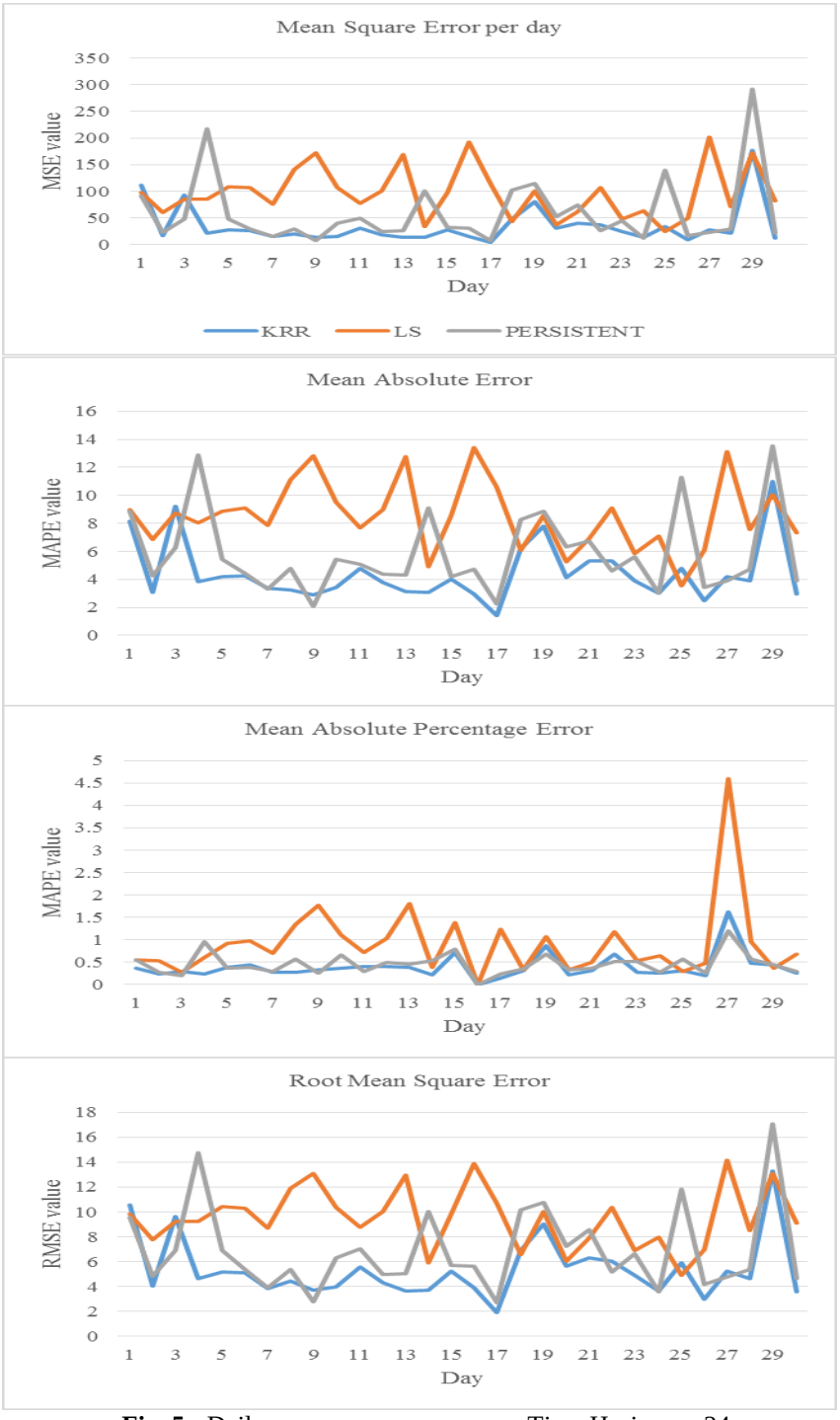


Fig. 5: Daily average error measure - Time Horizon = 24

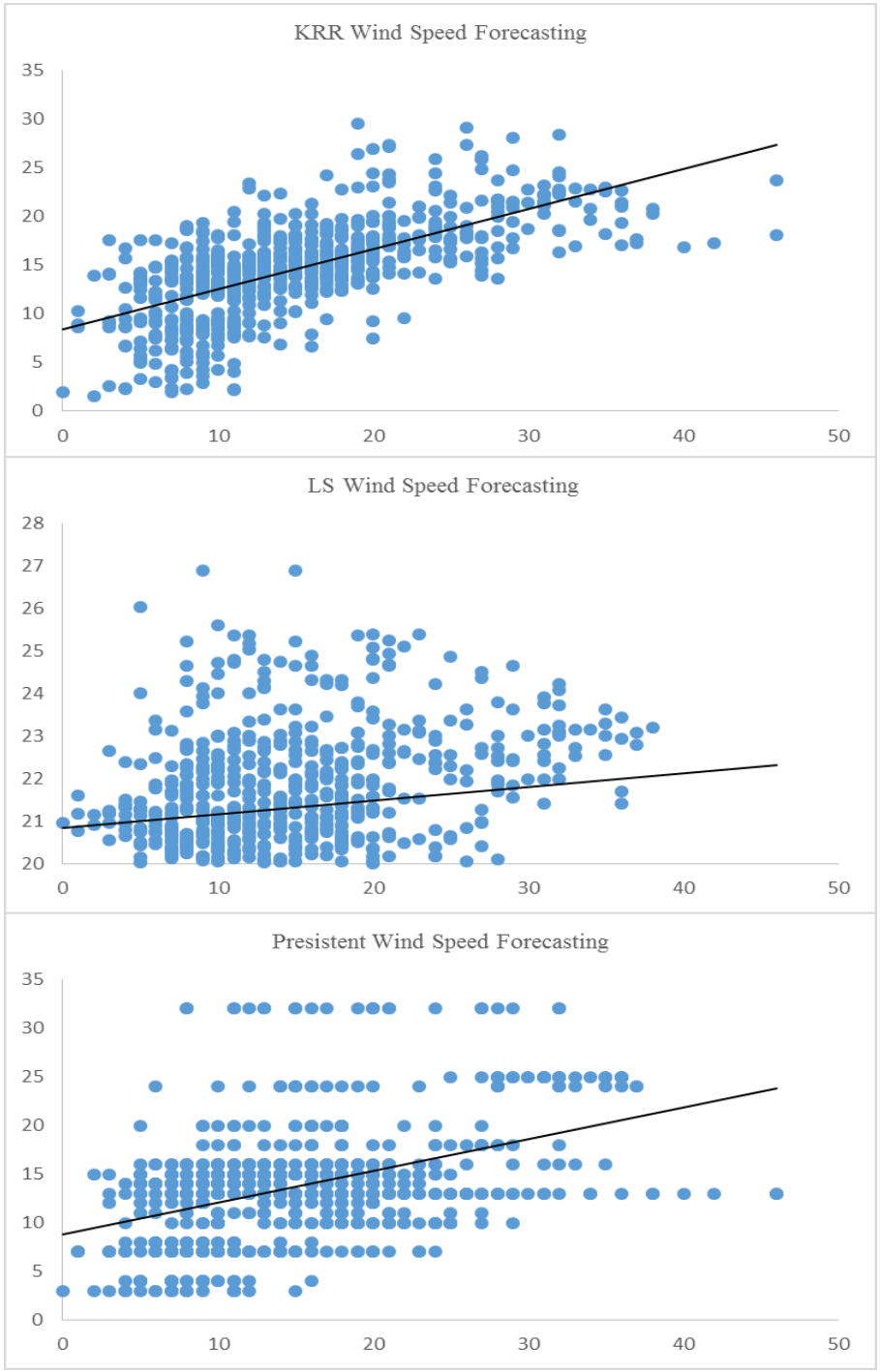


Fig. 6: Sample of the actual versus forecasted wind speed– time horizon = 24

CONCLUSION

To sum up, predicting wind speed is an important and challenging task in order to use wind power as a source of renewable energy. Wind speed data is highly variable and requires powerful methods for forecasting purposes. The KRR, LS and the persistence models were used to test the accuracy of wind speed forecasting. The MSE, RMSE, MAE, MAPE, and the R^2 value were set as key performance indicators (KPIs). The accuracy of the proposed forecasting methods is compared for three different time horizons (1 hour, 12 hours and 24 hours ahead). The KRR model generates the highest accuracy according to the KPIs, compared to the persistence and LS models in all the time horizons. Due to the limited data set, the training set consisted of a small number. Therefore, the KRR and LS models did not produce high R^2 value and a bigger data set would be required to generate accurate forecasts. With the increase of the lead time (horizon), the accuracy of each model decreases. Thus, KRR is one of the most powerful models that can be used for wind speed prediction.

ACKNOWLEDGEMENT

The author would like to thank Dr. Maher Maalouf and Dr. Tarek Elfouly of Khalifa University for all of their valuable input and comments.

REFERENCES

1. I. Sánchez, "Short-term prediction of wind energy production," International Journal of Forecasting, vol. 22, no. 1. pp. 43–56, 2006.
2. M. Lei, L. Shiyan, J. Chuanwen, L. Hongling, and Z. Yan, "A review on the forecasting of wind speed and generated power," Renewable Sustainable Energy Rev., vol. 13, no. 4, pp. 915–920, May 2009.
3. S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," North American Power Symposium 2010.
4. M. Maalouf and D. Homouz, "Kernel ridge regression using truncated newton method," Knowledge-Based Systems, vol. 71. pp. 339–344, 2014.
5. M. Maalouf and Z. Barsoum, "Failure strength prediction of aluminum spot-welded joints using kernel ridge regression," Int. J. Adv. Manuf. Technol., vol. 91, no. 9–12, pp. 3717–3725, Aug. 2017.

Evaluating the impact of solar and wind production uncertainty on prices using quantile regression

Mauro Bernardi*[1] and Francesco Lisi[2]

[1,2]Department of Statistical Sciences, University of Padova

June 27, 2019

In the past two decades the penetration and the importance of renewable energy sources (RES) has greatly increased. In turn, demand for accurate forecasts of solar and wind power, and their relations with energy demand and prices has increased too. In contrast with other conventional energy productions, photovoltaic and wind power generation is intrinsically stochastic and highly volatile, mainly due to weather conditions. As a consequence, energy traders, utilities and transmission systems operators, trading in the dispatching services markets, increasingly require reliable information concerning the effects of RES' uncertainty production on prices. Let $P_{t,h}$, $X_{i,t,h}$ and $PV_{t,h}$ be the (energy) price, the value of a exogenous variable $X_{i,t,h}$ for $i = 1, 2, \ldots, p-1$ and of the photovoltaic production on day t at hour h, respectively. In oder to measure the impact of RES uncertainty — in this case PV — on prices, we propose to use a (nonparametric) model which defines the relation between the price $P_{t,h}$ the set of $p-1$ explanatory variables and the photovoltaic production $PV_{t,h}$ (Lisi and Pelagatti, 2018), as follows:

$$P_{t,h} = f(X_{1,t,h}, \ldots, X_{p-1,t,h}) + g(PV_{t,h}) + \varepsilon_{t,h}, \tag{1}$$

where $f(\cdot)$ and $g(\cdot)$ are suitable nonparametric functions and $\varepsilon_{t,h}$ is an uncorrelated error term. The estimation of f and g allows us to obtain the predicted values at time $t+k$ and h, for any forecasting horizon $k > 0$:

$$\widehat{P}_{t+k,h} = \widehat{f}(X_{1,t+k,h}, \ldots, X_{p-1,t+k,h}) + \widehat{g}(PV_{t+k,h}). \tag{2}$$

How does this prediction is affected by the intrinsic variability of PV due to weather conditions? To answer this question we consider the set of values

$$\widehat{P}^{(m)}_{t+k,h} = \widehat{f}(X_{1,t+k,h}, \ldots, X_{p-1,t+k,h}) + \widehat{g}\left(PV^{(m)}_{t+k,h}\right), \qquad m = 1, 2, \ldots, M, \tag{3}$$

obtained considering always the same values of $X_{1,t+k,h}, \ldots, X_{p-1,t+k,h}$ and M suitable different values of $PV^{(m)}_{t+k,h}$. This allows us to define how much price can change

*Corresponding author: Via C. Battisti, 241, 35121 Padua, Italy. e-mail: mauro.bernardi@unipd.it, Phone.: +39.049.8274165.

at time $(t+k,h)$ due to different possible weather conditions.

Values of $PV_{t+k,h}^{(m)}$, for $m=1,2,\ldots,M$, in turn, are obtained by means of an auxiliary (nonparametric) quantile regression model based on calendar variables and, possibly, on installed capacity. Specifically, let $\tau_1<\tau_2<\cdots<\tau_S$ with $\tau_s\in(0,1)$ for $s=1,2,\ldots,S$, a sequence of quantile levels, for each $\tau_s\in(0,1)$, we assume a generalised additive regression model (Hastie and Tibshirani, 1986) for the conditional quantile of the response variable $PV_{t+k,h}$, defined as follows:

$$\mathcal{Q}_{\tau_s}\left(PV_{t,h}|X_{1,t,h},\ldots,X_{p-1,t,h}\right)=\sum_{j=1}^{J}f_j^s\left(X_{j,t,h}\right),\qquad s=1,2,\ldots,S,\tag{4}$$

where $Q_\tau\left(Y|\cdot\right)=\inf\left\{y\in\mathbb{R}:\widehat{F}_Y\left(y|\cdot\right)\geq\tau\right\}$ is the τ–th conditional quantile function of the response variable and $f_j^s\left(X_{j,t,h}\right)$ is a nonparametric continuous smooth function of the covariate $X_{j,t,h}$. In our model, the components $f_j^s\left(X_{j,t,h}\right)$ are approximated by regression splines. Quantile methods (Koenker, 2005) aim to estimate the regression parameters $\boldsymbol{\vartheta}^s\in\mathbb{R}^{p-1}$ without any assumption on the conditional distribution of PV, (see, e.g., Bernardi et al., 2018, for conditional quantile estimation of generalised additive models). The inversion of the estimated quantile function provides a nonparametric estimate of the conditional distribution of $PV_{t,h}$ at time (t,h), (see, e.g., Takeuchi et al., 2009).

This approach is applied on zonal data of the Italian electricity market (IPEX) for the period January 1, 2013 to March 31, 2019. In particular, for model (1) we use hourly time series of the zonal (NORTH) price as response while the considered regressors are the hourly power productions of photovoltaic, wind, hydro–river and hydro–thermal, the energy demand on the day–ahead Italian market, the price of gas on PSV, the installed capacities of photovoltaic and wind power productions and different time-related (or calendar) variables. Most of these variables are measured at zonal level.

References

Bernardi, M., Bottone, M., and Petrella, L. (2018). Bayesian quantile regression using the skew exponential power distribution. *Computational Statistics & Data Analysis*, 126:92 – 111.

Hastie, T. and Tibshirani, R. (1986). Generalized additive models. *Statistical Science*, 1(3):pp. 297–310.

Koenker, B. (2005). *Quantile Regression*. Cambridge University Press, Cambridge.

Lisi, F. and Pelagatti, M. M. (2018). Component estimation for electricity market data: Deterministic or stochastic? *Energy Economics*, 74:13 – 37.

Takeuchi, I., Nomura, K., and Kanamori, T. (2009). Nonparametric conditional density estimation using piecewise-linear solution path of kernel quantile regression. *Neural Computation*, 21(2):533–559. PMID: 19196229.

Interpretation of Kuwait Power System through ARIMA Model

S. Al-Osaimi and K. J. Sreekanth

Energy and Building Research Center, Kuwait Institute for Scientific Research, Kuwait

(sosaimi, sreekanthkj) @kisr.edu.kw

Abstract

Kuwait is considered one of the most important producing countries in the world whose economy depends heavily on the oil sector, by nearly 90%. The electricity production, transmission and distribution, and consumption for the state of Kuwait need a thorough understanding in order to identify the various factors that influence the power sector scenario. By analyzing power data and recognizing the effect of factors such as per capita income, the number of buildings, and annual production of electricity in the total primary energy supply of the state, the need for a comprehensive and sustainable power system in Kuwait can be forecasted. In this research, regression analyzes were used to determine the factors influencing between the various elements, so as to present the different future scenarios, using ARIMA, to predict energy consumption during the period from 2016 to 2030. The result shows that electricity consumption, and buildings have a higher influence than the GDP in 2030 based on the study.

Keywords: ARIMA, Regression Analysis, Forecast, and Energy.

1 Introduction

Kuwait is a small country with a small population that mainly depends on oil for the electricity industry. Over the years, energy efficiency projects have developed in the country. In addition, based on the vision of His Highness the Amir Sheikh Sabah Al-Ahmad Al-Sabah to allocate 15% of electricity from renewable sources, the need to establish the First Kuwait Energy Outlook comes. Initiating Kuwait Energy Outlook is important for all fields in the country in order to produce forecast scenarios. Whereas, energy consumption is an important economic index that represents the economic development of a city or a country (Sasan Barak, S. Saeedeh Sadegh, 2016). As (Sasan Barak, S. Saeedeh Sadegh, 2016) said to improve present and future energy supplies,

adfa, p. 1, 2011.

forecasting energy demands is essential. In order to make estimates and forecasts of the amount of energy sufficient to meet the needs of the community and identify the sources of this energy in the coming years. Also, it's important to develop plans and policies in the country to cover this need. Moreover, an accurate forecast is helpful for reasonable production plan arrangements and electricity policy developments (Wang, 2012). Furthermore, electricity forecasting could prove to be a useful policy tool for decision-makers. The first step in this project is to know the impact of various factors on total Primary Energy Supply in million tons of oil equivalents at the present time and prediction in the coming years. Electricity is not demanded its own sake; it is a derived demand that comes from the demand for lighting, heating, cooling, etc. Consequently, there are a number of exogenous factors that might influence electricity demand behavior (Zafer Dilaver, Lester C. Hunt, 2011). Since the number of building, GDP, and electricity consumption are all important drivers of electricity demand. These factors include the number of buildings, Gross domestic product (GDP), electricity consumed, etc. Because forecasting future energy production of Kuwait, which depend on these factors, is taken into consideration in Kuwait Energy Outlook.

There are multiple models which can be used for forecasting energy such as time series, regression, econometric, decomposition, co-integration, ARIMA, artificial systems such as the Artificial Neural Network (ANN), Grey prediction, Input-output, Fuzzy-logic, and the bottom-up models (Suganthi, L. and A.A. Samuel, 2012).In order to avoid a spurious or invalid forecast, ARIMA is a recommended approach since it is widely established (Ho, S.L. and M. Xie, 1998). ARIMA model is widely used in electricity demand analysis and is a high-precision approach for data forecasting (Wang, 2012). In this paper, simple Regression analysis and Autoregressive Integrated Moving Average (ARIMA) will be used. ARIMA has resulted in great achievements in both academic research and industrial applications during the last three decades (Chen and Wang, 2007). It is one of the most popular models for time series forecasting analysis, it has been originated from the autoregressive model (AR), the moving average model (MA) and the combination of the AR and MA, the ARMA models (Blanchard and Desrochers, 1984; Brown et al., 1984; Kamal and Jafri, 1997; Ho and Xie, 1998; Saab et al., 2001; Zhang, 2001; Ho et al., 2002).

2 Methodology

Different data were collected from multiple recourses such as Public Authority for Civil Information, World Bank, International Energy Agency, Central statistical Bureau, etc. Then by using excel three simple linear regression was done with one dependent factor which is Total Primary Energy Supply, and three independent factors. These independent factors are Electricity consumption, Number of building, and GPD. From regression result, checking if the value of R Square and Adjusted R Square are high and close to one. Then focus goes to P-value to be confidents it exceed 5%. As a validation step Hypothesis test was applied.

Selection of forecast method can be based on a few considerations such as availability of data, the time frame to perform the analysis, ease of method, forecast period and prior research. As mentioned before Autoregressive integrated moving average was selected to forecast. To start applying the ARIMA model, first autocorrelation (acf) and partial autocorrelation (pacf) functions should be determined. The ARIMA model can be used when the time series is stationary and there is no missing data in the within the time series (Volkan S- . Ediger, Sertac- Akar, 2006). At this stage is the process to ensure the data series is stationary. Stationary is essential in ARIMA forecasting model, data should be often stationary. Differencing is usually applied to data to remove the trend of data and stabilize the variance. Mann-Kendall test in Past3 software is used to check the stationary of the data. After that, the requested value for ARIMA such as d,p,q is defined. Graph one shows the methodology of the study. The final step is forecasting to 2030.

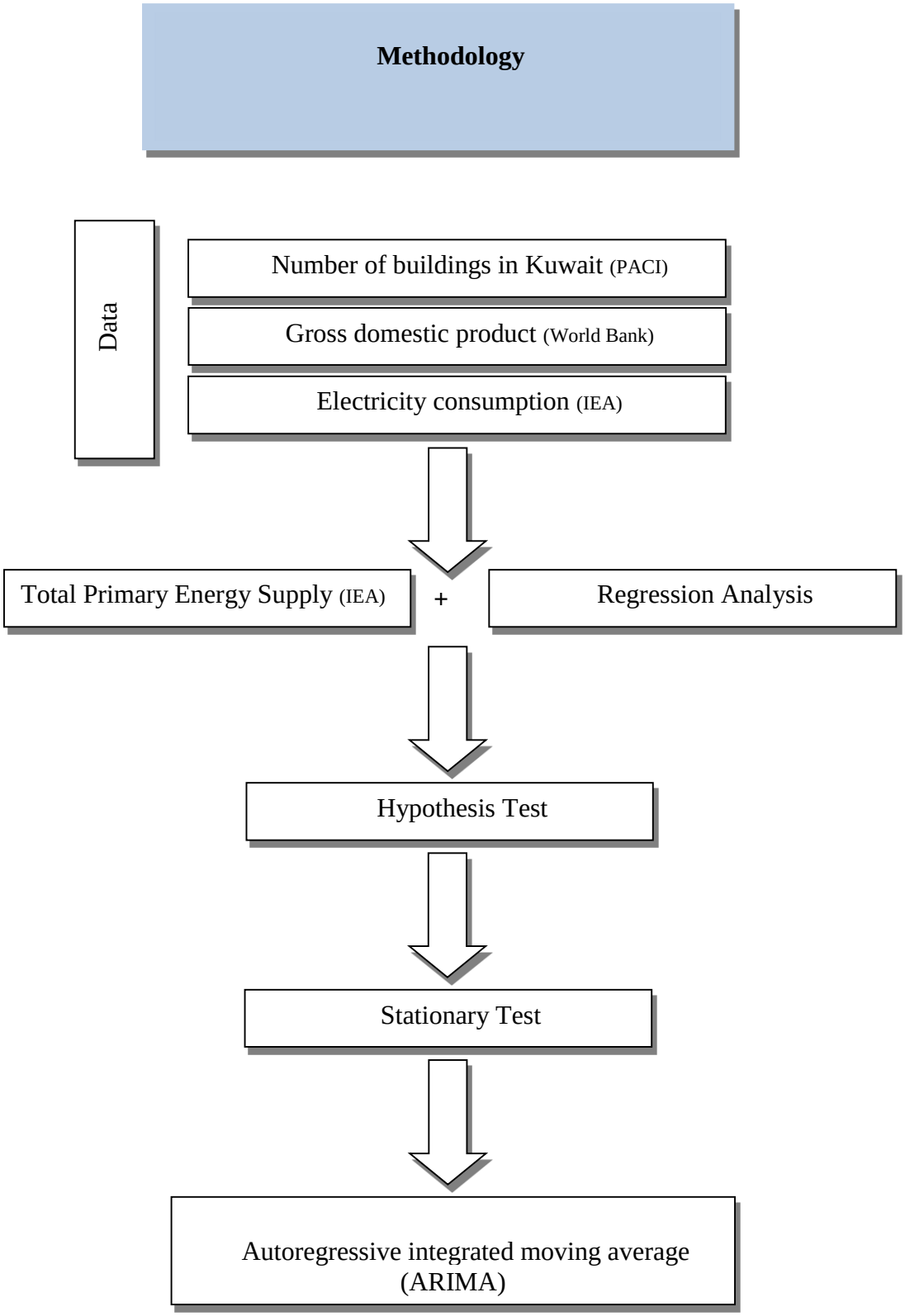


Figure (1). Methodology

3 Data and Analysis

Since there is a limitation of information, the available data for Primary Energy Supply and Electricity consumption are only from the nineties to 2016 as shown in the following tables. Lack of data is a critical problem in forecasting (Sasan Barak, S. Saeedeh Sadegh, 2016).

Table (1). Total primary energy supply

Year	Total primary energy supply (Mtoe)
1995	14.79
1996	14.55
1997	14.7
1998	16.79
1999	18.02
2000	18.72
2001	19.98
2002	20.54
2003	21.9
2004	23.37
2005	26.28
2006	25.67
2007	26.27
2008	28.72
2009	31.41
2010	32.09
2011	31.02
2012	34.33
2013	33.97
2014	31.63
2015	33.67
2016	35.84

Table (2). Electricity consumption

Year	Electricity consumption (TWh)
1995	21.11
1996	22.67
1997	23.78
1998	26.69
1999	28.1
2000	28.77
2001	30.53
2002	32.36
2003	35.42
2004	36.72
2005	38.77
2006	42.17
2007	42.8
2008	45.24
2009	46.6
2010	50.14
2011	50.38
2012	53.76
2013	53.58
2014	57.54
2015	58.56
2016	61.93

On the other hand, the data of the number of building in Kuwait were available in Public Authority for Civil Information (PACI) website from 1993 to 2018 as in figure 2. In addition, figure 3 shows the available data for the Gross domestic product (GDP) in the World Bank from 1990 to 2018.

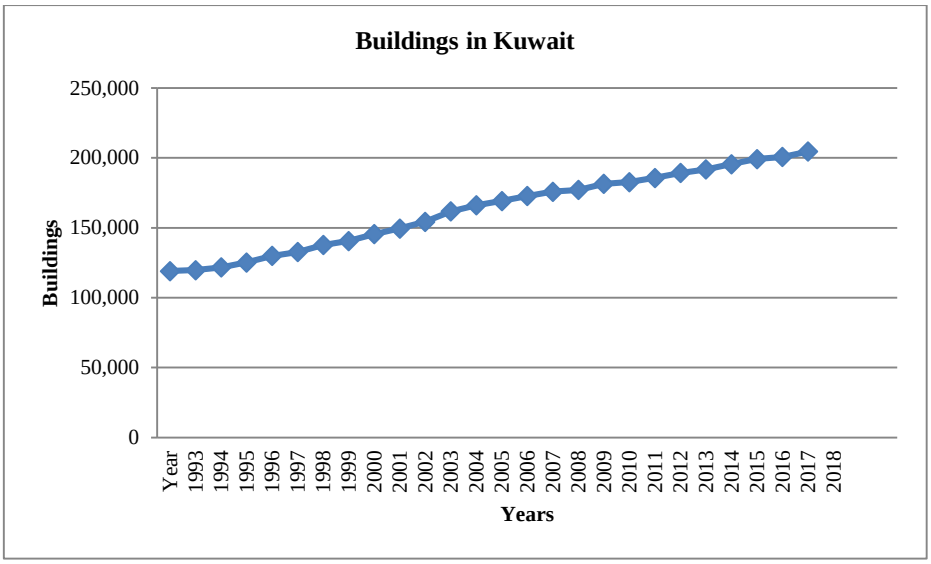


Figure (2). Number of buildings in Kuwait

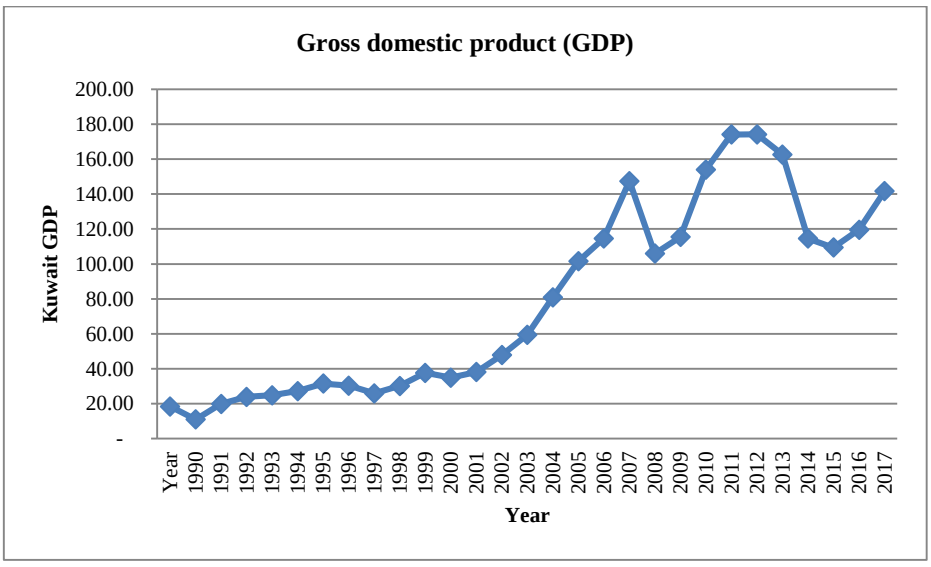


Figure (3). Kuwait Gross domestic product

After collecting all the required data, the period of the study is defined from 1995 to 2016. Then based on regression analysis, the data were divided into three groups:

Group (1):

Dependent Variable (DV)	Total Primary Energy Supply (Mtoe)
Independent Variable (IV)	Electricity consumption (Twh)

Group (2):

Dependent Variable (DV)	Total Primary Energy Supply (Mtoe)
Independent Variable (IV)	Number of buildings

Group (3):

Dependent Variable (DV)	Total Primary Energy Supply (Mtoe)
Independent Variable (IV)	Gross domestic product (GDP)

The following tables represent the result of simple linear regression for each group.

Table (3). Regression Analysis of Group 1

Regression Statistics		
R Square	0.965491	
Adjusted R Square	0.963766	
	Coefficients	P-value
Intercept	2.65677968	0.01481256
Electricity consumption (Twh)	0.55858458	4.29E-16

Table (4). Regression Analysis of Group 2

Regression Statistics		
R Square	0.9712264	
Adjusted R Square	0.9697878	
	Coefficients	P-value
Intercept	-22.0734867	1.35E-10
Electricity consumption (Twh)	0.00028999	6.95E-17

Table (5). Regression Analysis of Group 3

Regression Statistics		
R Square	0.8093383	
Adjusted R Square	0.7998052	
	Coefficients	P-value
Intercept	14.5909411	7.31E-10
Electricity consumption (Twh)	0.1216322	1.23E-08

After checking that all P values of all groups are less than 5%. Then Hypothesis test is applied. Moreover, the result of the test showed in table 6, 7, 8 for group 1, 2, and 3 respectively.

Table (6). Hypothesis test for Group 1

	Intercept	Building
H0 : $\beta 0 = 0$ & H1 : $\beta 0 \neq 0$	0.014812557 < 5%	-
H0 : $\beta 1 = 0$ & H1 : $\beta 1 \neq 0$	-	4.28742E-16 < 5%
Result	we reject the Null Hypothesis	

Table (7). Hypothesis test for Group 2

	Intercept	Building
H0 : $\beta 0 = 0$ & H1 : $\beta 0 \neq 0$	1.34742350786709E-10< 5%	-
H0 : $\beta 1 = 0$ & H1 : $\beta 1 \neq 0$	-	6.94517886082449E-17 < 5%
Result	we reject the Null Hypothesis	

Table (8). Hypothesis test for Group 3

	Intercept	Building
H0 : $\beta 0 = 0$ & H1 : $\beta 0 \neq 0$	7.31005317017345E-10 < 5%	-

H0 : β1 = 0 & H1 : β1 ≠ 0	-	1.23033700603841E-08 < 5%
Result	we reject the Null Hypothesis	

After the testing, the forecasting equation for group 1, 2, and 3 shown in the following equations:

Y = 2.65677968 + 0.558584582 X

Y = -22.07348670 + 0.0002899912 X

Y = 14.59094110 + 0. 121632197 X

In this paper Mann-Kendall test in Past3 software is used to check the stationary of the data. The test on the three series shows that all data are stationary after the first difference as shown in figure 4.

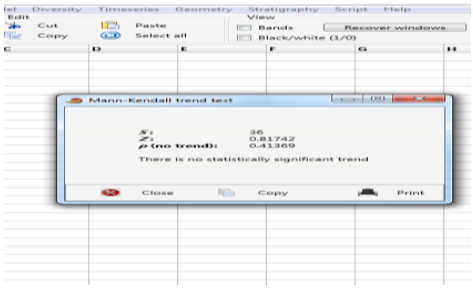


Figure (4). Result of **the** Mann-Kendall test

Statistical Package for the Social Sciences (**SPSS**) is used to conduct ACF and PACF graphs for each group. Based on the Box and Jenkins number of lags equal 6 in all groups for this study.

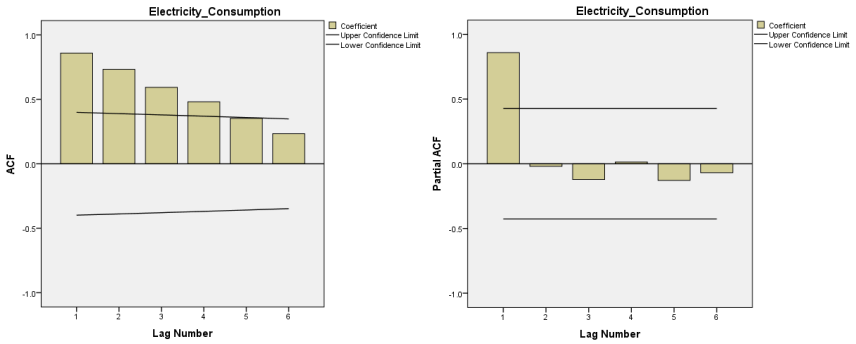


Figure (5). Autocorrelation Function and Partial Autocorrelation Function graph for group1

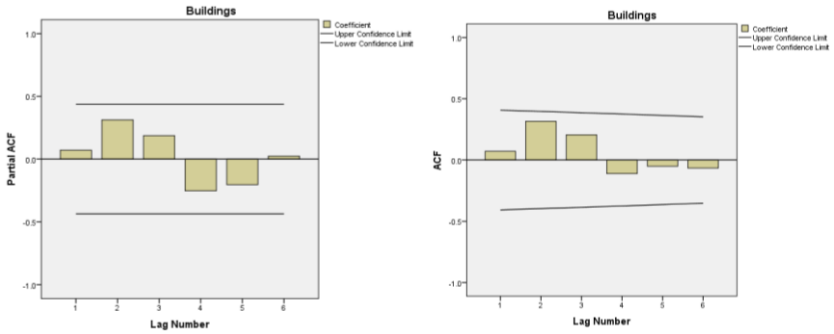


Figure (6). Autocorrelation Function and Partial Autocorrelation Function graph for group2

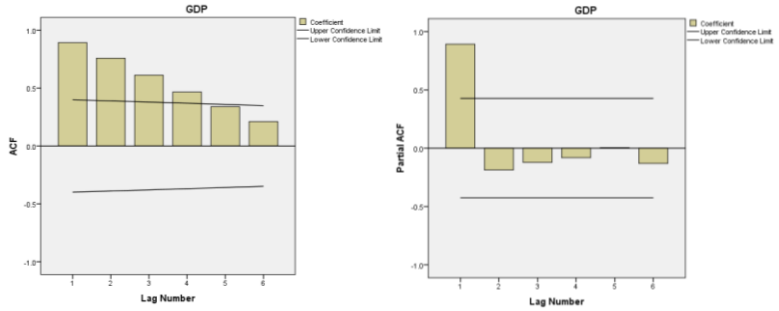


Figure (7). Autocorrelation Function and Partial Autocorrelation Function graph for group3

Based on these graphs P, d, and q are defined for each group. Then Xlstat is used for ARIMA forecast. The following charts represent the ARIMA output for each group.

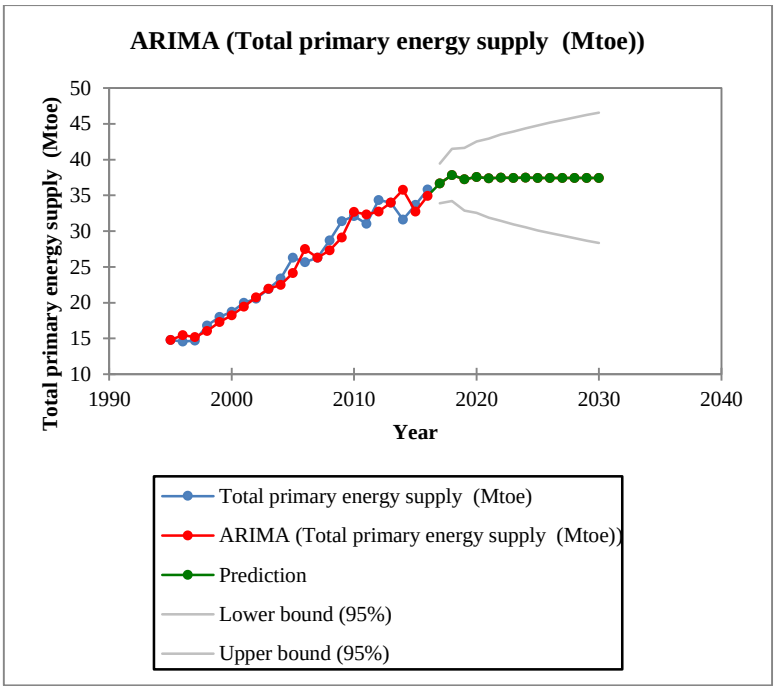


Figure (8). ARIMA output for group 1

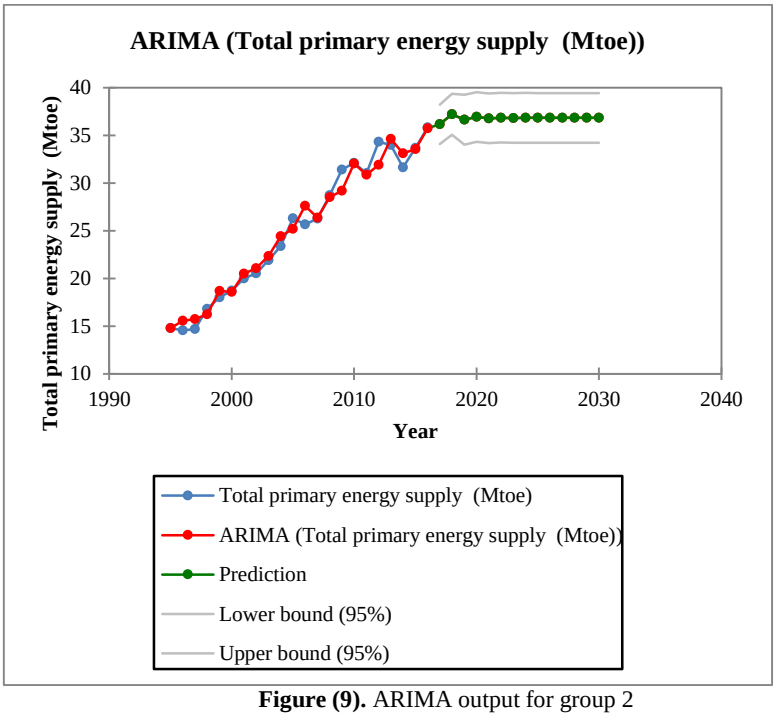


Figure (9). ARIMA output for group 2

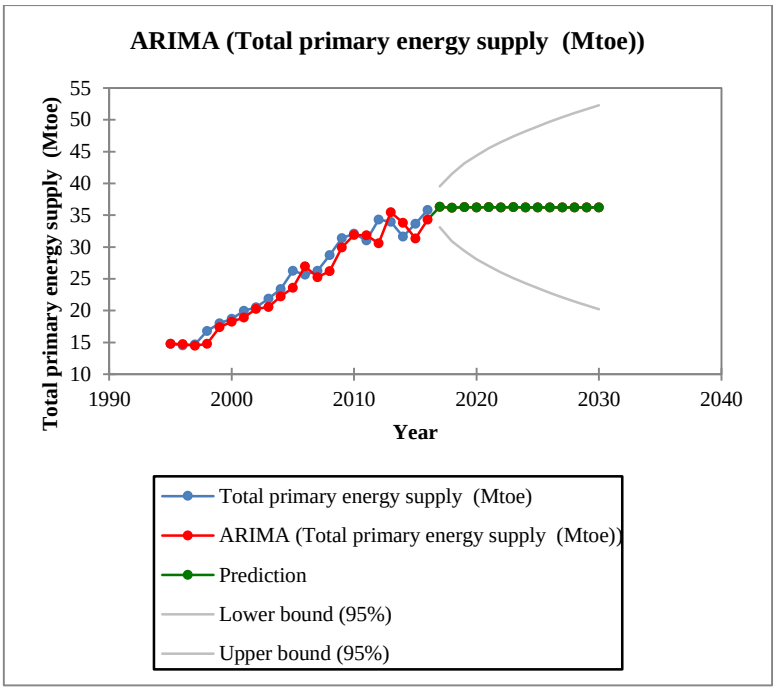


Figure (10). ARIMA output for group 3

4 Conclusions and Recommendations

Energy forecasting is difficult because it is affected by the rapid development of economy, technology, government decisions, and other factors. A major goal of the study assesses the effects of cretin factors (Electricity consumption, Number of buildings in Kuwait, and Kuwait GDP) on the total Kuwait primary supply. From the ARIMA graphs, it is evident that, the total primary energy and supply, is equal to 37.83, 37.20, and 36.19 Mtoe corresponding to 2018, and the associated electricity consumption is 66.57 Twh, Number of buildings is 204,623, and GDP is 141.678. Hence, it can be seen that the rapid increase in electricity consumption and buildings reflect on higher effect on the total primary energy and supply.This lead to shifting the attention and focus on these two factors in the event of decisions. For future research, consider other factors such as population, average income per capita, consumer behavior, and many other factors that may affect the primary energy supply.

5 References

1. Yuanyuan Wang, JianzhouWang, GeZhao, YaoDong, 2012. Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China. 284–294.
2. Volkan S-. Ediger, Sertac- Akar. 2006. ARIMA forecasting of primary energy demand by fuel in Turkey. 1701–1708.
3. Chaoqing Yuan , Sifeng Liu, Zhigeng Fang, 2016. Comparison of China's primary energy consumption forecasting by using ARIMA (the autoregressive integrated moving average) model and GM (1, 1) model. 384e390.
4. Rina Haigesa=, Y.D.Wang, A. Ghoshray, A.P. Roskilly, 2017. Forecasting electricity generation capacity in Malaysia: An Auto-Regressive Integrated Moving Average approach. 3471 – 3478.
5. Sasan Barak, S. Saeedeh Sadegh, 2016. Forecasting energy consumption using ensemble ARIMA–ANFIS hybrid algorithm. 9 2–104.
6. Zafer Dilaver, Lester C. Hunt, 2011. Turkish aggregate electricity demand: An outlook to 2020. 6686e6696.
7. Public Authority for Civil Information, https://www.paci.gov.kw/Default.aspx.
8. Central Statistical Bureau, https://www.csb.gov.kw/Default_EN.
9. World Bank, https://www.worldbank.org/.
10. International Energy Agency, https://www.iea.org/.

Estimating the Unknown Parameters of a Chaos-Based S-Box from Time Series

Salih Ergün

TÜBİTAK-Informatics and Information Security Research Center
PO Box 74, 41470, Gebze, Kocaeli, Turkey
salih.ergun@tubitak.gov.tr

Abstract. A novel estimate system is proposed to discover the security weaknesses of a chaos based S-Box and a chaos based random number generator (RNG) it contains. Convergence of the estimate system is proved using auto-synchronization. Secret parameter of the target chaos based RNG are recovered where the available information are the structure of the chaos based S-Box and a scalar time series observed from the target chaotic system. Simulation and numerical results verifying the feasibility of the estimate system are given such that, next bit can be predicted while the same output bit sequence of the chaos based RNG used for S-Box generation can be regenerated.

Keywords: Estimation, security analysis, random number generator, continuous-time chaos, time series, synchronization of chaotic systems, auto-synchronization

1 Introduction

Nowadays, technological developments emphasize the importance of innovations in the following field of circuits and systems: Small area occupation, hardware security, low power consumption and high speed operation. In relation to this, the fast and low power consuming random number generators (RNG) are positioned more clearly in the heart of the research as the main components of the security systems [1, 2]. Although most of the people are unaware that they use them, we use RNGs in our day-to-day work. We use RNG if you withdraw money from a bank's cash machine, order goods with a credit card on the internet, or watch pay TV. Public/private keys for asymmetric crypto algorithms, keys for hybrid and symmetric encryption systems, one-time pad, nonces and padding bytes are created by using RNGs [3].

Being aware of any knowledge about the structure of the RNG must not provide a useful estimate of the output bit sequence of the RNG. Even so, fulfilling the requirements for the confidentiality of security systems using RNG requires three privacy criteria as a must: 1. The RNG must fulfill all statistical randomness tests; 2. The preceding and following random bits can not be predicted [4] and; 3. Anyone should not generate the same output bit sequence of the RNG [5].

One of the basic principle of the cryptography is that according to Kerckho 's hypothesis [3], it is assumed that the overall security of any crypto system is completely dependent on the security of the key, and that all other parameters of the crypto system are publicly observable. Vulnerability analysis is complementary to cryptography. The interaction between these two cryptology branches creates a contemporary cryptography that becomes stronger due to the vulnerability analysis that reveals the weaknesses of the existing crypto systems.

Although the use of discrete-time chaotic maps in chaos based security systems has been acknowledged over a long period of time [6 8], it has been shown nowadays that continuous-time chaotic oscillators can be used to implement chaos based security systems [9 11, 13, 14]. In particular, a chaos based S-Box and a chaos based RNG it contains have been proposed in [11]. In this article, we target the chaos based RNG reported in [11], and propose an estimate system to analyze security vulnerabilities of the targeted chaos based S-Box.

The robustness of a crypto system depends on the key used, or in other words, the attacker's ability to estimate the key. The target RNG [11] defines the deterministic chaos as the true source of randomness, contrary to the RNG design reported in [13] in which the equivalent noise generated by circuit components is analyzed.

The organization of the article is as follows. In Section 2 the targeted S-Box and the RNG it contains are explained in detail; In the next Section 3 an estimate system is proposed for vulnerability analysis of the target S-Box and its feasibility is verified; Section 4 describes the numerical and simulation results that are followed by the conclusion section.

2 Target System

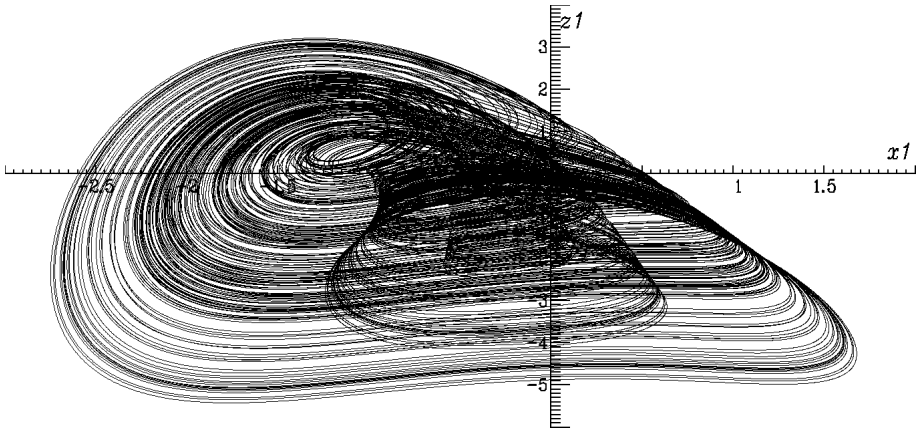


Fig. 1. Chaotic attractor obtained for $a_1 = 1$ using a 4^{th}-order Runge-Kutta algorithm.

Chaotic systems can be categorized into two groups: In relation to the evolution of underlying dynamical system, one is discrete time and the other is continuous time.

In the target paper [11], a simple autonomous continuous-time chaotic system is utilized, as the seed of the S-Box and the RNG, which is derived from a simple model. The analysis of the system yields the state equations given in [11] which transforms into the following equation:

$$\begin{aligned} x_1 &= 2y_1 - x_1 - z_1 \\ y_1 &= x_1 z_1 - x_1 y_1 - x_1 \\ z_1 &= -3x_1 y_1 + a_1 \end{aligned} \tag{1}$$

The equations in 1 generate chaos for the single-parameter a_1 in a large region ($0.5 < a_1 < 2.5$). The chaotic attractor given in Fig.1 is obtained for $a_1 = 1$ using a 4^{th}-order Runge-Kutta algorithm with an adaptive step size.

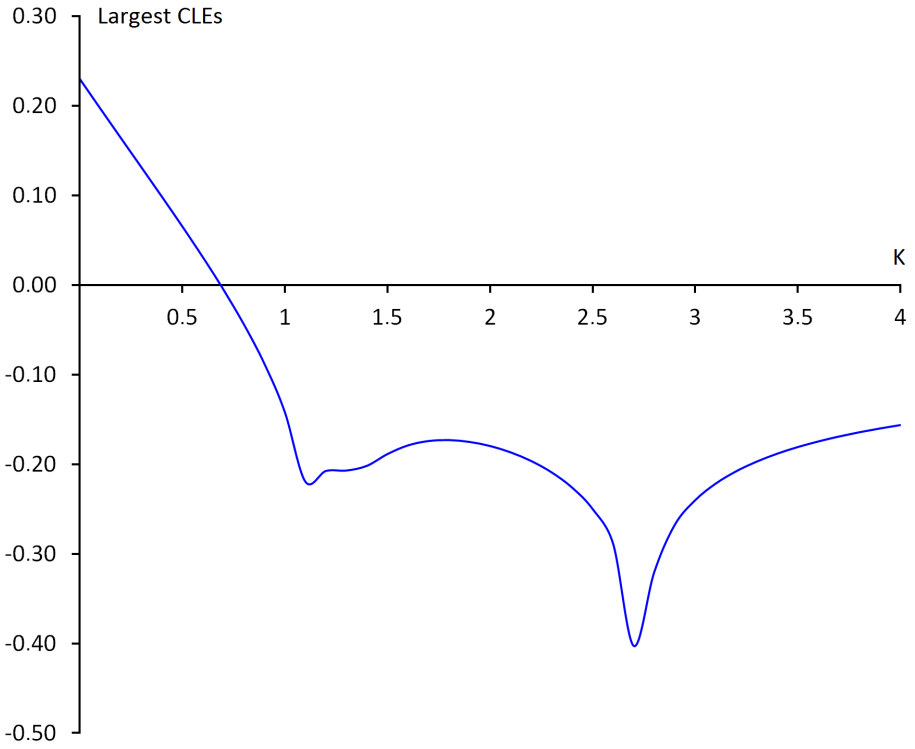


Fig. 2. Largest Conditional Lyapunov Exponents as a function of coupling strength K.

Random number generation method is explained in [11] where the mechanism is fundamentally based on solving the chaotic system using a numerical

algorithm. In [11], the core chaotic system is initially solved using a 4^{th}-order Runge-Kutta algorithm with an appropriate step size Δh and the time series of x,y and z are obtained. Then floating point numbers of x,y and z are converted to $32-bits$ binary numbers and least significant $2-bits$ are selected from these $32-bits$ binary numbers to create candidate random numbers. Finally candidate random numbers are subjected to NIST-800-22 test suite [15] to generate S_x, S_y and S_z output bit sequences if they pass all the test suite and otherwise they are discarded.

The chaos based S-Box design is also explained by [11] in detail. Random bit sequences S_x and S_z are used for S-Box generation. $8-bits$ of S_x and S_z are put through XOR operation to generate $8-bits$ of decimal values. If the generated decimal value exists in the S-Box then it is discarded otherwise it is used in the S-Box as a new component until unique 256 values are placed on S-Box. On the other hand, S_y bit sequence is used in the encryption-decryption processes. These processes are carried out by applying the bitwise XOR operation between the plain-cipher image and S_y random bit sequence, respectively.

The NIST-800-22 statistical tests [15] were preferred in [11] to analyze output randomness of the chaos-based RNG design. However, Big Crush [16] and Diehard [17] statistical test suites weren't applied to output bit stream of the RNG. Note that, the target RNG [11] do not fulfill the first secrecy criteria, which states that TRNG must pass all the statistical tests of randomness.

3 Estimate System

Since the ground-breaking paper of Pecora and Carroll, the synchronization of chaotic systems has become an increasingly sought-after field of research [18]. In this article, the convergence of the estimate and target systems is proven using the auto-synchronization, (synchronization of chaotic systems with unknown parameters) [19]. In order to analyze vulnerability of the target RNG, an estimate system given by the following equation 2 is proposed:

$$\begin{aligned} x_2 &= 2y_2 - x_2 - z_2 \\ y_2 &= x_2 z_2 - x_2 y_2 - x_2 \\ z_2 &= -3x_2 y_2 + a_2 + K(z_1 - z_2) \\ a_2 &= z1 - z2 \end{aligned} \tag{2}$$

where K is the coupling strength between the target and estimate systems and a_2 is the unknown control parameter of the target system to be estimated. The information available are the structure of the target RNG system and a scalar time series given by a observable where z_1 is the observable chaotic signal given in 2.

For analyzing the stability of auto-synchronization, we numerically calculate the Conditional Lyapunov Exponents (CLE) using standard 4th-order Runge-Kutta algorithm with fixed step size. CLEs for the estimate system are calculated from the set of ordinary di erential equations given in Eqn. 2 where QR decomposition method [20] is used. Numerical Jacobian is exploited which is calculated

numerically by using finite di erences. O set for numerical Jacobian = $10^{-0.008}$ and integration time step is 0.004 while integration steps per Jacobian map is 50.

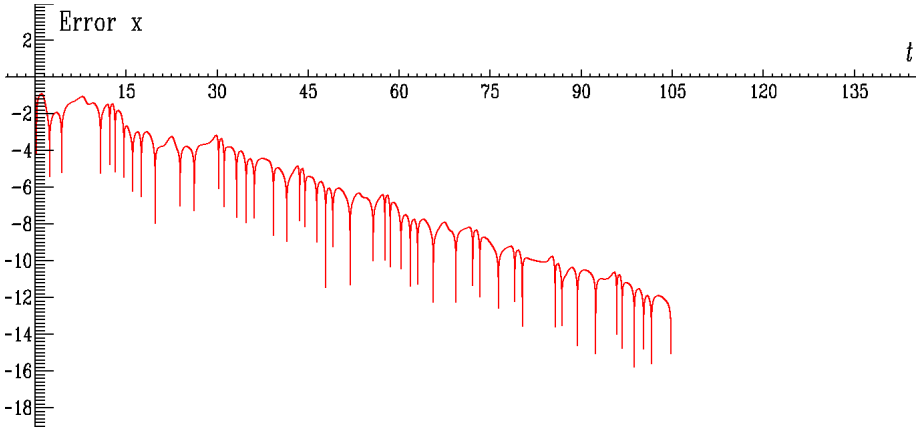


Fig. 3. Synchronization error Log $|e_x(t)|$ (red line).

In Fig.2, we plot largest CLEs as a function of coupling strength K. As shown in this figure, when K is greater than 0.68 then the largest CLE is negative and hence auto-synchronization of target and estimate systems is achieved and stable. For any K less than 0.68, largest CLE is positive and auto-synchronization is unstable.

4 Numerical Results

In this article, we numerically demonstrate the proposed estimate system using standard 4th-order Runge-Kutta algorithm with fixed step size. The estimate system expressed by the Eqn. 2 is designed that converges to target system as $x_2 \to x_1$ where $t \to \infty$ and t is the normalized time. Error signal a, x, y, and z of the auto-synchronization are defined as $e_a = a_1 - a_2$, $e_x = x_1 - x_2$, $e_y = y_1 - y_2$ and $e_z = z_1 - z_2$ respectively. Here proposed estimate system aims to find out appropriate coupling strengths such that $|e(t)| \to 0$ when $t \to \infty$.

Log $|e_a(t)|$, Log $|e_x(t)|$, Log $|e_y(t)|$ and Log $|e_z(t)|$, are given as a function of normalized time t in Fig.6, Fig.3, Fig.4 and Fig.5 respectively, for $K = 2.7$. It is observed from the given figure that the auto-synchronization is achieved in less than $110t$, where the synchronization e ect is better than that of $K = 0.69$.

Auto-synchronization of the estimate system is shown in Fig.7 where the convergence of the recovered parameter values a2 of the estimate systems to the known values a1 of the target system is illustrated. As shown from the given figure that, the proposed estimate system converges to the parameter a1 of the

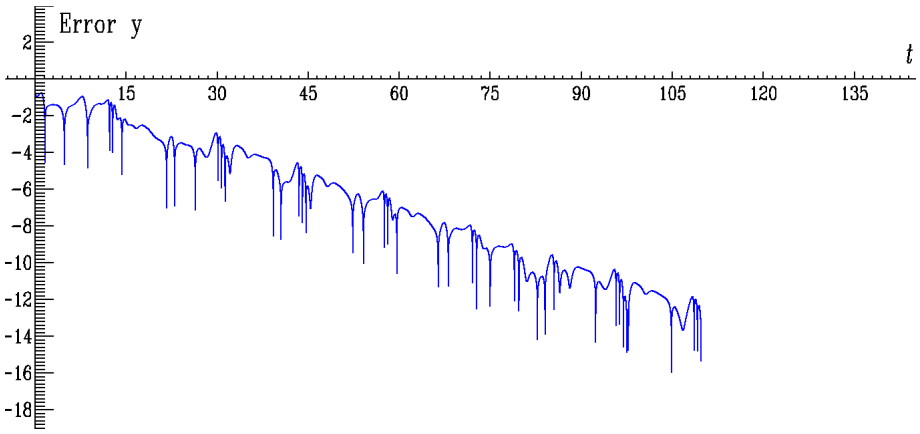


Fig. 4. Synchronization error Log $|e_y(t)|$ (blue line).

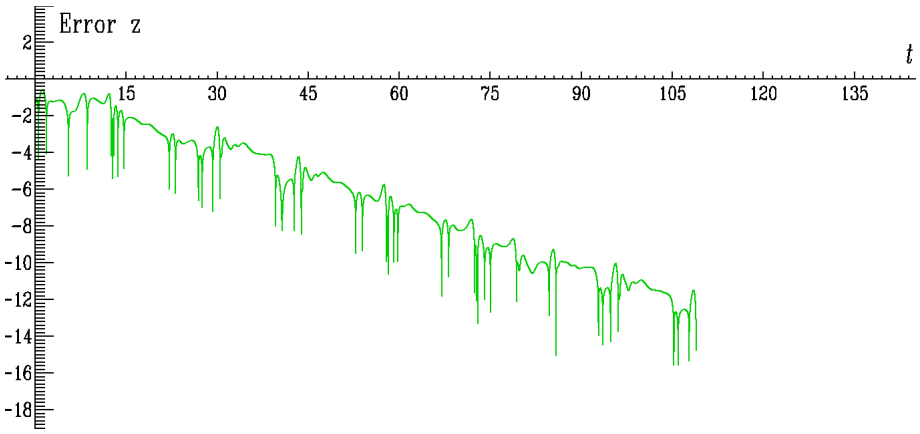


Fig. 5. Synchronization error Log $|e_z(t)|$ (green line).

target system for $0.7 < a_1 < 1.3$ and auto-synchronization is achieved in less than $110t$.

On the other hand, the other cryptanalysis results have been reported in [21, 22] where these papers use master slave synchronization scheme. In this work, we propose a novel chaotic system and further focus on estimating the secret parameters from time series where auto-synchronization scheme is used.

Simulation results of $x_1 - x_2$, $y_1 - y_2$ and $z_1 - z_2$, are depicted in Fig. 8, Fig. 9 and Fig. 10, which show non-synchronized behavior and synchronization of target and estimate systems.

From the figures it is observed that stable identical synchronization can be achieved. In these figures, a synchronized phenomenon has not been observed initially as shown by the black lines. The proposed estimate system approaches the target system in less than $110t$ and the stable identical synchronization is

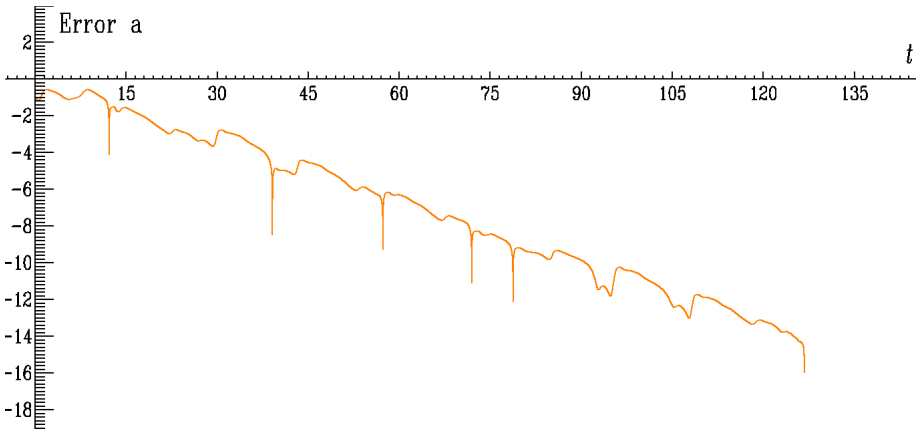


Fig. 6. Synchronization error Log $|e_a(t)|$ (orange line).

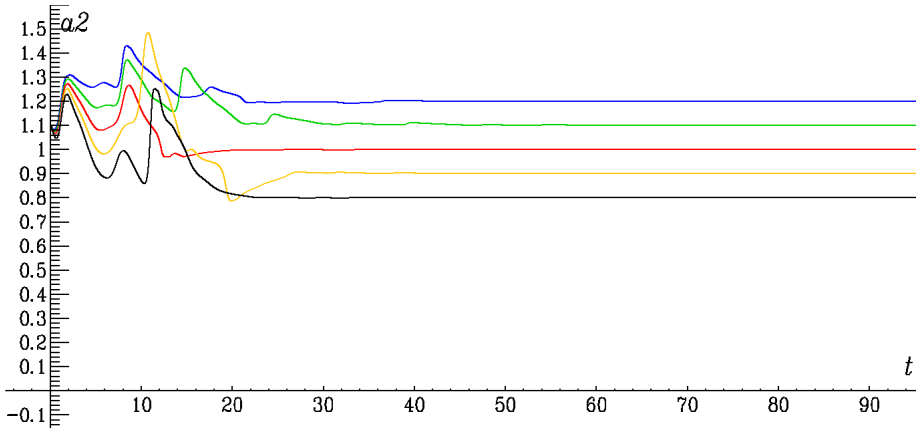


Fig. 7. Convergence of the parameter value a_2 of the estimate system to the fixed value a_1 of the target system for $0.7 < a_1 < 1.3$.

obtained where these synchronized phenomenon are shown by colored lines in Fig. 8, 9 and Fig. 10, respectively.

Since the identical synchronization of estimate and target systems is achieved in less than $110t$ ($x_2 \rightarrow x_1$), the unknown parameters of the target random number generation system are accurately recovered and the estimated values of x_1, y_1, z_1, and S_x, S_y, S_z bit sequences converge to their corresponding fixed values. As a result, it is clear that chaotic systems have achieved the identical synchronization and therefore the output bitstreams of the target-estimate systems and S-Box values are completely synchronized.

As a result, the proposed estimate system has not only reached the identical synchronization at the level of the chaotic state variables but also synchronized at the level of the generated bit sequence. Proposed system not only estimates the

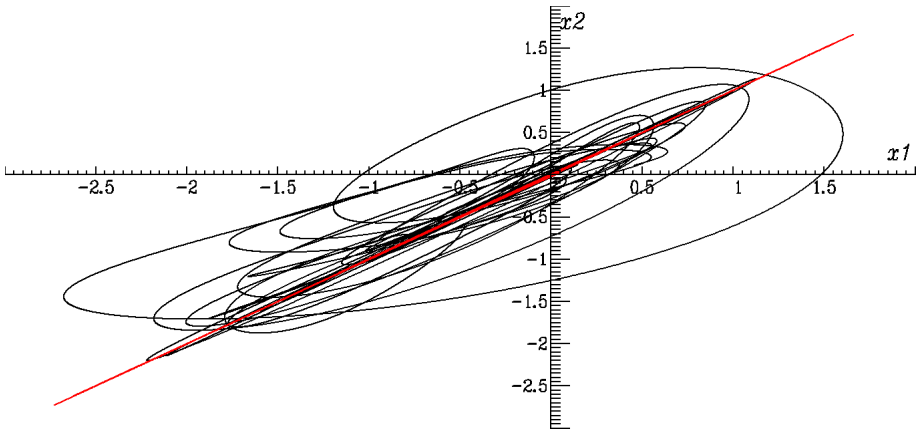


Fig. 8. Numerical results of $x_1 - x_2$ showing the synchronized and unsynchronized behaviors of target and estimate systems.

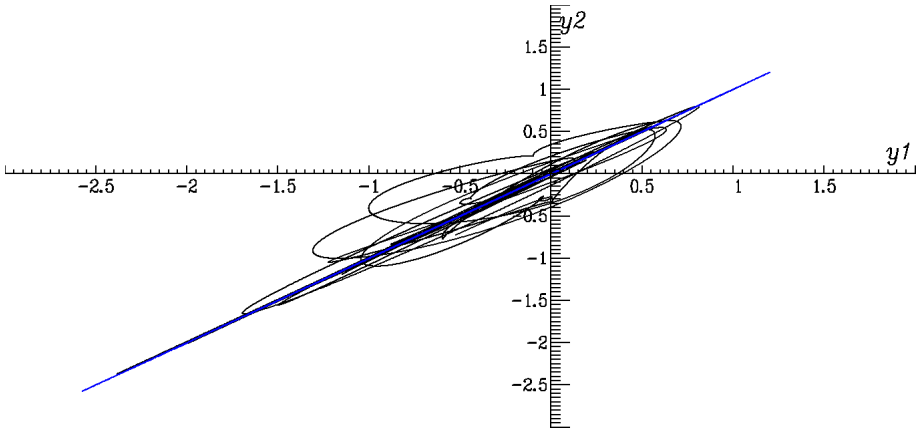


Fig. 9. Numerical results of $y_1 - y_2$ showing the synchronized and unsynchronized behaviors of target and estimate systems.

preceding and following bits of the target RNG and S-Box but also shows that the estimate system can generate the same output bit sequence of the target RNG and S-Box. The target RNG [11] satisfies neither the second nor third secrecy criteria that an RNG must fulfill. In conclusion, it has been verified that deterministic chaos can not be the true source of randomness.

5 Conclusions

In this paper, an estimate system is proposed to discover the security weaknesses of a chaos based S-Box and a chaos based random number generator (RNG) it contains. It is shown that secret parameters of the RNG can be recovered by

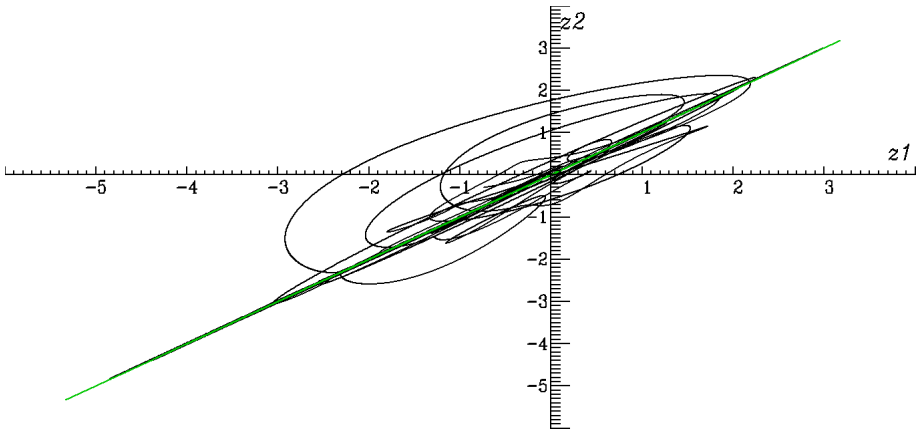


Fig. 10. Numerical results of $z_1 - z_2$ showing the synchronized and unsynchronized behaviors of target and estimate systems.

the proposed estimate system using auto-synchronization scheme. Although the only information available is the structure of the chaos based S-Box and a scalar time series, auto-synchronization of the estimate system is achieved and hence not only next bit but also whole output bit sequences are synchronized.

References

1. Ferguson, N., Schneier, B., Kohno, T.: Cryptography engineering: design principles and practical applications, Wiley Publishing, Inc. (2011)
2. Ü. Güler, S. Ergün, "A high speed IC random number generator based on phase noise in ring oscillators," Proc. IEEE International Symposium on Circuits and Systems (ISCAS '10), 2010, pp. 425-428
3. Ü. Güler, S. Ergün G. and Dündar, "A digital IC random number generator with logic gates only," Proc. of IEEE International Conference on Electronics, Circuits, and Systems (ICECS), Dec. 2010, pp. 239-242.
4. N.C. Göv, M.K. Mıhçak, and S. Ergün, "True Random Number Generation Via Sampling From Flat Band-Limited Gaussian Processes," IEEE Trans. Circuits and Systems I, vol. 58, no. 5, pp. 1044-1051, May 2011.
5. Schneier, B.: Applied Cryptography: Protocols, Algorithms and Source Code in C. Wiley Publishing, Inc. (2015)
6. F. Pareschi, G. Setti, R. Rovatti, "Implementation and Testing of High-Speed CMOS True Random Number Generators Based on Chaotic Systems", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 57, 12 (2010) 3124-3137.
7. J. L. Valtierra, E. Tlelo-Cuautle, A. Rodrguez-Vzquez, "A switched-capacitor skew-tent map implementation for random number generation", International Journal of Circuit Theory and Applications, Vol. 45, 2 (2017) 305315.
8. M. Kim, U. Ha, K. J. Lee, Y. Lee, H.J. Yoo, "A 82-nW Chaotic Map True Random Number Generator Based on a Sub-Ranging SAR ADC", IEEE Journal of Solid-State Circuits, Vol. 52, 7 (2017) 1953-1965.

9. S. Özoğuz, S. Ergün, "A Non-Autonomous IC Chaotic Oscillator and Its Application for Random Bit Generation", Proc. European Conference on Circuit Theory and Design (ECCTD '05), Vol. 2 (2005) pp. 165-168.
10. S. Ergün, "Modeling and Analysis of Chaos-Modulated Dual Oscillator-Based Random Number Generators," Proc. European Signal Processing Conference (EUSIPCO '08) pp. 1-5, Aug. 2008.
11. Ü. Çavusoglu, S. Kaçar, İ. Pehlivan, A. Zengin, "Secure image encryption algorithm design using a novel chaos based S-Box," Chaos, Solitons and Fractals, vol. 95, pp. 92-101, 2017
12. S. Ergün, S. Özoğuz, "A Chaos-Modulated Dual Oscillator-Based Truly Random Number Generator," Proc. IEEE International Symposium on Circuits and Systems (ISCAS '07), pp. 2482-2485, May 2007.
13. S. Ergün, Ü. Güler, and K. Asada, "A High Speed IC Truly Random Number Generator Based on Chaotic Sampling of Regular Waveform" IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E94-A, no.1, pp.180-190, Jan. 2011
14. S. Ergün, "Random numbers generation using continuous-time chaos" US Patent, Patent No US 008738675, May. 2014
15. L. Bassham, A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, N. Heckert, J. Dray , "SP 800-22 Rev. 1a A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications", Apr. 2010, http://doi.org/10.6028/NIST.SP.800-22r1a Available at https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf
16. P. L'Ecuyer, Universit'e de Montr'eal., "Empirical Testing of Random Number Generators", 2002, Available at http://www. iro.umontreal.ca/lecuyer/
17. G. Marsalgia, "Diehard: A Battery of Tests of Randomness", 1997, Available at http://stat.fsu.edu/∼geo/diehard.htm
18. L.M. Pecora, T.L. Carroll, "Synchronization of chaotic systems," Chaos: An Interdisciplinary Journal of Nonlinear Science, vol. 25, no. 9, 097611 pp. 1-12, Apr. 2015. https://doi.org/10.1063/1.4917383
19. Y. Liu, W. Tang, and L. Kocarev, "An Adaptive Observer Design for Auto-Synchronization of Lorenz System," International Journal of Bifurcation and Chaos, vol. 18, no. 8, pp. 2415-2423, Aug. 2008.
20. J. P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," American Physical Society, Reviews of Modern Physics, vol. 57, no. 3, 1, (1985), pp. 617-656. https://doi.org/10.1103/RevModPhys.57.617
21. S. Ergün, "On the security of a double-scroll based "true" random bit generator," 23^{rd} European Signal Processing Conference (EUSIPCO), Nice, 2015, pp. 2058-2061 doi: 10.1109/EUSIPCO.2015.7362746
22. S. Ergün, "On the security of chaos based true random generators," IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 2016, 99, pp. 363369. https://doi.org/10.1587/transfun.E99.A.363

GNSS based Automatic Anchor Positioning in Real Time Localization Systems

Andreas Heller, Ludwig Horsthemke, Marcel Gebing, Götz Kappen, and Peter Glösekötter

Münster University of Applied Sciences, Steinfurt, Germany
andreas.heller@fh-muenster.de
ludwig.horsthemke@fh-muenster.de
marcel.gebing@fh-muenster.de
goetz.kappen@fh-muenster.de
peter.gloesekoetter@fh-muenster.de

Abstract. This work describes the setup of an ultrawideband (UWB) realtime localization system (RTLS) in which the determination of anchor coordinates is automated by assistance of global navigation satellite systems (GNSS).

Keywords: UWB · GPS · RTLS

1 Motivation

Localization using ultra-wideband (UWB) radio frequency (RF) transceivers has gained a lot of interest in the recent years. Different ranging schemes have been implemented and evaluated. Usually, the distance from a mobile node (tag) to nodes with known positions (anchors) are acquired to derive the tags position. All have in common that the anchor positions have to be known precisley for the resulting deduced tags position to be accurate. Common practise includes the measurement of distances from reference objects and surfaces like walls or manual triangulation using tape measures, laser range finders (accuracy 3 mm [12]) or ideally total station theodolites (accuracy 0.8 mm+1ppm [3]). In non-line-of-sight setups these measurements potentially have to performed several times for an anchor in which the errors add up.

The aim of this work is to automate this process using the ranging facilities the system offers anyway. As described later additional information is necessary for which relatively cheap but inaccurate GNSS receivers are used (Horizontal position accuracy 2.5 m [13]). The target domain for the presented real time localization system (RTLS) is the optimization of agitation processes in biogas plant digesters. As the agitation process plays an important role in its effectiveness, simulations are carried out for different mixing concepts, with the aim of maximizing the yield. The developed RTLS provides a tool to assess the flow of substrate in these plants and support the simulations. With anchor placement outside on top of the digesters, the advantage of the possibility of GNSS reception is given.

2 Related work

Shi et al. [11] and Gentner et al. [5] both proposed SLAM inspired algorithms which use inertial measurement units (IMU) connected to tags to determine the anchors positions. As the integration of IMUs is planned in a later stage these algorithms will be evaluated for this work.

The ambiguity flip problem as described by Moravek et al. [7] is solved in this work by determining the constellation of the anchor nodes using a GNSS. Pelka et al. [9] propose a way in which no such external reference is necessary. But they still face a comparatively large average Euclidean positioning error of 0.62 m in a real world scenario.

3 Procedure

The following procedure describes a technique which is used to determine the position of the UWB anchors randomly positioned in a local coordinate frame, based on distance measurements. It is referred to as Assumption Based Coordinates (ABC) as described by Savarese et al. [10]. A setup of four initial anchors is described while any other node can then be localized by lateration. It is assumed that these four initial anchor nodes have radio contact to each other which can be guaranteed in the presented area of application. In addition the anchors need to be positioned in a way that they can get a GNSS fix.

1. Determine all distances between anchors d_{12},d_{13},... with d_{jk} defined as the mean over N measurements ρ_{jk} between anchors j and k

$$d_{jk} = \frac{1}{N} \sum_{m=1}^{N} \rho_{jk,m} \tag{1}$$

2. Set anchor 1 to origin $\mathbf{p_1} = (0, 0, 0)$.
3. Set anchor 2 on the x-axis at $\mathbf{p_2} = (d_{12}, 0, 0)$.
4. Set anchor 3 into the z-plane $\mathbf{p_3} = (x_3, y_3, 0)$ and determine its x- and y-coordinates. Care should be taken that two solutions are possible which differ in the sign of the y-component. From

$$\begin{aligned} (x_3 - x_1)^2 + (y_3 - y_1)^2 &= d_{13}^2, \\ (x_3 - x_2)^2 + (y_3 - y_2)^2 &= d_{23}^2 \end{aligned} \tag{2}$$

follows

$$\begin{aligned} x_3 &= \frac{d_{12}^2 + d_{13}^2 - d_{23}^2}{2d_{12}}, \\ y_3 &= \pm\sqrt{d_{13}^2 - x_3^2}. \end{aligned} \tag{3}$$

5. Determine position of anchor 4. Again one of two solutions has to be chosen from. They differ in the z-component's sign. Using the relationship

$$\begin{aligned}(x_4 - x_1)^2 + (y_4 - y_1)^2 + (z_4 - z_1)^2 &= d_{14}^2,\\ (x_4 - x_2)^2 + (y_4 - y_2)^2 + (z_4 - z_2)^2 &= d_{24}^2,\\ (x_4 - x_3)^2 + (y_4 - y_3)^2 + (z_4 - z_3)^2 &= d_{34}^2\end{aligned} \tag{4}$$

the remaining coordinates can be found to be

$$\begin{aligned}x_4 &= \frac{d_{12}^2 + d_{14}^2 - d_{24}^2}{2d_{12}},\\ y_4 &= \frac{d_{14}^2 - d_{34}^2 + x_3^2 + y_3^2 - 2x_3x_4}{2y_3},\\ z_4 &= \pm\sqrt{d_{14}^2 - x_4^2 - y_4^2}.\end{aligned} \tag{5}$$

If the resulting z-component z_4 is small compared to the accuracy of the GNSS this ambiguity can not be resolved and $z_4 = 0$ is assumed.

6. Transform the local system to one suited for the application, in this case the earth-centered, earth-fixed (ECEF) cartesian system used by GNSSs. Positions can subsequently be determined in other systems e.g. the geographic coordinate system or a local cartesian system often used in RTLS.
This leads to the absolute orientation problem which consists in finding the Euclidean transformation $\mathbf{R}, \mathbf{t}$ that aligns the two sets. It is found by minimizing the mean squared residual

$$f(\mathbf{R}, \mathbf{t}) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{q}_i - (\mathbf{R}\mathbf{p}_i + \mathbf{t})\|_2 \tag{6}$$

with the GNSS coordinates $\{\mathbf{q}_i\}$ and the local ABC system coordinates $\{\mathbf{p}_i\}$ as described by Huang et al. [2] and others. As the ABC systems are pairwise equivalent except for an arbitrary rotation two of the four possible solutions have to be evaluated. The one with the lower remaining error $f(\mathbf{R}, \mathbf{t})$ is selected as the new RTLS coordinate system.

7. *Optionally* it can be desired to have a local cartesian system with a z-plane parallel to the ground. This can be achieved by first transforming the ECEF coordinates to a geodetic system with latitude Φ, longitude λ and height h ([1]). For this work the World Geodetic System 1984 (WGS84) is used. In a second step the geodetic coordinates are transformed to local tangent plane coordinates as described by Samuel [4] which yield an East-North-Up (ENU) system.

4 Implementation

The nodes are based on printed circuit boards which include a DecaWave DWM1000 UWB transceiver module, a STM32F103 ARM Cortex-M3 processor and a low

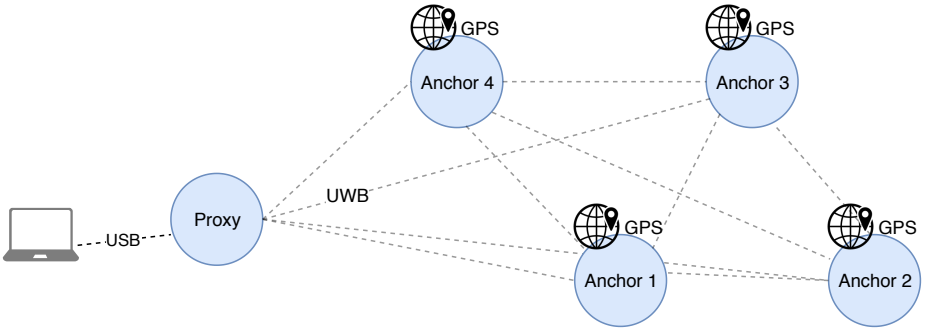
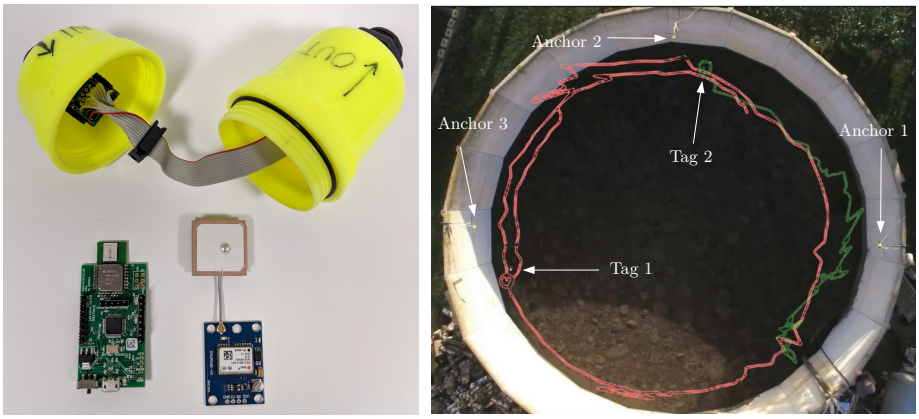


Fig. 1: Basic structure of the system.

quiescent current, low drop out voltage regulator. Anchors are additionally connected to uBlox NEO-6M modules via UART for the measurement of GPS coordinates. Nodes therefore implement a NMEA parser. As shown in Figure 1 one of these nodes serves as a proxy to the system via a virtual COM port over USB. A command interpreter in conjunction with a command interface to the UWB RTLS offers the acquisition of ranges from one anchor to a set of others. Additionally GPS coordinates can be requested. AltDS-TWR is chosen as a ranging scheme as described by Neirynck et al. [8] for its superior performance [6]. Figure 2a shows a disassembled anchor node with UWB transceiver board and GPS module developed for the application of substrate flow tracking in biogas fermenters. Figure 2b shows reconstructed traces of two tags overlayed to a video capture of a test run at an open slurry tank.



(a) Disassembled anchor node with UWB transceiver board and GPS module.

(b) Top view of tank with reconstructed traces of two tags overlayed after 2 minutes of operation.

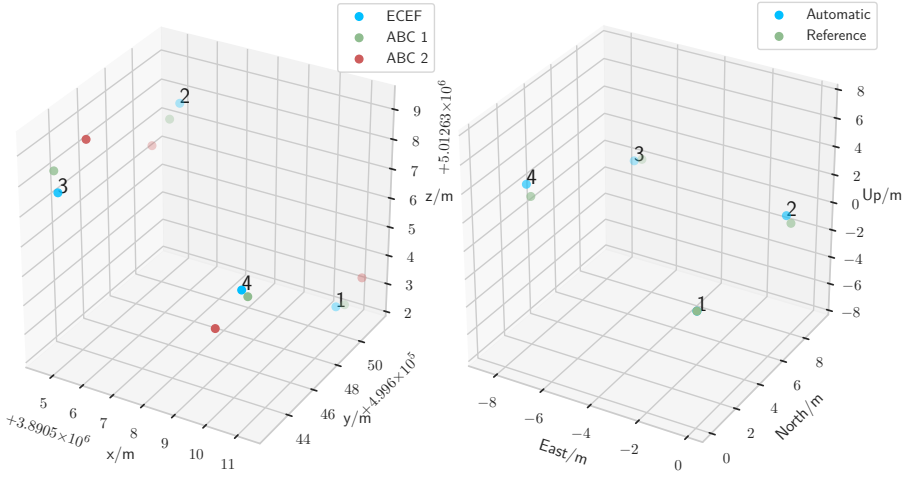
Fig. 2: Anchor setup and reconstructed position records in slurry tank.

5 Evaluation

The systems performance was evaluated by setup of the anchor nodes at the local positions given in Table 1 determined with a laser range finder. Figure 3a shows the ECEF coordinates acquired by the GNSS receivers in comparison to the rotated and translated ABC system based on UWB distance measurements. The mean error is 0.75 m. Figure 3b shows the ABC system subsequently transformed to the local ENU system next to the real reference positions. The mean error is 0.4 m.

Table 1: Local reference coordinates of anchors in test setup.

Anchor no.	x/m	y/m	z/m
1	0	0	0
2	0	7.76	0
3	-7.18	9.45	0
4	-8.24	2.63	1.84



(a) GPS and transformed ABC system in ECEF coordinates. Errors: $d_1 = 0.58\,\text{m}$, $d_2 = 1.09\,\text{m}$, $d_3 = 0.95\,\text{m}$, $d_4 = 0.37\,\text{m}$

(b) Real local positions and automatically acquired coordinate system. Errors: $d_1 = 0\,\text{m}$, $d_2 = 0.56\,\text{m}$, $d_3 = 0.37\,\text{m}$, $d_4 = 0.66\,\text{m}$

Fig. 3: Comparison of automatically determined anchor positions to GPS and local measurements.

6 Conclusion

This system offers a quick deployment of RTLS where the possibility of GNSS reception is given, such as the presented area of application. It saves time and can be more accurate than conventional techniques of anchor positioning at a comparatively low increase in cost.

Acknowledgement: This project was supported by the FNR (Fachagentur Nachwachsende Rohstoffe e.V.), funding no. 22400818.

References

1. u-blox ag: Application Note - Datum Transformations of GPS Positions. u-blox ag, Gloriastrasse 35 CH-8092 Zürich Switzerland (Jul 1999), https://microem.ru/files/2012/08/GPS.G1-X-00006.pdf
2. Arun, K.S., Huang, T.S., Blostein, S.D.: Least-squares fitting of two 3-d point sets. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-9**(5), 698–700 (Sep 1987). https://doi.org/10.1109/TPAMI.1987.4767965
3. Bosch: Zamo Digital laser measure datasheet (Jun 2019), https://www.bosch-do-it.de/media/garden/gardenmedia/manuals/1509947_160992A454_201806pdf..pdf
4. Drake, S.: Converting GPS coordinates [phi, lambda, h] to navigation coordinates (ENU). DSTO **DSTO-TN** (04 2002)
5. Gentner, C., Ulmschneider, M., Jost, T.: Cooperative simultaneous localization and mapping for pedestrians using low-cost ultra-wideband system and gyroscope. In: 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS). pp. 1197–1205 (April 2018). https://doi.org/10.1109/PLANS.2018.8373505
6. Lian Sang, C., Adams, M., Hörmann, T., Hesse, M., Porrmann, M., Rückert, U.: Numerical and Experimental Evaluation of Error Estimation for Two-Way Ranging Methods. Sensors **19**(3), 616 (Feb 2019). https://doi.org/10.3390/s19030616
7. Moravek, P., Komosny, D., Simek, M., Muller, J.: Multilateration and flip ambiguity mitigation in ad-hoc networks. Przeglad Elektrotechniczny **88** (01 2012)
8. Neirynck, D., Luk, E., Mclaughlin, M.: An alternative double-sided two-way ranging method (10 2016). https://doi.org/10.1109/WPNC.2016.7822844
9. Pelka, M., Goronzy, G., Hellbrück, H.: Iterative approach for anchor configuration of positioning systems. ICT Express **2**(1), 1 – 4 (2016). https://doi.org/10.1016/j.icte.2016.02.009
10. Savarese, C., Rabaey, J., Beutel, J.: Location in distributed ad-hoc wireless sensor networks. vol. 4, pp. 2037 – 2040 vol.4 (05 2001). https://doi.org/10.1109/ICASSP.2001.940391
11. Shi, Q., Zhao, S., Oui, X., Lu, M., Jia, M.: Anchor self-localization algorithm based on uwb ranging and inertial measurements. Tsinghua Science and Technology **24**(6), 728–737 (Dec 2019). https://doi.org/10.26599/TST.2018.9010102
12. Trimble: S9/S9 HP TOTAL STATION datasheet (Jun 2019), https://geospatial.trimble.com/sites/default/files/2019-06/022516-155G_TrimbleS9_DS_USL_0619_LRsec.pdf
13. uBlox: NEO-6 datasheet (Jun 2019), https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf

Comparison of machine-learning methods for multi-step-ahead prediction of wave and wind conditions

Mengning Wu[1], Zhen Gao[1, 2, 3], Christos Stefanakos[4], Sverre Haver[1,5]

[1] Department of Marine Technology, Norwegian University of Science and Technology (NTNU), Trondheim NO-7491, Norway
[2] Centre for Autonomous Marine Operations and Systems (AMOS), NTNU, Trondheim NO-7491, Norway
[3] Centre for Marine Operations in Virtual Environments (MOVE), NTNU, Trondheim NO-7491, Norway
[4] SINTEF Ocean, Department of Environment and New Resources, Trondheim NO-7465, Norway
[5] Department of Mechanical and Structural Engineering and Materials Science, University of Stavanger, Stavanger NO-4036, Norway

Abstract. Short-term prediction of wave and wind conditions is important for decision-making during the execution of marine operations. Typically, short-term weather conditions can be forecasted using physics-based models or data-driven models. This paper addresses machine-learning methods for weather forecast. So far, one-step-ahead forecasting can be handled well based on the various methods, while multi-step-ahead forecasting of weather conditions is still challenging and rarely studied. This paper aims to present a comparison of the application of different machine-learning methods in multi-step-ahead wave and wind predictions. Before prediction, two pre-processing techniques, namely decomposition technique and empirical mode decomposition (EMD) are considered on the time series to reduce the prediction complexity. Several widely known data-driven models including the autoregressive integrated moving average (ARIMA), artificial neural network (ANN), recurrent neural network (RNN) and the adaptive-network-based fuzzy inference system (ANFIS) are employed. To perform multi-step forecasting, three multi-step-ahead models (M-1, M-N and M-mN model) are considered. The data set used to assess the accuracy of the methods comprises hourly time series of the mean wind speed Uw, the significant wave height Hs and the spectral peak period Tp from 2001 to 2010 at the North Sea center. After prediction, an uncertainty quantification analysis is conducted to compare the forecasting performance of the methods. The results indicate that the considered machine-learning methods can provide acceptable predictions for weather conditions for the first several steps ahead. However, all methods encountered an essential problem that the forecast uncertainty would increase significantly with the forecast horizon. In this case, more research is needed to develop a method that incorporates physical processes into the data-driven model to improve the weather forecast accuracy.

Keywords: Machine-learning method, Weather forecast, Uncertainty quantification.

Introduction

Accurate weather forecasting can provide an important basis for planning marine operations and strengthening control during the execution phase. However, prediction of weather conditions is quite challenging due to the random and unsteady characteristics of wind and waves. In particular, multi-step-ahead prediction of weather conditions is typically faced with growing uncertainties, making it difficult to obtain accurate predictions over long forecast horizon.

In general, there are two main categories utilized for wave and wind forecasting, i.e. physics-based models and data-driven models. Physics-based models rely on physical process of the atmosphere and wave evolution and makes prediction by solving complex mathematical models that use weather data like temperatures, pressure, surface roughness and so on. Many physical models have been introduced, and the most popular ones are WAM (Group, 1988), Wave Watch III (Tolman, 1991) and SWAN (Booij et al., 1999) for wave forecasting and numerical weather prediction (NWP) (Cassola and Burlando, 2012; Landberg, 1999) for wind forecasting.

Instead of solving complex physics-based models, data-driven models have been developed for time series forecasting, which only based on the historical data. The data-driven model applies time-series statistical analyses or machine learning algorithms to generate predictions. In the conventional statistical models, the future values of wave and wind can be expressed by a linear function of historical data, which can provide a good balance between implementation simplicity and forecasting accuracy. The popular statistical models include the autoregressive (AR) model (Brown et al., 1984; Poggi et al., 2003), the autoregressive moving average (ARMA) model (Erdem and Shi, 2011; Lydia et al., 2016; Torres et al., 2005), the autoregressive integrated moving average (ARIMA) model (Kavasseri and Seetharaman, 2009), etc. However, they generally show significant limitations in the forecasting horizon and the ability to model nonlinear data patterns (Qin et al., 2017). In recent years, machine learning models have been greatly developed and utilized for wave and wind predictions. They can characterize a complicated relationship between input and output data through a network and provide forecasts by applying virous algorithms to the network. The popular and widely known machine learning models are support vector machines (SVMs) (Berbić et al., 2017; Kamranzad et al., 2011; Mohandes et al., 2004), artificial neural network (ANN) (Cadenas and Rivera, 2009; Chang et al., 2017; Flores et al., 2005; Jain and Deo, 2007; Li and Shi, 2010), recurrent neural network (RNN) (Mandal and Prabaharan, 2006; Olaofe, 2014) and adaptive-network-based fuzzy inference system (ANFIS) (Akpınar et al., 2014; Özger and Şen, 2007; Stefanakos, 2016a, b). In this paper, machine-learning methods comprising different data pre-processing techniques and data-driven models are proposed for multi-step-ahead weather forecasting. Furthermore, a comprehensively comparison on the prediction performance of the proposed methods is made by using the statistics of forecast errors for evaluating forecast uncertainties.

The remainder of this paper is organized as follows: Section 2 describes two data pre-processing techniques, which are the decomposition technique and EMD. In Section 3, the details of data-driven models for prediction are introduced, including ARIMA,

ANN, RNN and ANFIS. Section 4 presents three multi-step-ahead prediction models, namely M-1, M-N and M-mN model. Section 5 gives a brief description of the study site and the considered wave and wind data. Section 6 summarizes the applied hybrid machine-learning methods and corresponding results. Finally, the main conclusions are given in Section 7.

2 Data pre-processing

Data pre-processing is an important step in weather forecasting due to the random and unsteady characteristics of wind and wave conditions. In this subsection, two pre-processing techniques, which are decomposition technique and empirical mode decomposition (EMD), are described.

2.1 Decomposition technique

The decomposition technique was developed by Athanassoulis and Stefanakos (1995), which can be utilized to eliminate the non-stationary influence in weather prediction process. It converts the initial time series to stationary ones by extracting the monthly mean value vector and the covariance matrix. The decomposition model for multi-variate time series (Stefanakos and Schinas, 2014) is defined by

$$\underset{(\mathrm{N}\times 1)}{\boldsymbol{Y}(\boldsymbol{t})} = \underset{(N\times 1)}{\boldsymbol{M}(\boldsymbol{t})} + \underset{(N\times N)}{\boldsymbol{\Sigma}(\mathbf{t})} \underset{(N\times 1)}{\mathbf{W}(\mathbf{t})} \tag{1}$$

or, in matrix notation,

$$\begin{bmatrix} Y_1(t) \\ Y_2(t) \\ \vdots \\ Y_n(t) \\ \vdots \\ Y_N(t) \end{bmatrix} = \begin{bmatrix} M_1(t) \\ M_2(t) \\ \vdots \\ M_n(t) \\ \vdots \\ M_N(t) \end{bmatrix} + \begin{bmatrix} \Sigma_{11}(t) & \Sigma_{12}(t) & \cdots & \Sigma_{1N}(t) \\ \Sigma_{21}(t) & \Sigma_{22}(t) & \cdots & \Sigma_{2N}(t) \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma_{n1}(t) & \Sigma_{n2}(t) & \cdots & \Sigma_{nN}(t) \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma_{N1}(t) & \Sigma_{N2}(t) & \cdots & \Sigma_{NN}(t) \end{bmatrix} \begin{bmatrix} W_1(t) \\ W_2(t) \\ \vdots \\ W_n(t) \\ \vdots \\ W_N(t) \end{bmatrix} \tag{2}$$

where N is the number of time series. $\mathbf{Y}(t)$ and $\mathbf{W}(t)$ represent the initial and corresponding stationary time series, respectively. $\mathbf{M}(t)$ and $\boldsymbol{\Sigma}(t)$ are the monthly mean value vector and the covariance matrix with period of one year respectively, which are called the 'seasonal patterns' of the initial time series.

The seasonal patterns can be estimated by averaging the monthly mean values $M_{3,n}(j,m)$ and the covariance matrix $S_{3,nl}(j,m)$ over J years (Stefanakos et al., 2006), which are shown as Eq. (3) and (4). In the two equations, $Y_n(j,m,\tau_k)$ is a re-parameterized expression of $Y_n(t)$, where j is the year index, m is the month index and τ_k is the k^{th} (k=1,2,…,K_m) observation in the m^{th} (m=1,2,…,12) month.

By extracting the seasonal patterns from the initial time series, the stationary ones $\mathbf{W}(t)$ can be obtained, which is a zero-mean, stationary stochastic process. For a detailed introduction, refer to (Stefanakos et al., 2002; Stefanakos and Schinas, 2014).

$$\widetilde{M}_{3,n}(m) = \frac{1}{J}\sum_{j=1}^{J} M_{3,n}(j,m) = \frac{1}{J}\sum_{j=1}^{J}\frac{1}{K_m}\sum_{k=1}^{K_m} Y_n(j,m,\tau_k) \tag{3}$$

$$\tilde{S}_{3,nl}(m) = \frac{1}{J}\sum_{j=1}^{J} S_{3,nl}(j,m) =$$

$$\frac{1}{J}\sum_{j=1}^{J}\sqrt{\frac{1}{K_m}\sum_{k=1}^{K_m}[Y_n(j,m,\tau_k) - M_{3,n}(j,m)][Y_l(j,m,\tau_k) - M_{3,l}(j,m)]},\ n,l = 1,\cdots,N \quad (4)$$

2.2 EMD (Empirical Mode Decomposition)

EWD (Huang et al., 1998) is a self-adaptive time series decomposition technique which can analyze non-linear and non-stationary signals. The basic idea of EMD is to decompose the signal into a set of oscillatory components called intrinsic mode functions (IMFs) and a residue. An IMF is a complete and nearly orthogonal basis for the signal and it needs to fulfill two basic conditions: 1) the number of zero-crossings and extrema must be equal or differ at the most by one in the entire data set; 2) the mean value of the envelope defined by local minima and the one defined by local maxima is zero at any point in the IMF. According to the above definitions, the initial time series *Y(t)* can be decomposed as

$$Y(t) = \sum_{i=1}^{n-1} I_i(t) + r_n(t) \quad (5)$$

where $I_i(t)$ is the i[th] IMF and $r_n(t)$ is the residue of the initial signal. For a detailed description and solution process, see (Huang et al., 1998; Huang et al., 2003; Huang and Wu, 2008).

3 Machine-learning methods for prediction

In this subsection, three representative machine-learning models for prediction are briefly introduced, which are the artificial neural network (ANN), recurrent neural network (RNN) and the adaptive-network-based fuzzy inference system (ANFIS). Prior to these, a traditional statistical model, i.e. the autoregressive integrated moving average (ARIMA) is first introduced.

3.1 ARIMA

The autoregressive integrated moving average (ARIMA) model is one of the most classical forecasting techniques that based on the time series statistical analysis. It is the combination of autoregressive (AR), integrated (I) and moving average (MA) processes. In order to reflect the structure of an ARIMA model, it is generally denoted as ARIMA(p,d,q), where: 1) p is the order of the AR model, representing the number of lags of the differenced series; 2) d is the order of the differencing, which is used to stable the initial data; 3) q is the order of MA model, representing the number of lags of the prediction errors. A typical ARIMA model which expressing the data at time t as a linear function of previous data and white noise errors can be defined in the form

$$\hat{Y}(t) = \sum_{i=1}^{p} \emptyset_i \hat{Y}(t-i) + \sum_{j=1}^{q} \theta_j e(t-j) + \mu(t) \quad (6)$$

where $\hat{Y}(t)$ is the predicted data at time t. $\hat{Y}(t-i)$ and $e(t-j)$ are the past data

and random errors at time t-i and t-j, respectively. It should be noted that the data $\hat{Y}(t)$ in Eq. (6) are stationary after making a d-order difference to the initial data. Other parameters like $\emptyset_i$ and θ_j are the unknown i[th] autoregressive and j[th] moving average coefficients respectively, and $\mu(t)$ is a constant term. For a detailed description, see (Box et al., 2015).

In this study, the length of the time series for analysis in the ARIMA model is first considered. Then, the construction of the model of each fixed-length time series is studied, mainly includes identifying the p, q and d indices, and determining the $\emptyset$ and θ parameters contained in the model. Finally, the developed ARIMA model is utilized to predict future time series.

3.2 ANN

The artificial neural network (ANN) is a typical machine learning model that mimics the process of biological neurons system for receiving, processing and transmitting information (Rosenblatt, 1958). Figure 1 demonstrates a basic feedforward neural network, which consists of one input layer, one output layer and one hidden layer. Each layer is made up of a set of interconnected neurons, and each of which contains an activation function *f*. Besides, the neurons in different layers are connected to each other by weights *w*. To display it clearly, Eq. (7) shows the output h_i of the hidden layer which is calculated based on the weighted inputs of the input layer

$$h_i = f(\sum_{j=1}^{n} w_{ij} x_{ij} + b_i) \tag{7}$$

where *f* is the activation function of i[th] neuron in hidden layer. w_{ij} is the weight value from input x_j to hidden layer's neuron h_i, and b_i is the bias of neuron h_i.

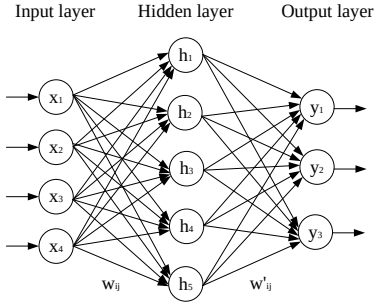


Fig. 1. The single-layer feedforward neural network

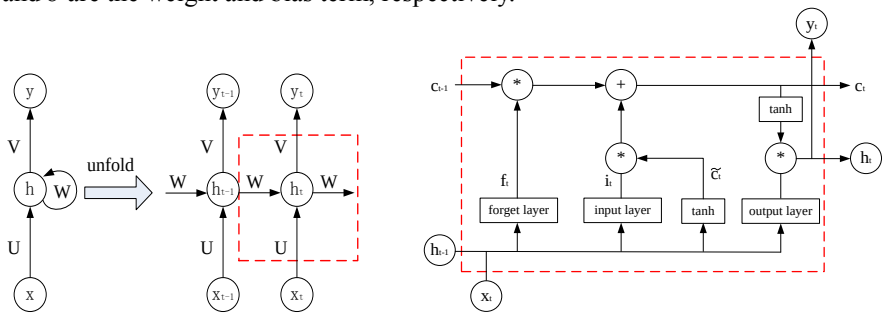
Based on the ANN, several advanced neural networks have been proposed to more effectively approximate human brain activities. One representative is back-propagation neural network (BPNN). BPNN was proposed by Rumelhart et al. (1988), as a way to improve the accuracy of ANN calculation. Compared to the ANN, BPNN not only has a forward propagation process of information, but also has an error back-propagation process to adjust the weights and bias after each training epoch based on a gradient descent method. By this way, the optimal weights and bias can be better identified. For a detailed description, see (LeCun et al., 1988; Rumelhart et al., 1988). In the study, all ANN-based models apply the BP algorithm.

3.3 RNN

RNN is another advanced ANN which is especially designed for analyzing sequential data. Compared to a classical ANN, RNN can preserve the intrinsic temporal dependencies in the time series based on the existence of loop structure where the current output depends on the previous hidden neurons. As illustrated in Figure 2(a), x_t is the input, y_t is the output and h_t is the hidden output at time step t. U, V and W are the input weights, output weights and recurrent layer weights, respectively. In RNN, the hidden state h_t is computed not only based on the current input x_t, but also on the previous hidden state h_{t-1}, from which can make full use of the information of the previous sequence.

$$h_t = tanh(Ux_t + Wh_{t-1}) \tag{8}$$

However, researches have proved that RNN may suffer from the vanishing gradient problem during the training process, especially in long sequence (Bengio et al., 1994). As a result, LSTM as an extended version was proposed by Hochreiter and Schmidhuber (1997), which can tackle this problem through its special structure of hidden layers. In contrast to a standard RNN, three gate structures, namely forget gate, input gate and output gate, are incorporated in the hidden layer in LSTM, that enable to control information flow between different time steps and avoid the long-term dependencies problem. Figure 2(b) shows the special hidden layer of LSTM, where i_t, f_t, and o_t represent the input, forget and output gates at time t, respectively; and c_t and $\tilde{c}_t$ represent the memory cell and the new memory cell at time t, respectively. To be specific, the forget gate f_t decides the proportion of information to be forgot in the new memory cell c_t through Eq. (9). The input gate i_t decides the proportion of new information to be added in the new memory cell c_t through Eq. (10). The output gate o_t controls the proportion of new information to be exported through Eq. (11). In addition, memory cell c_t and $\tilde{c}_t$ are calculated by Eq. (12) and (13), respectively. Finally, the output y_t is calculated as Eq. (14). In Eqs. (9)-(12), σ represents the sigmoid function. w and b are the weight and bias term, respectively.



(a) RNN (b) LSTM

Fig. 2. The structures of RNN and LSTM

$$i_t = \sigma(w_i x_t + u_i h_{t-1} + b_i) \tag{10}$$

$$o_t = \sigma(w_o x_t + u_o h_{t-1} + b_o) \quad (11)$$

$$\widetilde{c_t} = tanh(w_c x_t + u_c h_{t-1} + b_c) \quad (12)$$

$$c_t = f_t c_{t-1} + i_t \widetilde{c_t} \quad (13)$$

$$y_t = o_t tanh(c_t) \quad (14)$$

3.4 ANFIS

In recent years, the adaptive-network-based fuzzy inference system (ANFIS) has become an attractive model for time series forecasting. It is a hybrid model which combines a fuzzy inference system (FIS) and ANN. FIS is a process mapping inputs to an output based on the membership functions (MFs) and IF-THEN rules. MFs characterize fuzziness and IF-THEN rules represent the conditional statements containing fuzzy logic. A basic structure of a FIS is illustrated in Figure 3, which is generally composed of four components: 1) fuzzy knowledge base, that contains the MFs and IF-THEN rules; 2) fuzzifier, that fuzzifies the crisp inputs into fuzzy inputs by using the MFs; 3) inference engines, that maps fuzzy inputs to fuzzy outputs according to the IF-THEN rules; 4) defuzzifier, that defuzzifies the fuzzy outputs into a crisp output.

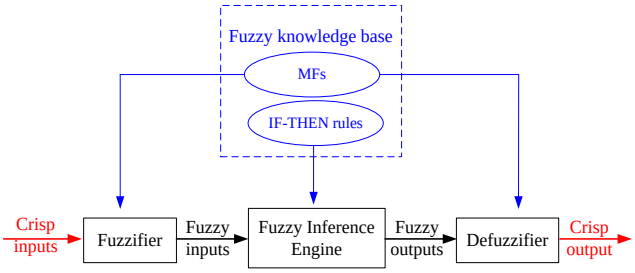


Fig. 3. The structure of FIS

In the FIS, both MFs and IF-THEN rules can only be determined by experts or past available data, so they cannot be adjusted automatically. As a result, a new strategy called ANFIS was proposed by Jang (1993) that uses ANN together with FIS to solve this problem. In the ANFIS, ANN is introduced to optimize the parameters of IF-THEN rules and MFs using a hybrid learning algorithm of the gradient descent and the least-squares estimate. Based on this, ANFIS can take advantage of the advantages of both methods and has been successfully applied to many research fields, especially in field dealing with time series prediction. For a detailed introduction about FIS and ANFIS, refer to (Jang, 1993; Takagi and Sugeno, 1983, 1985).

4 Multi-step-ahead prediction models

To realize short-term prediction of weather conditions, three multi-step-ahead prediction models are utilized, i.e. M-1 model, M-N model and M-mN model. To

maintain the consistently of the variables, *X(t)* denotes data at time t and *f* denotes the prediction model. Besides, *N* and *M* denote the number of forecast step (i.e. forecast horizon) and input data, respectively.

4.1 M-1 model

The M-1 model means applying the previous M data to predict the next data based on a one-step-ahead prediction model, which can be illustrated in Eq. (15). To obtain N steps ahead predictions, this prediction process needs to be performed N times recursively. The mechanism of this model is shown in Figure 4. As illustrated in Figure 4, the actual and forecasted data are represented by white and yellow boxes, respectively. Obviously, the forecasted value at time step t+1 is considered as part of the input set for predicting the next value at time step t+2, on the basis of the same one-step-ahead model. This model is relatively easy since only one training process is required and the developed model does not change in the whole prediction process. However, as the number of time steps increases, more and more predicted data are used as inputs for prediction. This may lead to the problem of error accumulation.

$$X(t+1) = f_1(X(t), X(t-1), \ldots, X(t-M+1)) \tag{15}$$

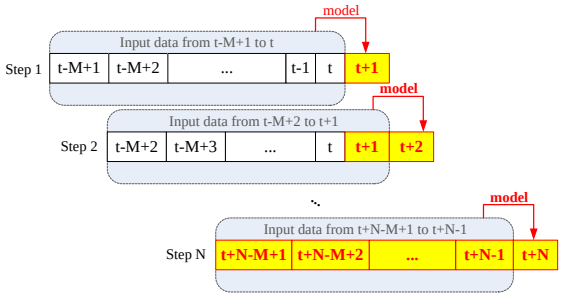


Fig. 4. The architecture of the M-1 model

4.2 M-N model

In the M-N model, it is necessary to develop specific models for each time step within the forecast horizon. The prediction model can be represented by Eq. (16), where f_N is the prediction model at step N. This implies that to obtain N steps ahead predictions, N corresponding models need to be established. Figure 5 shows the mechanism of this M-N model. In Figure 5, the forecasted values are represented by boxed with different colors. It illustrates that the input set is always composed of the historical M data, and all forecast values are predicted based on these known data. As a result, this model can prevent accumulated errors. However, there are also some problems with the application of the M-N model. First, compared to the M-1 model, it is time-consuming because N prediction models need to be developed. Second, it is difficult to capture the relationship between inputs and output when N is large, since their correlation is inherently weak. In addition, predictions are generated independently at different time steps, which may result in a discontinuities predicted time series.

$$X(t+N) = f_N(X(t), X(t-1), \ldots, X(t-M+1)) \quad (16)$$

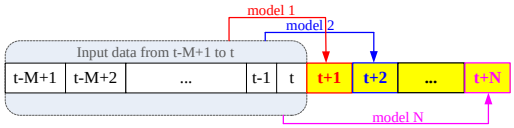


Fig. 5. The architecture of the M-N model

4.3 M-mN model

The previous two models can be considered as single-output models. By contrast, the M-mN model is a multiple-output model. The prediction model is represented by Eq. (17) and its architecture is sketched in Figure 6. From Figure 6, one can observe that in the M-mN model, the previous M data can be used to predict N values simultaneously. This implies that compared with the M-1 model, this model can avoid the accumulation of errors. In addition, only one model with M inputs and N outputs needs to be developed, which significantly reduces the computational time and maintains the dependencies among the forecasted values compared to the M-N model. However, when N is relatively large, predicting all future values by only one model structure may reduce the flexibility of the forecasting and may result in low prediction accuracy.

$$[X(t+1), X(t+2), \ldots, X(t+N)] = f(X(t), X(t-1), \ldots, X(t-M+1)) \quad (17)$$

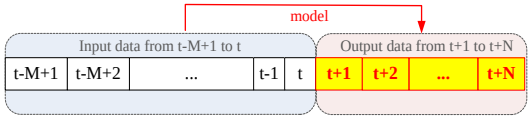


Fig. 6. The architecture of the M-mN model

5 Study area and data set

In this study, the hindcast wind and wave data at the North Sea center (site 15, Figure 7) are adopted. Specifically, ten-year hourly time series of the mean wind speed U_w, the significant wave height H_s and the peak spectral wave period T_p are gathered from January 2001 to December 2010. According to the execution time of a typical marine operation, the aim of this study is to perform 24-steps-ahead predictions for wave and wind conditions. The mentioned data is divided into two groups, which are the first nine-year data (2001-2009) and the tenth-year data (2010). The first one is utilized to generate stationary time series of years 2001-2009 and extract seasonal patterns via the decomposition technique. Then the second group of data can be decomposed into the corresponding stationary time series based on the seasonal patterns estimated from the first group. For a more detailed description, see (Wu et al., 2019). After that, the obtained stationary time series for the first nine-year and the tenth-year are utilized to train and test the data-driven models, respectively. It should be noted that the last 1000

of the training data is utilized as the validation data, to provides an unbiased evaluation of a model fit on the training dataset while tuning the model's hyperparameters (Bishop, 1995).

Fig. 7. Map of the North Sea area and the location of the North Sea Center

6 Results and discussion

Table 1 summaries the proposed hybrid prediction methods in this study based on different pre-processing techniques, data-driven models and multi-step-ahead prediction models. Since the efficacy of the decomposition technique has been successfully verified in the time series prediction of weather conditions (Stefanakos, 2016a; Wu et al., 2018), all methods except 7 apply it as the data pre-processing technique. In order to investigate the influence of pre-processing technique, EMD is applied to continue decompose the obtained data to a series of quasi-stationary IMFs in methods 5, 6, 10 and 11. In addition, both ANN and RNN-based methods use M-mN model for time series prediction (methods 4 and 7), while ANFIS-based method does not because ANFIS can only generate one output value. The ARIMA model, ANN-based model and ANFIS-based model are performed by using Matlab, and RNN-based model is performed by using TensorFlow.

Table 1. The list of proposed prediction methods.

No.	Method	Pre-processing technique		Data-driven model				Multi-step-ahead model			No. of ε_M
		Decomposition	EMD	ARIMA	ANN	RNN	ANFIS	M-1	M-N	M-mN	
1	D-ARIMA	√		√							365
2	D-ANN-M-1	√			√			√			365
3	D-ANN-M-N	√			√				√		365
4	D-ANN-M-mN	√			√					√	365

5	D-EMD-ANN-M-1	√	√	√			√			365
6	D-EMD-ANN-M-N	√	√	√				√		365
7	RNN-mN				√				√	365
8	D-ANFIS-M-1	√					√			365
9	D-ANFIS-M-N	√						√		365
10	D-EMD-ANFIS-M-1	√	√			√	√			365
11	D-EMD-ANFIS-M-N	√	√			√		√		365

To assess the multi-step prediction performance of the proposed methods, a forecast error factor $\varepsilon_M(t)$ is introduced to calculate the forecast error of each prediction value as shown in Eq. (18). The forecast uncertainty in the prediction model is quantified by the mean value and standard deviation of the series of ε_M from the testing phase. The smaller they are, the less the forecast uncertainty is.

$$\varepsilon_M(t) = \frac{f(t)-a(t)}{a(t)} \tag{18}$$

where $f(t)$ and $a(t)$ are the forecasted and actual value at the same time t, respectively. In the following subsections, the forecast uncertainty results of each prediction method and the comparison between the different methods are displayed. It needs to be emphasized that due to page limitations, only prediction results of H_s are summarized and compared in this paper.

6.1 Forecasting use ARIMA

In this subsection, the traditional ARIMA model is employed for forecasting significant wave height. To construct the model, a differencing approach is applied to stable the initial time series, in which the autocorrelation function (ACF) and partial autocorrelation function (PACF) are both utilized to determine the order of differencing. According to the patterns in ACF and PACF plots, ARIMA(p,q,2) is considered for this time series. Additionally, the Bayesian Information Criterion (BIC) is used to specify the orders for the auto-regressive and moving average in each ARIMA model, namely indices p and q. The model yielding the minimum BIC value is selected. After determining the structure of ARIMA model, the next step is to perform 24-steps-ahead H_s prediction. In this study, the data of year 2010 is considered as testing data, and daily prediction values (i.e. 24-steps ahead predictions) are obtained by using a specific ARIMA model. The model applies fixed-length of previous data as the training set, and the structure of the model depends on it. The length of training set is chosen to be 1, 2, 4, 6 and 8 months for comparison. The sketch of ARIMA prediction method used in this study is illustrated in Figure 8. After prediction and uncertainty quantification, the forecast uncertainty at each forecast step of each model is calculated and plotted in

Figure 9.

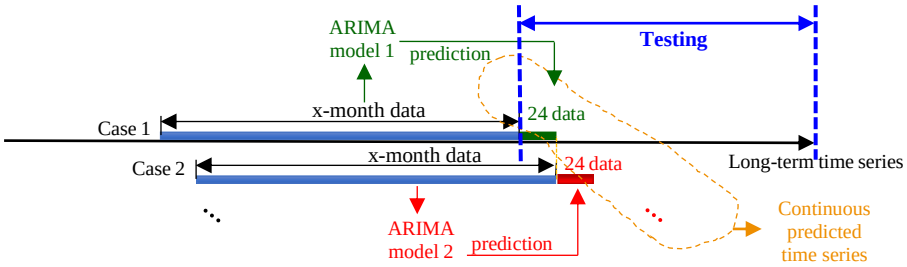


Fig. 8. Sketch of ARIMA prediction method

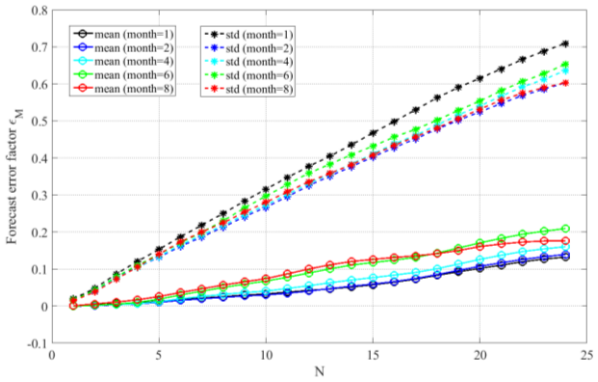


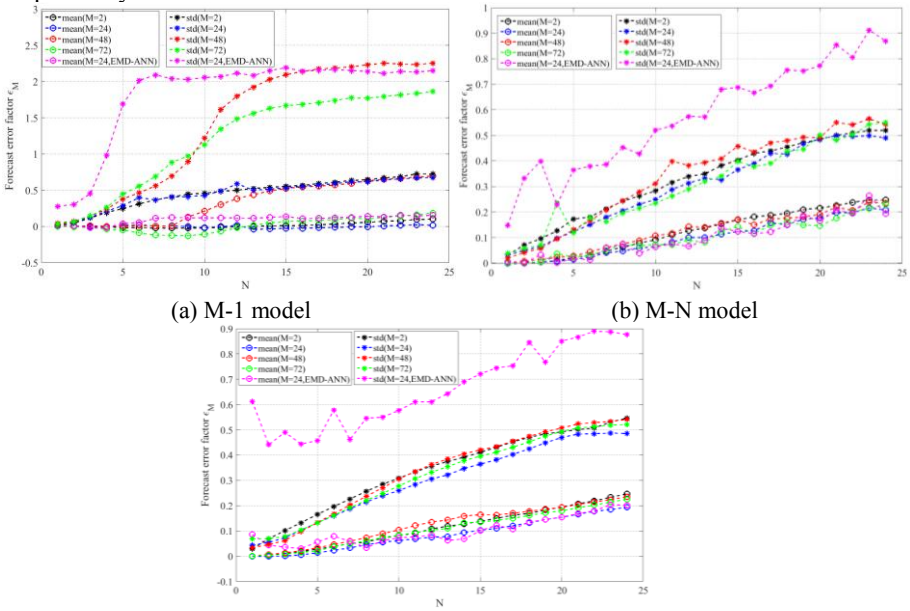
Fig. 9. Forecast uncertainty in H_s prediction based on ARIMA models

Figure 9 shows that both mean value and standard deviation of forecast error factors increase nearly linearly with the forecast horizon. By observing the different lines in the figure, it is found that there is little difference in forecast performance regardless of the length of applied data. In comparison, the relative better ARIMA model is the one using the previous 2 months of data to predict the next 24 hours data, and further increasing the length of training data does not improve the forecast accuracy. With regard to the computational time, it should be noted that this method is very time-consuming since each prediction is performed by a specific prediction model constructed based on its previous two months of data.

6.2 Forecasting use ANN

The ANN model used in this study is a three-layer feed-forward back-propagation network. The number of nodes in the input layer is equal to the size of input set M. In order to determine how many previous data should be selected as the inputs, M is chosen as 2, 24, 48 and 72. The second layer is a hidden layer which includes three neurons, and the third layer is an output layer, whose number of nodes is one in the M-1 and M-N model and N in the M-mN model. In order to study the impact of data pre-processing technique, an ANN-based method with double pre-processing techniques

(decomposition technique and EMD) is also constructed, with the input size M of 24. Figure 10 depicts the forecast uncertainty at each forecast step based on the D-ANN method with three different multi-step-ahead prediction models. By comparing D-ANN method with D-EMD-ANN method in three subfigures, it is apparent that the forecast accuracy of D-ANN is superior to that of D-EMD-ANN method. This may because after performing the double data pre-processing, the obtained a set of series are very sensitive to noise and may run into the problem of mode mixing. These problems can cause large deviation between the final prediction which reconstructed by the predicted components and the actual data. Furthermore, D-ANN models with different M values display similar accuracy in multi-step-ahead prediction. This is clearly observed in Figure 10(b) and (c). While in Figure 10(a), when M value is larger than 24, the forecast ability of the D-ANN M-1 method is significantly deteriorated. This can be attributed to the fact that the M-1 model applies a one-step-ahead prediction model iteratively, which is susceptible to the over-fitting problem given a large number of input data. In summary, by considering both the computation time and the forecast uncertainty, the optimal M value is chosen to be 24 for all three ANN-based methods. As an example, six consecutive prediction results based on the D-ANN M-1 method (M=24) are shown in Figure 11, where black and red lines represent actual and predicted time series, respectively. Green and blue marks show the beginning and end of each prediction, respectively.



(a) M-1 model (b) M-N model

(c) M-mN model

Fig. 20. Forecast uncertainty in Hs prediction based on ANN-based methods

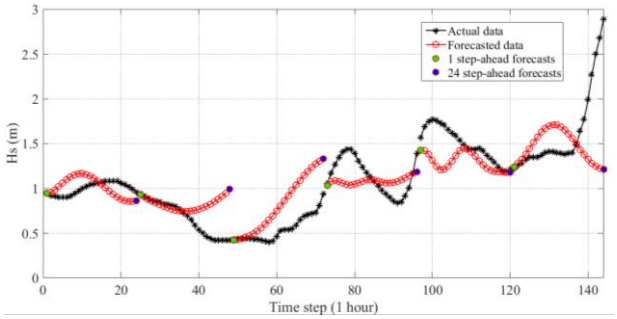
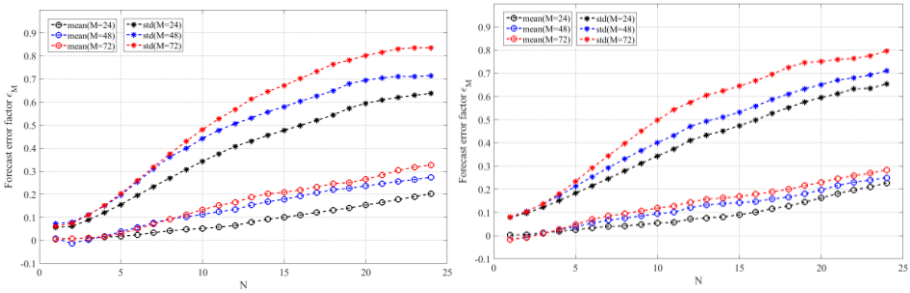


Fig. 31. Six prediction results based on ANN M-1 method

6.3 Forecasting use RNN

In this subsection, LSTM network which is a special kind of RNN model is utilized for 24 steps-ahead significant wave height prediction. It is set up to consist of two LSTM layers and the size of hidden unit in each LSTM layer is 32. In addition, the tanh function is considered as the activation function of the LSTM layer and the Adam algorithm is utilized for optimization. Since LSTM is designed to handle sequence dependence, only M-mN model is considered in this subsection. To evaluate the influence of the selection of input variables, two kinds of LSTM network are established. One only utilizes H_s as input variable, and the other utilizes both H_s and U_w as input variables. Moreover, to determine the optimal value of input set M, M is fixed to 24, 48 and 72 with the same LSTM network. After calculation, the forecast uncertainty results from different LSTM-based methods are summarized in Figure 12.



(a) H_s as input variables (b) H_s and U_w as input variables

Fig. 42. Forecast uncertainty in Hs prediction based on LSTM methods

From both Figure 12(a) and (b), one can observe that all three methods can give a more accurate prediction at the beginning, while as the forecast step increases, the RNN M-mN model using previous one day's data as input performs better than models with more input data. This means that the RNN model can capture the internal relationship in a time series using fewer past data. By further comparing the best results in subfigure (a) and (b), the optimal RNN-based method in the study area is considered to be the RNN M-mN model with H_s as the input variable and M of 24.

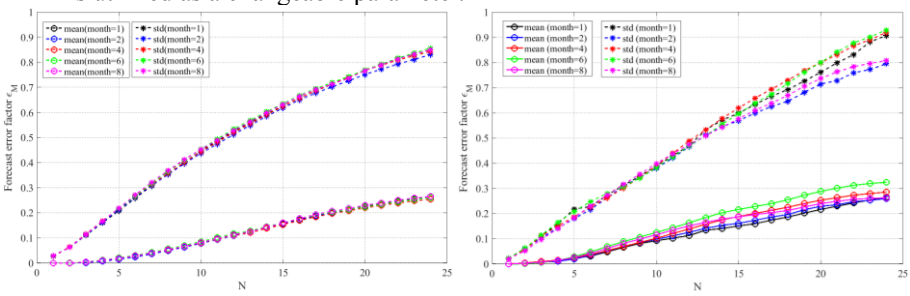
6.4 Forecasting use ANFIS

In this subsection, simple ANFIS models are developed for 24-steps-ahead significant wave height prediction. In the models, all input variables are partitioned into two fuzzy set, and the Gaussian-type and linear-type functions are selected as the membership function for inputs and output, respectively. Since ANFIS belongs to the single-output approach, only M-1 and M-N models are considered. The best ANFIS M-1 model is determined by comparing the performance of models with different M values. With regard to the ANFIS M-N model, other variables like wind direction *Dir* and wind speed U_w can also be considered as input variables as waves are influenced by the strength of the wind. To make the paper concise, the best ANFIS M-1 and M-N model are given directly in Eq. (19) and (20), respectively. Detailed calculations and comparisons can be found in (Wu et al., 2019) .

$$\mathrm{H}_s(\mathrm{t}+1) = f\big(\mathrm{H}_s(\mathrm{t}), \mathrm{H}_s(\mathrm{t}-1)\big) \tag{19}$$

$$\mathrm{H}_s(\mathrm{t}+\mathrm{N}) = f\big(\mathrm{H}_s(\mathrm{t}), \mathrm{U}_w(\mathrm{t}), Dir(\mathrm{t}), \mathrm{H}_s(\mathrm{t}-1), \mathrm{U}_w(\mathrm{t}-1)\big) \tag{20}$$

Similar to Section 6.3, ANFIS-based methods with double pre-processing techniques are also constructed to study the impact of data pre-processing technique on the ANFIS model. To select the optimal EMD-ANFIS model, the decomposition length of the EMD is utilized as a changeable parameter.



(a) M-1 model (b) M-N model

Fig. 53. Forecast uncertainty in Hs prediction based on EMD-ANFIS methods

Figure 13 shows the comparison of forecast uncertainties based on EMD-ANFIS methods. As displayed in Figure 13, there is not much difference in the forecast uncertainty among EMD-ANFIS methods, especially when using the M-1 model. This suggests that ANFIS-based method is more flexible for different kinds of data and can give more stable results. By considering both the computation time and the forecast uncertainty, the EMD-ANFIS M-1 model with one-month decomposition length and the END-ANFIS M-N model with two-month decomposition length are considered as the best models in Figure 13(a) and (b), respectively.

6.5 Comparison

This subsection provides a comparative analysis on the forecasting performance among

the optimal structure of the proposed methods described in the above subsections. The number of ε_M used to quantify the uncertainty of each method is summarized in the last column of Table 1, and the corresponding forecast uncertainty results are plotted in Figure 14.

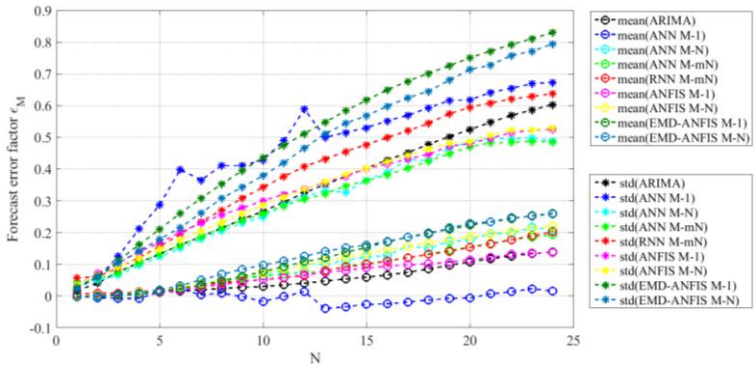
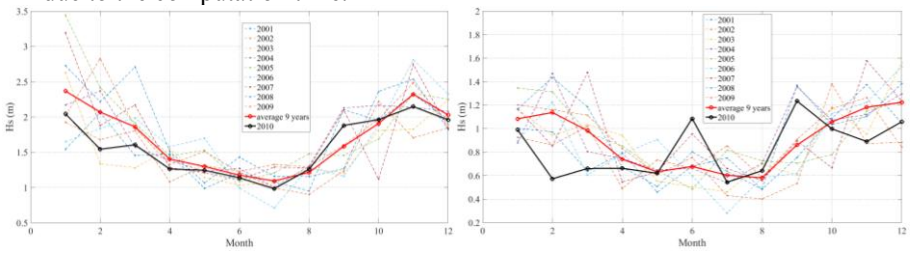


Fig. 14. Comparison of forecast uncertainty based on machine-learning methods

Based on the results shown in Figure 14, the following can be observed.

1) ANFIS-based method with double pre-processing techniques results in the worst performance, no matter which multi-step-ahead model is adopted. This indicates that the prediction performance is very sensitive to the nature of the data and using multiple pre-processing techniques may not be helpful for generating predictions, at least for the study area.
2) Beside these two methods, it is observed that the standard deviations of ε_M for RNN-mN model without any pre-processing techniques (red line) are higher than those of other hybrid methods with the decomposition technique. This implies that it will be helpful to use the pre-processing technique in the time series forecasting. Meanwhile, ARIMA model (black line) exhibits higher forecast uncertainties at longer forecast horizons than the remaining machine learning methods. This may because the linear function employed in the ARIMA model have difficulty in capturing the rapidly changing waves to a certain degree, especially when the forecast horizon is relatively long.
3) Furthermore, there is not a clear indication of which of the remaining three ANN-based methods and two ANFIS-based methods is the best one in multi-step-ahead prediction of H_s. Specifically, the ANN M-1 method has lower mean values and higher standard deviations of ε_M than other four methods. While the accuracies of the other four models, namely ANN M-N, ANN M-mN, ANFIS M-1 and ANFIS M-N model are similar. For example, from all methods, the means and standard deviations of ε_M in the first ten steps are less than 0.1 and 0.3 respectively, and around 0.2 and 0.5 at the 24th step. By comparison, the ANFIS M-1 method (pink) provides the best results for the study time series, since its forecast uncertainty is slightly lower over long forecast horizons. Applying this method, the means and standard deviations of ε_M at 1st, 10th, 20th and 24th step ahead are -0.003, 0.052, 0.114, 0.137 and 0.032, 0.300, 0.480, 0.524, respectively.

4) In general, Figure 14 illustrates that the forecast uncertainty would increase with the forecast horizon, no matter which hybrid machine-learning method is adopted. This is a reasonable observation. On one hand, there might be no correlation in the data within the time horizon of 7-12 and more steps. On the other hand, it's more difficult to develop an accurate prediction model when the forecast step is large. In addition to the above reasons, the errors caused by the decomposition technique are also accumulated in the final prediction results. This can be observed from Figure 15, where the average of the monthly mean values and standard deviations over the previous nine years (red line) do not represent well the statistics of the tenth-year data (black line), especially for the monthly standard deviations. The impact on one-step-ahead prediction may be small but may increase as the number of steps increases. This implies that the data considered in the work is very unstable, and more years of data may be required for data pre-processing.
5) It should be noted that there are some limitations in this study. The structures of prediction methods developed in this study are very simple, which is reflected in fewer hidden layers and neurons in ANN- and RNN-based models, fewer fuzzy sets of input variables and simple MFs in ANFIS-based models and so on. Since the selection of neural network type and parameters may greatly affect the performance of weather forecasting, it is possible to select structures and parameters through systematic analysis and establish more appropriate models. In addition, only small M values are considered in ANFIS-based methods in this study, due to the computation time.



(a) Mean value (b) Standard deviation

Fig. 65. Monthly values of statistics of significant wave height

7 Conclusion

Forecasting of weather conditions is of great importance to marine operations. However, due to the unstable and intermittent characteristics of wave and wind data, it is difficult to obtain accurate predictions for future weather conditions, especially when performing multi-step-ahead predictions. This study investigates the performance of different machine-learning methods for multi-step-ahead weather forecasting, based on different data pre-processing techniques (decomposition technique and EMD), data-driven models (ARIMA, ANN, RNN and ANFIS) and multi-step-ahead models (M-1, M-N and M-mN model). The proposed methods are trained and tested using hourly time series in 2001-2009 and 2010 at the North Sea center, respectively. The statistics

of forecast error factors are utilized for evaluating the forecast uncertainty of the methods. After determining their optimal structure, a comparison of machine-learning methods for multi-step-ahead prediction of wave conditions is made. The results show that all machine-learning methods are difficult to perform prediction over long forecast horizon. Specifically, all hybrid machine-learning methods have better performance for predicting significant wave height from first several steps ahead due to the lower level of forecast uncertainty. However, as the number of forecast step further increases, the forecast ability would decrease significantly. This phenomenon has not improved obviously by changing the prediction model or developing a more complex hybrid prediction method. Under this circumstance, one can conclude that it's hard to get accuracy predictions from pure machine-learning method with only historical time series. In order to improve the forecast quality of weather conditions, unlike the pure machine-learning methods that only take the time series data in consideration, the hybrid method combining both physical process and data-driven model might be considered in the future.

Reference

Akpınar, A., Özger, M., Kömürcü, M.I., 2014. Prediction of wave parameters by using fuzzy inference system and the parametric models along the south coasts of the Black Sea. Journal of Marine Science and Technology 19 (1), 1-14.

Athanassoulis, G., Stefanakos, C.N., 1995. A nonstationary stochastic model for long‐term time series of significant wave height. Journal of Geophysical Research: Oceans 100 (C8), 16149-16162.

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks 5 (2), 157-166.

Berbić, J., Ocvirk, E., Carević, D., Lončar, G., 2017. Application of neural networks and support vector machine for significant wave height prediction. Oceanologia 59 (3), 331-349.

Bishop, C.M., 1995. Neural networks for pattern recognition. Oxford university press.

Booij, N., Ris, R.C., Holthuijsen, L.H., 1999. A third‐generation wave model for coastal regions: 1. Model description and validation. Journal of Geophysical Research: Oceans 104 (C4), 7649-7666.

Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M., 2015. Time series analysis: forecasting and control. John Wiley & Sons.

Brown, B.G., Katz, R.W., Murphy, A.H., 1984. Time series models to simulate and forecast wind speed and wind power. Journal of climate and applied meteorology 23 (8), 1184-1195.

Cadenas, E., Rivera, W., 2009. Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks. Renewable Energy 34 (1), 274-278.

Cassola, F., Burlando, M., 2012. Wind speed and wind energy forecast through Kalman filtering of Numerical Weather Prediction model output. Applied Energy 99, 154-166.

Chang, G., Lu, H., Chang, Y., Lee, Y., 2017. An improved neural network-based approach for short-term wind speed and power forecast. Renewable Energy 105, 301-311.

Erdem, E., Shi, J., 2011. ARMA based approaches for forecasting the tuple of wind speed and direction. Applied Energy 88 (4), 1405-1414.

Flores, P., Tapia, A., Tapia, G., 2005. Application of a control algorithm for wind speed prediction and active power generation. Renewable Energy 30 (4), 523-536.

Group, T.W., 1988. The WAM model—A third generation ocean wave prediction model. Journal of Physical Oceanography 18 (12), 1775-1810.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9 (8), 1735-1780.
Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C., Liu, H.H., 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences 454 (1971), 903-995.
Huang, N.E., Wu, M.L., Qu, W., Long, S.R., Shen, S.S., 2003. Applications of Hilbert–Huang transform to non‐stationary financial time series analysis. Applied stochastic models in business and industry 19 (3), 245-268.
Huang, N.E., Wu, Z., 2008. A review on Hilbert‐Huang transform: Method and its applications to geophysical studies. Reviews of geophysics 46 (2).
Jain, P., Deo, M., 2007. Real-time wave forecasts off the western Indian coast. Applied Ocean Research 29 (1-2), 72-79.
Jang, J.-S., 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics 23 (3), 665-685.
Kamranzad, B., Etemad-Shahidi, A., Kazeminezhad, M., 2011. Wave height forecasting in Dayyer, the Persian Gulf. Ocean Engineering 38 (1), 248-255.
Kavasseri, R.G., Seetharaman, K., 2009. Day-ahead wind speed forecasting using f-ARIMA models. Renewable Energy 34 (5), 1388-1393.
Landberg, L., 1999. Short-term prediction of the power production from wind farms. Journal of Wind Engineering and Industrial Aerodynamics 80 (1-2), 207-220.
LeCun, Y., Touresky, D., Hinton, G., Sejnowski, T., 1988. A theoretical framework for back-propagation, Proceedings of the 1988 connectionist models summer school. CMU, Pittsburgh, Pa: Morgan Kaufmann, pp. 21-28.
Li, G., Shi, J., 2010. On comparing three artificial neural networks for wind speed forecasting. Applied Energy 87 (7), 2313-2320.
Lydia, M., Kumar, S.S., Selvakumar, A.I., Kumar, G.E.P., 2016. Linear and non-linear autoregressive models for short-term wind speed forecasting. Energy conversion and management 112, 115-124.
Mandal, S., Prabaharan, N., 2006. Ocean wave forecasting using recurrent neural networks. Ocean Engineering 33 (10), 1401-1410.
Mohandes, M.A., Halawani, T.O., Rehman, S., Hussain, A.A., 2004. Support vector machines for wind speed prediction. Renewable Energy 29 (6), 939-947.
Olaofe, Z.O., 2014. A 5-day wind speed & power forecasts using a layer recurrent neural network (LRNN). Sustainable Energy Technologies and Assessments 6, 1-24.
Özger, M., Şen, Z., 2007. Prediction of wave parameters by using fuzzy logic approach. Ocean Engineering 34 (3-4), 460-469.
Poggi, P., Muselli, M., Notton, G., Cristofari, C., Louche, A., 2003. Forecasting and simulating wind speed in Corsica by using an autoregressive model. Energy conversion and management 44 (20), 3177-3196.
Qin, M., Li, Z., Du, Z., 2017. Red tide time series forecasting by combining ARIMA and deep belief network. Knowledge-Based Systems 125, 39-52.
Rosenblatt, F., 1958. The perceptron: a probabilistic model for information storage and

organization in the brain. Psychological review 65 (6), 386.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1988. Learning representations by back-propagating errors. Cognitive modeling 5 (3), 1.
Stefanakos, C., 2016a. Fuzzy time series forecasting of nonstationary wind and wave data. Ocean Engineering 121, 1-12.
Stefanakos, C., 2016b. Nonstationary Prediction of Wind and Waves in the Pacific Ocean using Fuzzy Inference Systems, The 26th International Ocean and Polar Engineering Conference. International Society of Offshore and Polar Engineers.
Stefanakos, C.N., Athanassoulis, G., Barstow, S., 2002. Multiscale time series modelling of significant wave height, The Twelfth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers.
Stefanakos, C.N., Athanassoulis, G., Barstow, S., 2006. Time series modeling of significant wave height in multiple scales, combining various sources of data. Journal of Geophysical Research: Oceans 111 (C10).
Stefanakos, C.N., Schinas, O., 2014. Forecasting bunker prices; A nonstationary, multivariate methodology. Transportation Research Part C: Emerging Technologies 38, 177-194.
Takagi, T., Sugeno, M., 1983. Derivation of fuzzy control rules from human operator's control actions. IFAC Proceedings Volumes 16 (13), 55-60.
TAKAGI, T., SUGENO, M., 1985. Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE transactions on systems, man, and cybernetics 15 (1).
Tolman, H.L., 1991. A third-generation model for wind waves on slowly varying, unsteady, and inhomogeneous depths and currents. Journal of Physical Oceanography 21 (6), 782-797.
Torres, J.L., Garcia, A., De Blas, M., De Francisco, A., 2005. Forecast of hourly average wind speed with ARMA models in Navarre (Spain). Solar Energy 79 (1), 65-77.
Wu, M., Stefanakos, C., Gao, Z., 2018. Prediction of short-term wind and wave conditions using Adaptive Network-based Fuzzy Inference System (ANFIS) for marine operations, Advances in Renewable Energies Offshore: Proceedings of the 3rd International Conference on Renewable Energies Offshore (RENEW 2018), October 8-10, 2018, Lisbon, Portugal. CRC Press, p. 83.
Wu, M., Stefanakos, C., Gao, Z., Haver, S., 2019. Prediction of short-term wind and wave conditions for marine operations using a multi-step-ahead decomposition-ANFIS model and quantification of its uncertainty. Ocean Engineering, Submitted.

Forecasting Anomalous Events and Performance Correlation Analysis in Event Data

Sonya Leech[1] and Bojan Božić[2]

[1] IBM, Dublin, Ireland,
leechsy@ie.ibm.com
[2] Technological University Dublin, Dublin, Ireland
bojan.bozic@dit.ie

Abstract. Classical and Deep Learning methods are quite common approaches for anomaly detection. Extensive research has been conducted on single point anomalies. Collective anomalies that occur over a set of two or more durations are less likely to happen by chance than that of a single point anomaly. Being able to observe and predict these anomalous events may reduce the risk of a server's performance. This paper presents a comparative analysis into time-series forecasting of collective anomalous events using two procedures. One is a classical SARIMA model and the other is a deep learning Long-Short Term Memory (LSTM) model. It then looks to identify if an influx of message events have an impact on CPU and memory performance.
The findings of the study conclude that SARIMA was suitable for time series modeling due to the elimination of heteroskedasticity once transformations were implemented, however it was not suitable for anomaly detection based on an existing level shift in the data. The deep learning LSTM model resulted in more accurate time-series predictions with a better ability to be able to handle this level shift. The findings also concluded that an influx of event messages did not have an impact on CPU and memory performance.

1 Introduction

When unusual patterns occur in data this is classified as an anomalous event also known as an outlier. An outlier is a single extreme event. Detecting these anomalous events can be considered a support aid for a variety of different business organizations. It can be used in cyber security to aid to detect cyber attacks [2]. It can also be used in the financial sector for credit card fraud or the betting domain for gambling fraud. It can also aid in intrusion detection for network security or even in census data [5]. Being able to predict when a system or application log message is exceeding the normal operational bounds allows the IT support people become more proactive than reactive to their business process. A collective anomaly is when more than one irregularity occurs consistently over a set amount of observations in a dataset. Our research is based on collective anomalous events.

From this comes a need for an automated anomaly detection tool [2] that can identify rare events or behaviours in data that differ significantly from the norm. These anomalies can come in the form of point, contextual or collective anomalies [2]. Being able to track, control and understand these anomalous events can aid a business in its ability to better handle and control these events. Some of these anomalous events may impact or bottleneck the performance of a server leading to significant cost implications. Such is the case that when Amazon has an additional 100 ms delay in their response times it impacts them by a 1% reduction in sales [4].

Our paper explores models and approaches to address this problem and is structured as follows: Section 2 lists related work in the fields of classification, anomaly detection, and models. Section 3 describes our approach to anomaly detection, while section 4 shows the evaluation of our approach. Finally, section 5 gives an overview of future work and conclusion.

2 Related Work

Anomalous detection can be implemented by looking at points in time. A single point that is distant from the majority of observations can be considered an anomaly. Considerations need to be taken to decide under what conditions a deviation is classified as an anomaly. Different classifications can be implemented, those are point, collective and contextual [7].

Two types of outliers are discussed. Those are the Additive Outlier (AO) [3] and the Innovational Outlier (IO) [1]. The AO outlier occurs over a single observation like a point in time. It is something that may occur due to random chance. The IO outlier is identified when it remains an outlier over several observations. It does not drop back to a normal value until some time has passed [8].

Changes in the structure of data can also be an outlier. Different types of changes exist. Three of these structures are discussed in the paper. Those are the Level Shift(LS), the Variance Change (VC) and Transient Change(TC) [8].

Before data can be modelled, the model parameters (p,d,q) need to be defined. These model parameters can be used as an aid to define which model to use. [6] comments that *"Some methods indicate superior performance when error based metrics are used, while others perform better when precision values are adopted as accuracy measures"*.

3 Anomaly Detection

To find anomalies in data one needs to look at extreme values or values that deviate from the norm that are not reflective of cyclic seasonality or trends. For anomaly detection residual error, principal component analysis, cooks distance and level shift are some of the tools used to determine if data deviating from the norm is an actual anomaly or not. These anomalies are based on unexplained observations and are also known as outliers and both these words are used quite

interchangeably in the research papers. Types of anomalies are a point, contextual and collective. Point anomalies are also known as additive outliers which are defined by [3] and his interpretation on how to capture them is via a likelihood ratio test. These anomalies are a sudden sharp increase in value followed by a sudden change back to normal. Collective anomalies are when a consecutive number of anomalies occur throughout observations also known as transient change outliers. These collective anomalies can be caused by a seasonal shift in the data which is known as a level shift.

Collective anomalies are the scope of this project. Our investigation is to identify collective anomalies and compare them against that of the ARIMA and LSTM models. A simple approach used to detect if anomalies occurred is to evaluate how many points the data deviated from the mean using a standard deviation (STD) function. A twenty-four-hour rolling window for the STD was used. The sigma levels were based on two STD's so that the anomaly was not limited to only identifying really large spikes in the data. Due to their being no domain experts involved, no outliers in the data were removed and data was analysed based on all data points. The residual errors were graphed to see if there was any visual observation of anomalies in the data based on the two standard deviation confidence level. Anomalies were only conducted on Informational type messages due to the constraints of time.

3.1 ARIMA

Figure 1 shows the residual errors from the ARIMA model. It is observed that anomalies have occurred in the model based on the points that deviate outside of the upper and lower two STD confidence interval boundaries. Visually it is hard to tell if collective anomalies have occurred.

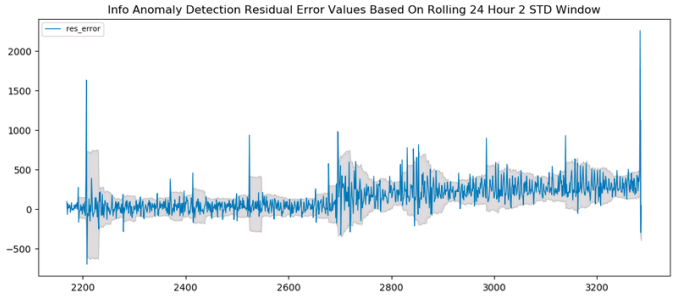


Fig. 1. Info ARIMA Residual Errors

Two graphs have been created. Here collective anomalies have been detected. Two graphs have been plotted. These graphs are filtered to show a reduction in the dataset that is concentrated in showing detected collective anomalies. From figure 2 we can visually see more clearly the anomalies detected. Three collective anomalies have occurred within the full dataset.

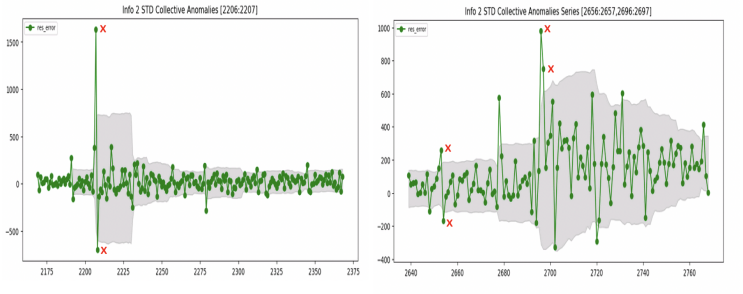


Fig. 2. Info : Two STD Collective Anomalies

When using three STD the amount of point and contextual anomalies detected is 33 and 2 respectively. For two STD it is 43 and 3, while for one STD it is 65 and 6. As expected there is more one STD anomalies than that of two and three STDs. Two STDs for Informational type messages observed three collective anomalies between series 2206 and 2207 and series 2652 to 2653 and series 2696 to 2697.

There is significant variance in the STD around series 2220. The data was then further analysed to see if some sort of a pattern existed that caused the significant spike to occur. We can see that the data reached its peak very sharply over one hour and was not, in fact, a gradual incline. It may be determined that this is due to missing data and a further check was done to determine if the data was indeed missing.

Series	Date	Value
2206	2019-03-26 14:00:00	689.0
2207	2019-03-26 15:00:00	2115.0
2208	2019-03-26 16:00:00	420.0

Table 1. Info Anomaly Detection Missing Data Check for Spike

We can see from table 1 that this is not the case, that there was, in fact, no missing data for that period. A domain expert would need to asses this incline

to give a better indication as to the reason for the significant increase.

3.2 LSTM

The residuals of the LSTM model were graphed in figure 3 with the two STD boundaries added. Most of the residuals are centered around zero except for the residuals near-series 900. The residual graph appears stationary and does not show the level shift that occurs in the ARIMA residual graph 1.

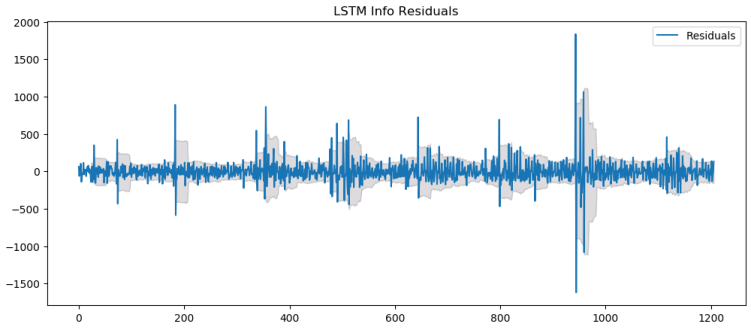


Fig. 3. Info LSTM Residuals

Anomalies have been plotted with an x on figure 4. Eighty-four point and fifteen collective anomalies have been detected. For each collective anomaly detected all's it anomalies have been plotted. Out of the fifteen collective anomalies detected thirteen occurred within a two-hour window and two occurred within a three-hour window. The green x's represent the three-hour window and the red x's indicate the two-hour window.

Anomaly Comparison

For ARIMA it detected three collective anomalies while LSTM detected fifteen.

4 Evaluation

For the initial two months of the data table 2 shows that a total of 1.5 million event type messages were produced. Out of those messages the error type events produced the highest number of events equating to a total of 55.5%.

For any analysis herein the alpha will be 0.05

4.1 Normality

Distribution analysis was done for each of the severity type messages. The first graph to the left in figure 5 shows info type messages not being of a normal

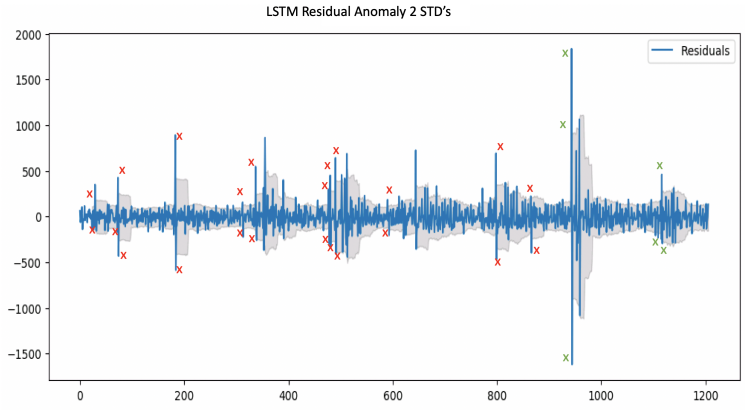


Fig. 4. Info LSTM Collective Anomalies On The Residuals

	Total	**Percent**
Total	1,574,682	100%
Info	560,828	35.62%
Error	874,336	55.52%
Warn	139,518	8.86%

Table 2. Count of Aggregated Log Data

distribution. The graph displays a platykurtic kurtosis with positive right-tailed skewness. Its quantile plot underneath it does confirm that the data is not normally distributed but does observe some fitting on the regression line. We also observe some outliers in the data. The middle and right graphs which show the warn and error distributions indicate a very volatile dataset due to the high volume of low counts of messages and a small volume of high count messages. The three quantile plots show that the data is not of a normal distribution for each of the severity type events.

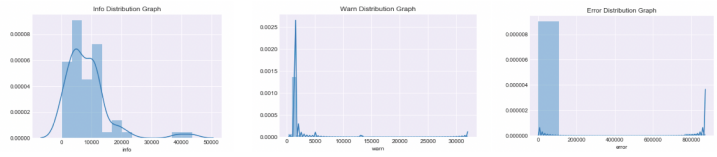


Fig. 5. Info, Warn, Error Daily Distribution Analysis

Fig. 6. Daily Quantile Plots

For data to be of normal distribution its skewness should be zero and its kurtosis should be three. As per table 3 info, warn and error do not conform to the skewness and kurtosis values to be of a normal distribution.

	Skewness	**Kurtosis**
Info	2.5	9.1
Warn	6.7	48.7
Error	34.7	1261

Table 3. Daily Skewness-Kurtosis

SW and AD normalcy goodness of fit tests were conducted on the data.

Log Type	**Test**	**Test Statistic**	**P Value**
Info			
	SW	0.7	0.0
	AD	3.2	1.0
Warn			
	SW	0.1	3.2
	AD	21.6	1.0
Error			
	SW	0.0	0.0
	AD	585.9	1.0

Table 4. Daily Goodness Of Fit Tests

SW Test
Null Hypothesis: The data is normally distributed.
Alternative Hypothesis: The data is not normally distributed.
If p-value ¡ 0.05 reject the null hypothesis. The data is not normally distributed.

AD Test
Null Hypothesis : The data is normally distributed.

Alternative Hypothesis: The data is not normally distributed.
Critical values [10%: 0.62, 5% : 0.74, 1% : 1.03]
If test statistic ¿ critical values : Reject the null hypothesis the data is not normally distributed.

Normalcy Results
Info :
We reject the null hypothesis of the SW test p=0.0. There is statistical evidence to suggest the data is not of a normal distribution. With the AD test (test statistic =3.2 ¿ 5% at 0.74) we reject the null hypothesis. The data is not normally distributed.

Warn :
We fail to reject the null hypothesis of the SW test p=3.2. The AD test (test statistic =21.6 ¿ 5% at 0.74) is showing strong evidence to suggest that the data is not normally distributed.

Error :
We reject the null hypothesis of the SW test p=0.0. There is statistical evidence to suggest the data is not of a normal distribution. The AD test (test statistic =585.9 ¿ 5% at 0.74) is showing strong evidence to suggest that the data is not normally distributed.

4.2 Seasonality & Trends

Trend and seasonal graphs were created for info, warn and error type events. STL decomposition was done with the frequency set to weekly using the additive model. A monthly period was ignored due to the lack of initial data for analysis.

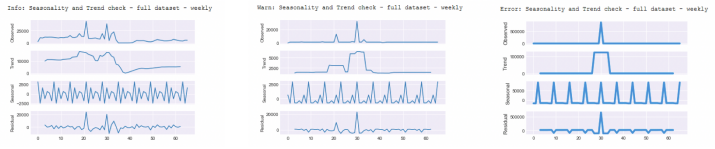


Fig. 7. Daily Seasonal Decomposition Analysis

As per table 7 we visually observe that trend and seasonality do exist in the dataset. The graphs are displayed in order of observed, trend, seasonality and residuals. The trend is shown in the 2nd graph of the grouped graphs. For trend info type events do show a variance change while warn events to show a transient type change and error events show the same as warn but not as apparent. Seasonality is shown in the third row of the grouped graphs and there

does seem to be a repeat pattern over the time series. These patterns become more apparent when higher levels of frequency are used. A correlogram was also created to identify trends in the dataset.

4.3 Correlation

Pearson's correlation analysis was implemented on the daily data to see if any of the event types have any type of relationship with each other. It is observed from figure 8 that info type events have a very strong correlation with warn events (0.7). Info events also have a strong correlation with error events (0.5). Error and warn events do show a significant correlation with each other of (0.9). The results of the Pearson's test conclude that there is strong statistical evidence of relationships between each of the different event types.

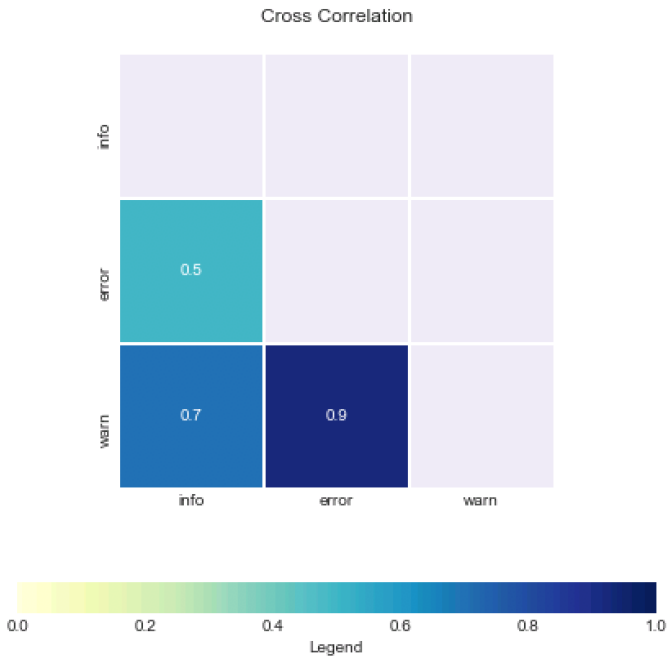


Fig. 8. Daily Pearson Correlation Test : Info : Warn : Error

For Daily data, it was observed that 55% of the events were generated by the error severity event and only 8% were generated by the warn severity event. A 55:35 split was detected between error and info event types. From these statistical counts, it would appear that an error event may have occurred over a

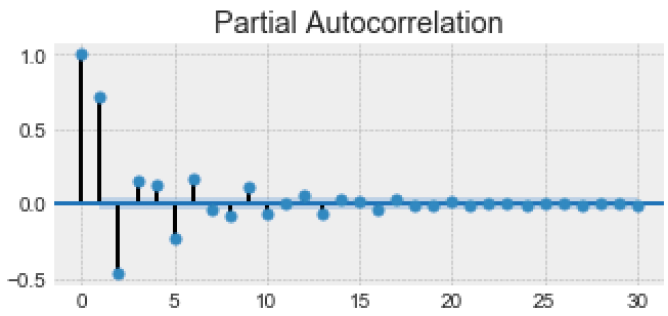


Fig. 9. Warn PACF First 30 Lags Filtered Observation

considerable amount of time that caused it to surpass the info type message count. Observationally from these values, it would appear that no correlation exists between the warn and error type events or it may be the case that the error events that occurred may have been stuck in an iterative loop over a considerable period.

For time series modelling we need to conclude from the data if it fits a certain pattern or shape. The results of these tests may indicate the need for further tests or transformations to be done before the data can be modelled. Those types of tests are normality, unit root, stationarity, volatility, trend, seasonality and time series dependence tests. The majority of these tests have been conducted on the daily data. In the following we will show the evaluation of warnings.

Testing For Normality:
Warn type events did display volatility in the data. With the SW test p=3.2, we fail to reject the null hypothesis, the data is normally distributed. The AD test (test statistic =21.6 ¿ critical value at 5% = 0.74) rejects the null hypothesis. There is evidence to suggest the data is not normally distributed. Skewness = 6.7 shows a heavy right-tailed distribution with a leptokurtic kurtosis = 48.7 both of which indicates variance in the data. The histogram and the quantile plot show that the data is not of a normal distribution as it does not fit anywhere along the regression line and the majority of the values in the histogram occur within the zero to one thousand range. Based on the combined tests the evidence is conflicting. If the low number of high outliers were removed from the dataset this may change the results of the skewness and kurtosis test. It may also change the shape of the histogram and the distribution of the fit along the regression line. Further analysis would need to be conducted with the outliers removed to see if they occurred by random chance and are not seen to be part of the normal observation.

Testing For Stationarity:
Testing for stationarity the ADF unit root test p=0.00 implies that the time

series has no unit root and is stationary. For KPSS unit root (test statistic =0.12 ¡ critical value 0.46) provides evidence to suggest that the time series is stationary so we fail to reject the null hypothesis. A high degree of variance and mean are an indication that the time series is not stationary. Using the statistical KPSS and ADF tests their is strong evidence to suggest that the warn type event data is stationary. The high variance and mean in the data may be partially due to the high outlier values detected in the dataset.

Testing For Trend And Seasonality:
A transient type of change was observed in the trend chart. Seasonality does exist over repeat observations. The correlogram does not show any trend or seasonality. The CH test failed to detect seasonality or trend in the dataset. Based on the visual and statistical evidence more tests will need to be conducted on the data to provide more solid reasoning for accepting or rejecting the hypothesis that seasonality and trend exist.

5 Conclusion and Future Work

A Box-Jenkins SARIMA model and a highly sophisticated neural network LSTM model were analyzed. Log messages with a severity type of info, error and warn was tested. SARIMA was tested on untransformed data, 1st difference, natural log, and square root transformations. Different parameter factors were taken into consideration before deciding which model to use. Those factors came from the results of the unit root, normality, heteroskedasticity, time series dependency, and seasonality tests. RMSE was used for the model accuracy measures. A 1st difference transformation was applied to the LSTM model.

When testing for seasonality it was evident from the results of the test in comparison to the results of the STL tests that the CH test was not able to detect seasonality or trend for the majority of cases. It was, however, a little better at predicting seasonality at the higher frequency level for monthly data over the hourly periods. The CH for seasonality needs the first transformation to be done on the data before it can be applied. It is not able to handle higher level seasonal dimensionality in the data. This test should be eliminated from the study as it was not the best tool of choice. It was unfortunate that the limitations of the CH test were not evident in the research papers first read and only after questioning the results of the tests did we find the necessary papers. For time-series prediction the results of the models concluded that the SARIMA model was not suitable for modeling predictions due to the existing shift in the data after the first principle transformation was done. The LSTM model was far more superior and better suited to handle the shift in the data. It is recommended that a further transformation is done on the data to remove the existing seasonality or trend in the data.

A rolling twenty-four window two STD approach was used for anomaly detection. The LSTM model was able to better predict anomalies than that of

SARIMA. It is recommended that a better-suited algorithm that supports a level shift in the data should be implemented like LS or TC. Other recommendations would be to try Principal Component Analysis (PCA) or Cooks distance. A hybrid model of SARIMA and LSTM could be implemented so that the classical model can be able to better handle the seasonality in the data. As only info type events were analyzed for anomalies error and warn event type events should be tested in future studies.

The correlation on info type messages, CPU and memory was quite low and the evidence suggested that this should be rejected. Further correlation tests on CPU, memory and disk usage should be tested against the warn and error type events. There were thirty-two CPU's on the server. Correlation analysis should be further refined by looking at the correlation between each CPU and each log event type message as an overall percentage metric might hide a potential load on these anomalous events.A Pearson correlation test was used for the analysis. As the CPU metric did display seasonality while the info type events also displayed non-stationarity it would be better if a Kendal or Spearman's correlation was implemented instead.

References

1. Nathan S. Balke. Detecting level shifts in time series. *Journal of Business And Economic Statistics*, 11(1):81–92, 1993.
2. Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):7, 2009.
3. A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society Series B (Methodological)*, 34(3):350–363, 1972.
4. Olumuyiwa Ibidunmoye, Francisco Hernandez-Rodriguez, and Erik Elmroth. Performance anomaly detection and bottleneck identification. *ACM Computing Surveys*, 48, 06 2015.
5. C-T Lu, Dechang Chen, and Yufeng Kou. Algorithms for spatial outlier detection. In *Third IEEE International Conference on Data Mining*, volume ICDM'03, pages 597–600. IEEE, IEEE Computer Society, 2003.
6. Nijat Mehdiyev, David Enke, Peter Fettke, and Peter Loos. Evaluating forecasting methods by considering different accuracy measures. *Procedia Computer Science*, 95:264–271, 12 2016.
7. Karanjit Singh and Shuchita Upadhyaya. Outlier detection: Applications and techniques. *International Journal of Computer Science Issues*, 9, 01 2012.
8. Ruey S. Tsay. Outliers, level shifts, and variance changes in time series. *Journal of Forecasting*, 7(1):1–20, 1988.

GPU forecasting for big data problems.

Juan R. Trapero[1⋆], Enrique Holgado[1], Francisco Ramos[2], and Diego J. Pedregal[1]

[1] University of Castilla-La Mancha,
Department of Business Administration. 13071 Ciudad Real, Spain
[2] University of Castilla-La Mancha, Department of Electrical, Electronics, Control and Communications Engineering, 13071 Ciudad Real, Spain

Abstract. High Performance Computing via General Purpose Graphical Processing Unit (GPU) is a potential instrument to speed up computational times. In a world where big data is becoming a revolution, GPU could play an important role. This work intends to analyze the performance of GPU by implementing the calculation of probabilistic forecasts based on single exponential smoothing in conjunction with simulated predictive distributions. Essentially, supply chain companies must deal with a high number of forecasts at SKU level. In this context, reducing the computational times can be a source of a competitive advantage. Since the forecasts are usually made independently between SKUs, this problem can be easily parallelized and GPU computing can exploit such parallelization. To the best of authors knowledge, this is the first time GPU is applied to a supply chain demand forecasting context. Firstly, we will show how to adapt the programming of probabilistic forecasts in a parallel fashion. Then, real data coming from a manufacturer company will be used to illustrate the differences between GPU and traditional CPU computing. The results show that GPU can significantly increase the computational speedup ratio more than 30 times with respect to traditional CPU computing.

Keywords: forecasting, GPU, big data, supply chain management

1 Introduction

The world digitalization is generating a massive amount of data that can be obtained from different sources [2]. Dealing with such an amount and variety of information has challenged the traditional data analysis methods, [2, 4]. Therefore, organisations have to cope with the "Big Data" concept [1].

One of the main improvements that big data may bring in a supply chain context is more accurate forecasts, which implies improving customer service levels, while lowering inventory costs, waste, and working capital [3]. Authors

⋆ This work was supported by the European Regional Development Fund and Spanish Government (MINECO/FEDER, UE) under the project with reference DPI2015-64133-R.

in [5] carried out a literature review about big data analytics in supply chain management and concluded that only 3 papers out of 88 are centered on demand forecasting.

Authors in [6] show that less complex optimization routines more oriented to face big data problems do not reduce significantly the forecasting accuracy, although it does improve the computation speed. The necessity of a compromise between computational speed and forecasting accuracy is also pointed out in a retail industry context in [7]. For instance, Seaman in [7] quantifies the forecasting problem for Walmart with upwards of a trillion forecasts being needed, calculating over 10 millions forecasts a second. In this sense, [7] indicates the "parallelizability" of the forecasting models used as a key factor to improve the speed of forecasting, as well as, the underlying computational infrastructure to support the parallelization.

This work aims at exploring the use of GPU to improve the computational speed of calculating probabilistic forecasts via simulated prediction distributions. Essentially, we show how the forecasting method should be implemented in a parallel fashion to take advantage of the potential benefits of GPU. We particularize our results for a well-known method as single exponential smoothing for point forecasting and simulated prediction distributions through Monte Carlo experiments for variability forecasts. These techniques will be implemented in MATLAB using "built-in" functions to circumvent the use of CUDA language. Thus, simply usual MATLAB coding is necessary to apply the proposed approach.

2 Probabilistic Forecasts

The probabilistic forecasts are calculated as follows: First, shipments information for n SKUs obtained from a real company are used to compute the point forecast by using a single exponential smoothing. Second, with the point forecasts, the mean and variance of residuals are computed and fed into the Monte Carlo experiment which generates m replications. Third, the calculation of the probability forecasts is obtained based on those replications.

3 Results

In our dataset we have a total of 173 SKUs sales data. Out interest is to analyze the computational times for a bigger number of SKUs. To do that, we have resampled the original dataset to obtain as many SKUs sales data as we need on the basis of the original dataset.

Table 1 shows the speedup ratios calculated for different volumes of SKUs from 100 up to 5000, where the speedup ratio is computed as CPU time required to do the calculations divided by GPU time counterpart, such as: $speedup = CPU_{time}/GPU_{time}$.

The first row shows the speed up ratio of the MATLAB function **fmincon** run on a CPU with respect to the grid and search (with a number of steps defined

by $n_a = 20$) run in a GPU. Such a ratio varies between 32 and 42, that is, the GPU option was more than 30x times faster than the CPU alternative. The second and third row computes the speed up ratio using the same optimization technique (grid and search) with different values of n_a. For values of $n > 100$ a lower value of n_a provides higher speed up ratios.

It is interesting to note that there is a value of n that maximizes the speed up ratio and it is $n = 1000$. That is an interesting fact, since, a priori, one could wrongly think that the higher the value of n, the higher the speed up ratio of the GPU.

This is an important conclusion that can be utilized when scaling this problem. Essentially, for a big data framework where the number of SKUs may compromise the memory size, MATLAB allows to work with different GPU cards (in parallel[3]) and this experiment shed some light about how many SKUS should be distributed in a GPUs cluster to obtain the maximum speed up ratio, i.e., the lower computational times.

Table 1. Speed up ratios calculated for different values of n and rounded to the nearest integer.

Speed up	n			
	100	1000	2500	5000
fmincon	32	42	40	38
grid and search (n_a=40)	28	39	37	34
grid and search (n_a=20)	29	41	40	36

4 Conclusions

This article presents a methodology to implement in a GPU the calculation of probabilistic forecasts in a parallel manner, where the point forecasting algorithm is a widely used forecasting technique as SES and the quantile forecasts can be obtained empirically via simulated predictive distributions.

To the best of authors' knowledge, this is the first time that GPU computing is used for the problem of supply chain forecasting. The main result show that GPU computing obtains similar forecasting accuracy in a reduced computational time.

References

Addo-Tenkorang, R., Helo, P.T.: Big data applications in operations/supply-chain management: A literature review. Computers & Industrial Engineering 101, 528 – 543 (2016)

[3] Recent versions of MATLAB permit to use the **parfor** command on GPU cards

Blazquez, D., Domenech, J.: Big data sources and methods for social and economic analyses. Technological Forecasting and Social Change 130, 99 – 113 (2018)
Chase, C.W.: Using big data to enhance demand-driven forecasting and planning. The Journal of Business Forecasting 32(2), 27–32 (Summer 2013)
Gandomi, A., Haider, M.: Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management 35(2), 137 – 144 (2015)
Nguyen, T., Zhou, L., Spiegler, V., Ieromonachou, P., Lin, Y.: Big data analytics in supply chain management: A state-of-the-art literature review. Computers & Operations Research 98, 254 – 264 (2018)
Nikolopoulos, K., Petropoulos, F.: Forecasting for big data: Does suboptimality matter? Computers & Operations Research (2017)
Seaman, B.: Considerations of a retail forecasting practitioner. International Journal of Forecasting (2018)

Could the supply of a chain big data analytics market register a better forecast performance for the Stock Markets? – A comparative software analysis

Diana A. Mendes, Nuno B. Ferreira, Vivaldo M. Mendes

ISCTE Business School and BRU-IUL, Av. Forças Armadas,
1649-026 Lisboa, Portugal.
Tel: 351-217-903-000.
E-mail: nuno.ferreira@iscte-iul.pt

Abstract: The dimension of the finance industry that has been most influenced by technological advances in the last decade it is the speed and frequency with which financial transactions are decided and executed. Data analytics, business intelligence, machine learning, algorithmic trading are the leading tech trends of this decade and the most active and innovative segments of the information technology market. In consequence, we can assist in a quick development process to explore new methodologies and to extract and analyse the rich information that the big data market sets contain. For example, in 2019, the biggest U.S. enterprises tend to migrate their data to hybrid cloud solutions helping agents to centralise management of big-data assets distributed between private and public clouds.

In this paper, we use the G7 indexes stock markets prices for periods between 10 and 50 years of daily historical data (from Yahoo Finance). The purpose of the analysis it is twofold: first - several forecasting methods are applied, and we search for the minimum forecasting error, second – we are interested in the time duration and the software velocity attained in the forecasting process.

For the first purpose, we use supervised deep learning methods, namely Recurrent Neural Networks and Long Short-Term Memory (LSTM) architectures. We have found that the LSTM configuration works the best out of all the combinations we have tried (for our dataset). LSTMs are very powerful in sequence prediction problems because they can store past information, which is crucial in predicting its future price. The forecast error is measured by the Mean Absolute Percent Error and by the Mean Square Error. We use Python software, where Keras, TensorFlow and Pandas are the packages with the main role.

The second important point that we analyse in this article it is the execution time. We use different software, namely R, Julia and Python, and we aim to measure the trade-off between the algorithm complexity and the speed of execution. A comparison of the below results with ARIMA models forecast accuracy and execution time it is also considered.

Photovoltaic Power Forecasting Using Back-Propagation Artificial Neural Network*

Hamza COUSCOUS[1] Abderrahman BENCHEKROUN Khaled ALMAKSOUR
Arnaud DAVIGNY[2] Dhaker ABBES[3]

LABORATORY OF ELECTRICAL ENGINEERING AND POWER ELECTRONICS (L2EP)
HEI Yncrea HDF, 13 Rue de Toul, 59014 Lille Cedex, France

[1]hamza.couscous@yncrea.fr, [2]arnaud.davigny@yncrea.fr,
[3]dhaker.abbes@yncrea.fr

Abstract. Known for being a reliable alternative, microgrids have been widely deployed recently in power distribution field in order to guarantee a constant power supply especially in isolated zones. Moreover, microgrids have increasingly known the penetration of renewable energy as environmentally friendly energy sources. However, the intermittency of these sources oblige specialists to think about tools allowing determining their potential, over a predetermined time interval, in order to ensure energy security. For this reason, renewable energy forecasting is crucial. Thus, in this paper, a feed forward back-propagation neural network is used to forecast next 24 hours photovoltaic (PV) power of one of the catholic university buildings ‘‘îlot RIZOMM’’. The accuracy of the model built is evaluated with some performance metrics. Thereafter, prediction results of PV power are compared to those provided by SteadySat, an industrial solution developed by the company *SteadySun*. It is shown that the prediction Mean Absolute Errors (MAEs) of the model are of 3.05% in a clear sky day, 4.95% in a cloudy day and 5.98% in a partly cloudy day.

Keywords: Artificial Neural Network; back-propagation; forecasting; photovoltaic power.

1 Introduction

Fossil energy production causes important CO2 emissions that contribute significantly in climate change phenomenon. Considering also the depletion of fossil energy, several renewable energies have been developed to replace it. The idea behind this new orientation is to ensure the access to a clean and environment friendly energy with a reasonable price.
Nevertheless, most of renewable resources are known to be intermittent and unstable; the energy produced using these means depends totally on the weather conditions. Given that affordable efficient storage technologies are not well developed yet, managing the energy produced accordingly with the load has become a necessity in order to

*The data used in this paper for PV production come from the power plant of ‘’îlot RIZOMM’’ demonstrator located on the roof of the Catholic Institute of Lille (ICL).

minimize losses and ensure once again an equal access to energy at any time. In addition, power reserve is strongly needed to guarantee the balance between energy supply and consumption when any malfunction occurs [1]. Thus, forecasting both energy production and consumption has become mandatory in order to achieve good energy management and continuity of service.

As a response to the urgent need of a better energy scheduling strategy, several models and techniques have been developed to predict renewable energy production as well as load consumption. They can be divided into three categories: Conventional methods, AI-based methods and hybrid methods [2].

Hybrid methods, defined as combination of two or many different forecasting methods, can give more accurate results than the others, because they gather advantages of the combined models and eventually cancel their weaknesses [2].

Each category works perfectly for a specific application. Known for being the simplest to build and to implement, conventional methods can deal accurately with problems with cyclic aspects, such as energy demand forecasting [3]. In [4], simple linear regression showed great performances to predict hourly residential energy consumption using load history and weather parameters. It is to mention that load's profile is strongly related to seasonal variations, weather conditions and economic factors [2]. Thus, a specific conventional model with specific parameters is required to each situation rather than a general one.

However, conventional forecasting methods have some limits especially in complex non-linear problems with discontinuous data [5]. That is why researchers tried to build and use other non-conventional methods in complex forecasting applications. Several works have been published in the past years. Some of these works were oriented to predict solar radiation/ photovoltaic (PV) power generation [1,6], while others were focused on wind power generation [7]. But otherwise, most of them used a specific AI method known as Artificial Neural Network (ANN), which is the more commonly applied technique in complex forecasting problems.

PV power forecasting for instance is one of the most complex problems to solve, since it is affected by many weather factors, such as solar irradiance, temperature, dew point, wind speed, cloud coverage and others [5]. In other words, the accuracy of PV power forecasting is related mainly to the accuracy of weather observations and predictions, which is still difficult to entirely achieve especially for some parameters such as cloud coverage [6].

This work introduces an ANN model to forecast next 24 day hourly PV power production of the "îlot RIZOMM" using meteorological parameters. The performance of the model is evaluated based on root mean square error and mean absolute error. The output of the model is also compared to the predictions made by the industrial solution SteadySat, a technology developed by the company *SteadySun* to forecast solar power based on satellite imagery [8].

The paper is structured as follows: Section 2 presents the feedforward neural network topology with back-propagation as training algorithm, in addition to the general procedure of implementation of this kind of neural networks; Section 3 applies the described ANN topology to the case of the PV plant of "îlot RIZOMM", and presents the

results that are thereafter being discussed and compared to the predictions of SteadySat; lastly, Section 4 presents the conclusions and some perspectives.

2 Back-propagation ANN

ANN is a non-linear method, fully inspired from the human brain, that aims to imitate natural intelligence in order to solve complex problems. ANN is composed of a set of neurons that interconnect the layers (input layer--hidden layer--output layer) via different weights. During the learning phase, the connection weights are self-adjusted in order to fit the most the existing sample data and thus model accurately the relationship between inputs and outputs. Then, once properly trained, the ANN constructs a non-parametric input-output map which gives it the ability to make very accurate predictions using new input data without generating explicit mathematical expressions [9,6]. ANN has been used in a large range of applications, including forecasting, pattern recognition, classification, etc. [6]

Several architectures of ANN exist such as: feedforward neural network that allows information to travel only from input to output with no feedbacks, and recurrent neural network that can have signals floating in both directions by looping computed outputs, from earlier inputs, back into the network. In this paper, a three-layer feedforward neural network structure is considered as shown in Figure 1.

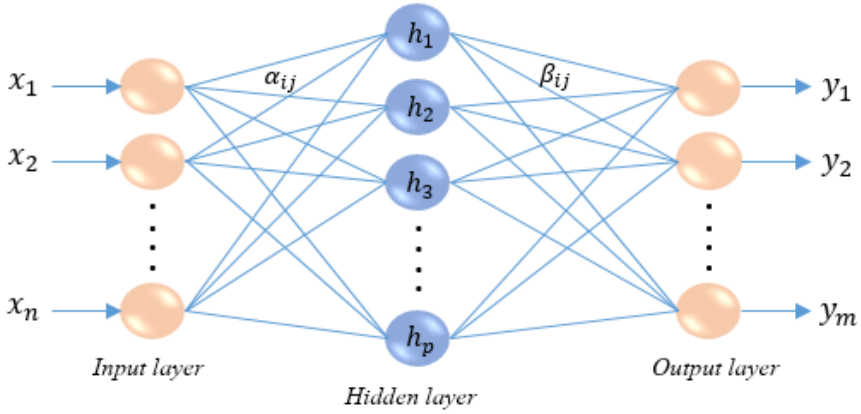


Figure 1: Three-layer feedforward BP network configuration

To explain the relationship between different layers of an ANN, let's consider $I = (x_1,x_2,\ldots,x_n)^T$ as the network's input, $O = (y_1,y_2,\ldots,y_m)^T$ as the network's output, $A = (h_1,h_2,\ldots,h_p)^T$ as the hidden layer variables and (α_{ij},β_{ij}) as the connection weights. Inside the ANN, a set of calculations are performed according to equations (1) and (2):

$$h_j = f_1\left(\sum_{i=1}^{n} \alpha_{ij}\, x_i\right), \qquad j = 1, \ldots, p \qquad (1)$$

$$y_j = f_2\left(\sum_{i=1}^{p} \beta_{ij}\, h_i\right), \qquad j = 1, \ldots, m \qquad (2)$$

Where f_1, f_2 are the activation functions that are often chosen among three types: Linear function, logsig and tansig; described in Table 1, Figure 1 and Figure 2.

Table 1. Mathematical expressions of activation functions

Name	Expression
Linear function	$f(x) = x$
Log-sigmoid transfer function (logsig)	$f(x) = \frac{1}{1 + e^{-x}}$
Hyperbolic tangent sigmoid function (tansig)	$f(x) = \frac{2}{1 + e^{-2x}} - 1$

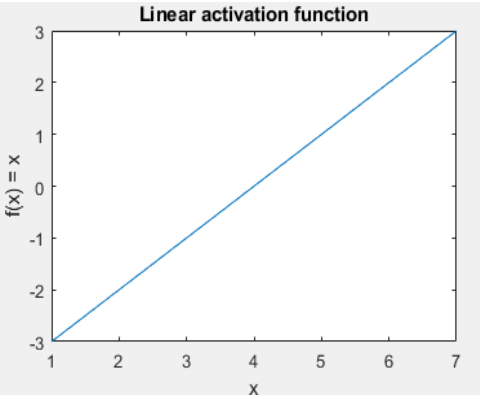


Figure 2: Linear function

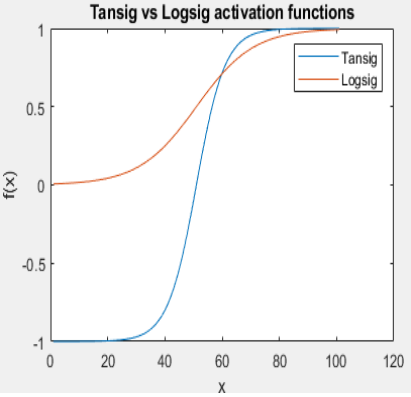


Figure 3: Tansig vs Logsig functions

Hence to solve the forecasting problem, the connection weights, which are the only unknown parameters in the neural network architecture, have to be determined accurately. As a first step, the weights are randomly settled before the network's training, then they are progressively adjusted with iterations in a way to minimize the error between the real targets and network's outputs. This is called the learning process. It is the most crucial step in using neural networks because the final weights will be used later to make predictions using new input data. That is why, this step has to be completed successfully.

To reach the optimal values of the connection weights, several optimization algorithms are used as training algorithms. Based on the gradient descent algorithm, Back-propagation (BP) is one of the most used algorithm in feed-forward neural networks. The principle of BP training algorithm is summarized in Figure 4.

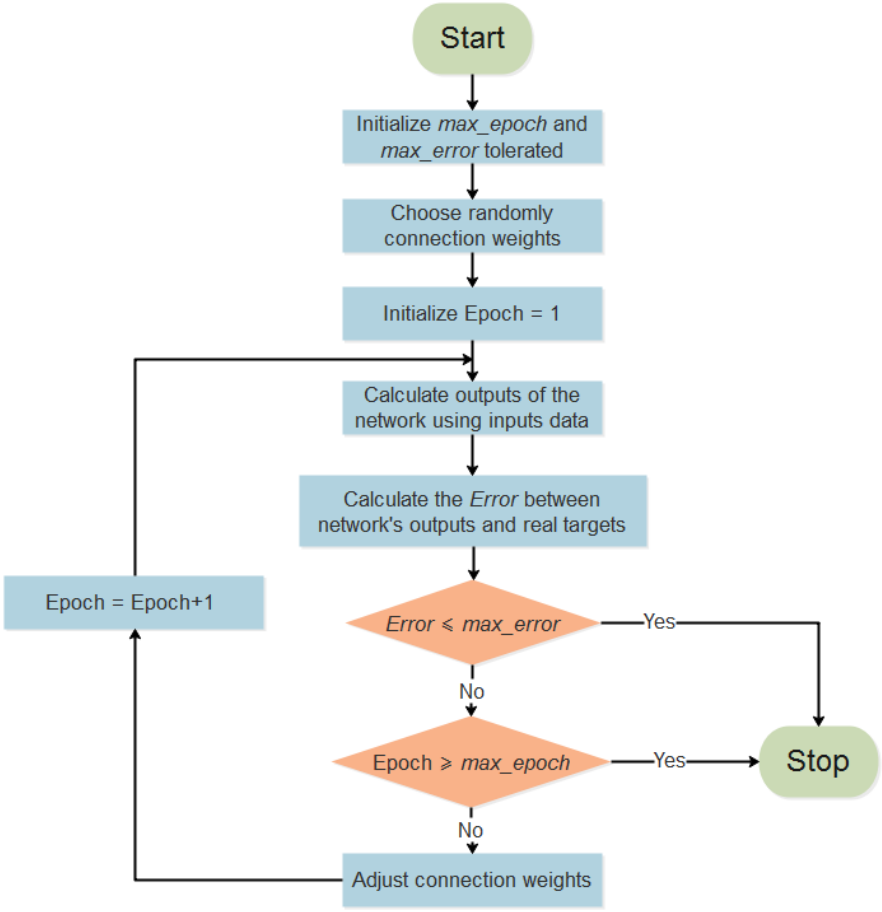


Figure 4: Back-propagation algorithm flow chart

Once the network is created and its parameters are defined, the connection weights are initialized randomly. Then, the error between real targets and network's outputs, which are computed using the equations shown before, is calculated and compared to the maximum error tolerated. The user chooses this error, it depends on the artificial neural network's application field. Thereafter, the connection weights are adjusted as in gradient descent algorithm, in order to get a minimum of the error between targets and computed outputs [6]. This process is repeated until convergence is achieved and/or maximum number of epochs is reached.

In a general way, building a performing neural network passes through several steps as shown in Figure 5. They can be summarized in three main stages:

A. Data preparation

Data preparation is one of the most important stages in solving forecasting problems as it aims to eliminate, to the maximum extent possible, any noise that might affect the ANN's accuracy. ANN are so sensitive to any incoherence in training data, especially when the amount of this last is not very big. That is why an effective and careful data preparation is highly recommended as a preliminary step to using ANN. Two key sub-steps are mandatory:

a. **Correlation study**: which aims to determine the input factors with the highest influence upon the output. These inputs will be considered in ANN training and testing processes. Concretely, correlation coefficient between input $X = \{(x_i) \mid 1 \leq i \leq N\}$ and output $Y = \{(y_i) \mid 1 \leq i \leq N\}$ is calculated through the equation (3).

$$\rho = \frac{\sum_{i=1}^{N}(x_i - \bar{x}).(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}} \qquad (3)$$

b. **Data manipulation**: which aims to prepare properly the input vector of the network. This sub-step differs from one application to another. For instance, in the case of time series inputs, the same data source and time scale need to be respected. However, data normalization to the range of [-1,1] is a common point to all ANN applications, in order to avoid neurons' saturation during the learning process [6].

B. Set up the ANN

In this stage, the network's parameters are set up: number of hidden layers and number of neurons per hidden layer. Until now, there is no unique analytical method to determine exactly these numbers [10]. Therefore, testing many architectures is generally carried out until the optimal solution is found.

C. Train and test the network

In the first place, the network is trained using back-propagation algorithm explained before. Then, it is tested with new data. Sometimes, the network fits "too much" the training set and may fail to make future predictions reliably; it is the overfitting phenomenon. Thus, in order to be able to detect this problem, the ANN performance must be evaluated at both training and testing stages.

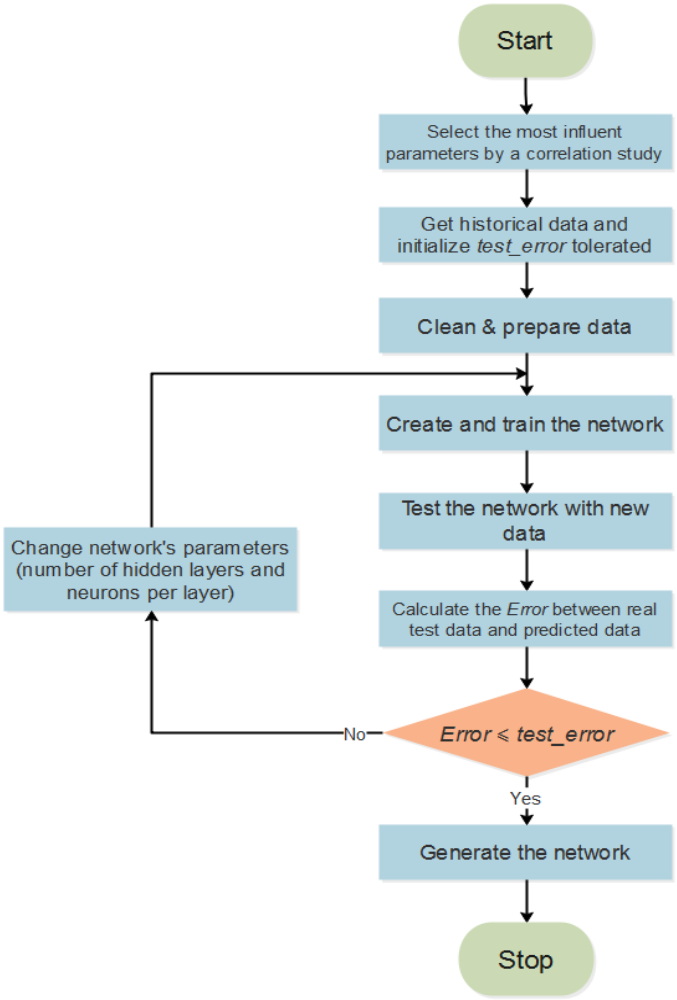


Figure 5: Back-propagation ANN implementation steps

3 Case study

3.1 "Ilot RIZOMM" Project

Since 2012, *Yncréa Hauts-de-France* has shifted towards a new energy policy which is more smart and sustainable. Several projects have been launched concerning renewable energy, electric vehicles and smart grid, Live Tree is one of them. The goal of this project is to connect different sources of renewable energy, consumption points and energy storage technologies via an electrical smart grid. As a part of Live Tree, the demonstrator of the "îlot RIZOMM" was implemented [11]. It is a 189 kWp PV plant, installed over 1114 m² surface on the roof of building in July 2018. Therefore, real

measured PV power data of the RIZOMM demonstrator located on the roof of the Catholic Institute of Lille (ICL) are used to train and test the ANN presented in this paper.

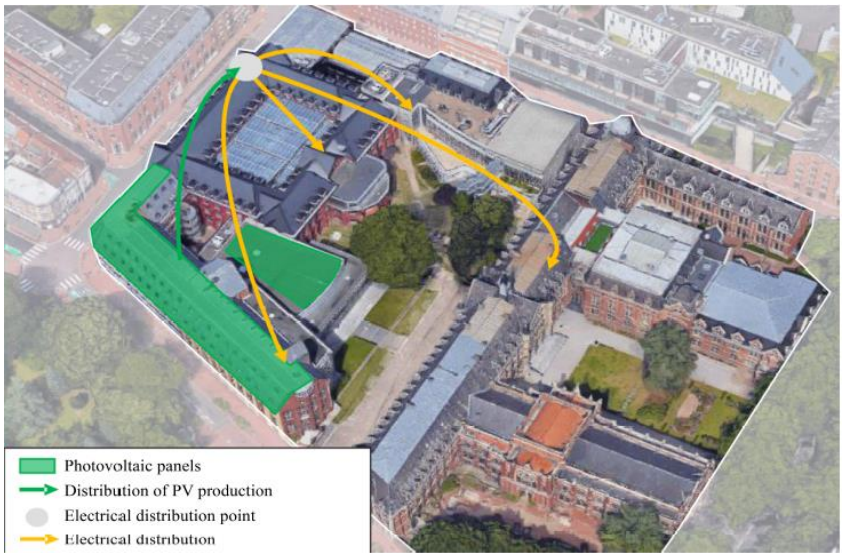


Figure 6: Photovoltaic electrical generator for self-consumption in the "îlot RIZOMM'' at the Université Catholique, Lille.

3.2 ANN architecture

PV power production is related to weather conditions, mainly to solar radiation. To forecast next 24 hours PV power production, several meteorological parameters have been selected at the first place. Then, after the correlation study, only six factors were selected, in addition of the hour of the day n ∈ [|0,23|], to form the input vector I of the neural network: Global Solar radiation (GSR), dry-bulb temperature (T), humidity (H) and dew point (Dwp). The results of the correlation study are presented in Table 2.

Table 2. Correlation study results between meteo parameters and PV power

Meteorological parameter	Correlation coefficient with PV power
Rainfall	-0.0197
Humidity	0.5282
Dew point	0.2408
Wind speed	0.0734
Global solar radiation	0.7525
Atmospheric pressure	-0.0053
Dry-bulb temperature	0.4938

Moreover, given the fact that the peak of PV power production is reached for a panel's temperature of 25°C, it turns out that wind speed (WS) and air temperature can reflect the panels' temperature and thus help the ANN to better understand the relationship between weather conditions and PV power production. For this reason, wind speed has been included in the input vector I even though its correlation coefficient with PV power is low. The final input vector is then:

$$I = [n, GSR, T, H, WS, Dwp] \qquad (4)$$

The ANN built is composed of three layers; input layer, one hidden layer and output layer.

- **Input layer**

Historical hourly data of factors of the vector I and measured PV power of RIZOMM demonstrator for the period from 12/07/2018 to 03/04/2019 are used to train the network. Meteorological data are extracted from [12].

- **Hidden layer**

Only one hidden layer is considered in this model. Trial-and-error method has been used to determine the suitable number of neurons in this layer [1].

- **Output layer**

Next day hourly PV power production represent the output for the network.

In order to evaluate the performance of the model, two error indicators are calculated as function of forecasted values $G_{f,i}$, measured value $G_{m,i}$ and number of observations N: the root mean square error (RMSE) (5) and the mean absolute error (MAE) (6) [6].

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(G_{f,i} - G_{m,i})^2} \qquad (5)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(|G_{f,i} - G_{m,i}|) \qquad (6)$$

3.3 Results and discussion

After training the ANN, its model was tested on three different types of days; a clear sky day, a cloudy day and a partly cloudy day. Meteorological parameters of these days were provided to the network, PV power production of the RIZOMM demonstrator for each day was forecasted and then compared to the real measured one. The results of errors calculation are shown in Table 3.

Table 3. Error indicators of the BP ANN model

Error (%)	Clear sky	Cloudy	Partly cloudy
Train RMSE	10.82	10.21	10.52
Train MAE	5.83	5.16	5.69
Test RMSE	4.86	4.95	10.65
Test MAE	3.05	3.13	5.98

We can see that the model has succeeded in forecasting accurately the hourly PV power for a clear sky day (figures 7,8) with an MAE error of about 3%. For a cloudy day (figures 9,10), the MAE error of PV power prediction doesn't surpass 3.13%. While in a partly cloudy day (figures 11,12), the RMSE error was more important for a partly cloudy day, around 6%. This can be explained by the fluctuating and instable weather conditions, under which solar radiation shows unpredictable changes due essentially to the permanent movement of clouds on the same day.
To validate its efficiency, results of the ANN model of this paper are compared to the predictions of the industrial module SteadySat. This last uses satellite images recovered 1 to 4 times per hour. Combined with weather forecasts, satellite imagery allows the evolution of the cloud cover and the production profile to be refined for the coming hours. Using fine modeling and advanced mathematical algorithms, the energy production for the coming hours is forecasted (6 hours in advance) by SteadySat [8].

Three forecast curves are provided: minimum forecast (P_{min}), maximum forecast (P_{max}) and average forecast (P_{moy}). This last will be used for the comparison of ANN outputs and SteadySat's predictions. SteadySat's prediction errors over one day are shown in Table 4.

Table 4. Error indicators of SteadySat technology power

Day type / Error (%)	Clear sky	Cloudy	Partly cloudy
RMSE	4.58	7.43	11.89
MAE	2.86	4.5	7.12

We can see that the prediction made by the BP ANN model is close to the real PV power and to the prediction of SteadySat on clear sky and cloudy days. However, for partly cloudy days, SteadySat gives a better PV power predicted curve, but in terms of global error on this day, the BP ANN is better as shown in tables 3 and 4. In fact, SteadySat refreshes its input data many times in the same day, by adding the real meteorological parameters and the forecasted ones obtained with a proper weather forecast solution SteadyMet. This makes SteadySat follow better the shape of PV power curve on partly cloudy days.

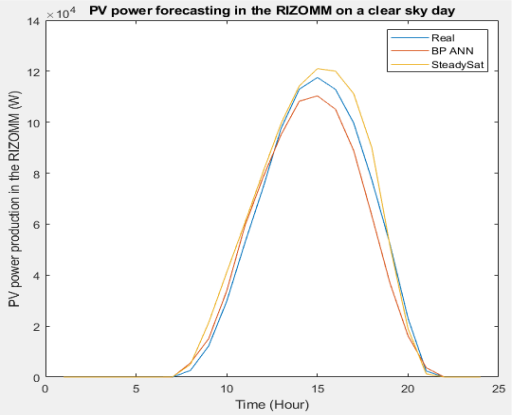


Figure 7: Measured and forecasted PV power on a clear sky day

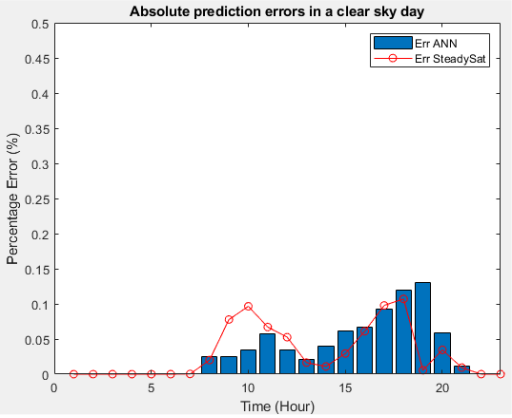


Figure 8: Absolute percentage error of both forecasting models considered on a clear sky day

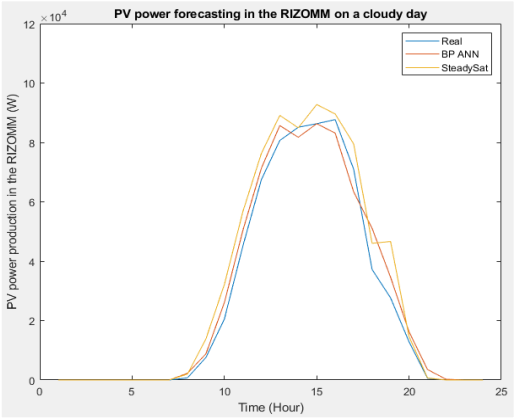


Figure 9: Measured and forecasted PV power on a cloudy day

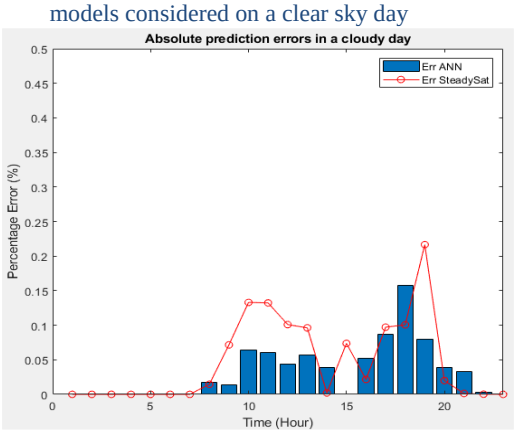


Figure 10: Absolute percentage error of both forecasting models considered on a cloudy day

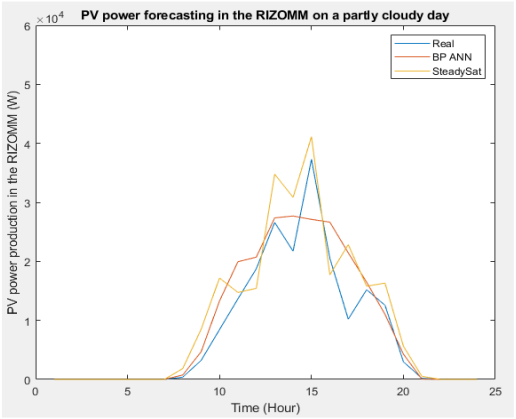


Figure 11: Measured and forecasted PV power on a partly cloudy day

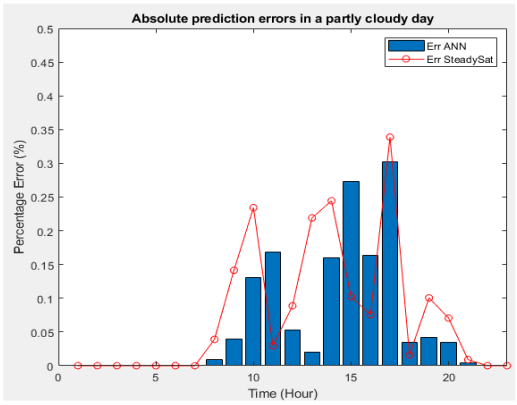


Figure 12: Absolute percentage error of both forecasting models considered on a partly cloudy day

4 Conclusions

Solar energy is one of the most potential green energies that exist today. It can respond perfectly to the energetic need of a good part of earth's population. However, the intermittency character is the biggest problem to face. In this context, a back-propagation feedforward neural network is presented in this paper, in order to forecast next 24 hours PV power production of the "îlot RIZOMM". The results show a good accuracy of the model built. In addition, the comparison of the ANN results and SteadySat's predictions shows that the performance of the model is quite acceptable.
As a perspective, an algorithm to calculate the solar radiation directly on the solar panels may be interesting to establish, in order to use this variable as one of the model's inputs instead of the global solar radiation. This can be a good way to improve the PV power prediction of the ANN model.

5 References

1. X. Yan, D. Abbes, and B. Francois, 'Solar radiation forecasting using artificial neural network for local power reserve', 2014 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM), pp. 1–6, Nov-2014.
2. I. Ghalehkhondabi, E. Ardjmand, G. Weckman, and W. Young, An overview of energy demand forecasting methods published in 2005–2015, vol. 8. 2016.
3. M. A. Mat Daut, M. Y. Hassan, H. Abdullah, H. A. Rahman, M. P. Abdullah, and F. Hussin, 'Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review', Renewable and Sustainable Energy Reviews, vol. 70, pp. 1108–1118, Apr. 2017.
4. N. Fumo and M. A. Rafe Biswas, 'Regression analysis for prediction of residential energy consumption', Renew. Sustain. Energy Rev., vol. 47, pp. 332–343, Jul.2015.
5. R. H. Inman, H. T. C. Pedro, and C. F. M. Coimbra, 'Solar forecasting methods for renewable energy integration', Prog. Energy Combust. Sci., vol. 39, no. 6, pp. 535–576, Dec.2013.
6. H. Zhao, S. Su, Z. Mi, and F. Wang, 'Short-Term Solar Irradiance Forecasting Model Based on Artificial Neural Network Using Statistical Feature Parameters', Energies, pp. 1355–1370, 01-May-2012.
7. J. Varanasi and M. M. Tripathi, 'Artificial neural network based wind power forecasting in belgium', in 2016 IEEE 7th Power India International Conference (PIICON), 2016, pp. 1–6.
8. 'SteadySat - solar production forecasting for next hours', SteadySun. [Online]. Available: https://steady-sun.com/technology/steadysat/. [Accessed: 21-May-2019].
9. T. Senjyu, H. Takara, K. Uezato, and T. Funabashi, 'One-hour-ahead load forecasting using neural network', IEEE Transactions on Power Systems, vol. 17, no. 1, pp. 113–118, Feb. 2002.
10. K.Gnana Sheela, S.N. Deep, 'Review on Methods to Fix Number of Hidden Neurons in Neural Networks', Mathematical Problems in Engineering. 2013.
11. B. Robyns and all, 'Electrical Energy Storage for Buildings in Smart Grids', ISTE 2019, ISBN: 9781848216129.
12. https://www.infoclimat.fr/

Likelihood Estimation for Hunter Syndrome using ZIP model and Simulated Data

Behrouz Ehsani-Moghaddam (PhD) [1]; John A. Queenan (PhD) [1]; Jennifer MacKenzie (MD) [2]; Richard V. Birtwhistle (MD, MSc) [1]

[1] Canadian Primary Care Sentinel Surveillance Network, Department of Family Medicine, Queen's University, 220 Bagot Street, P.O. Bag 8888, Kingston, Ontario, K7L 5E9, Canada.
[2] McMaster University, Department of Pediatrics, Division of Genetics, Room 3N11-G, 1280 Main Street West, Hamilton, Ontario, L8S 4K1, Canada.
Corresponding Author:
Behrouz Ehsani-Moghaddam:
Email: behrouz.ehsani@dfm.queensu.ca
ORCID iD: https://orcid.org/0000-0002-5038-1118

Abstract. Hunter syndrome is a rare disease, caused by a deficiency in the activity of the lysosomal enzyme, alpha-L-iduronidase. We created a dataset using Monte Carlo simulation technique containing 18 variables including 16 symptoms, 1 variable as patient age and 1 variable as disease status. A ZIP regression model was created to estimate the likelihood of having Hunter syndrome in individual patients using the simulated data. Out of 17 predictors, 14 variables were selected and remained in the final model. The result of ZIP model validation by data splitting revealed that the overall accuracy, sensitivity, specificity, positive predictive value and negative predictive value were 1.0, 0.983, 1.0, 0.892 and 1.0, respectively. The factor analysis detected 8 factors and 11 symptoms that accounted for nearly 60% of the total variance in the dataset. These findings suggest that these 11 symptoms should be considered as the most important features when evaluating new cases for diagnosis and the simulated dataset can be used successfully for future studies of the disease such as for predictive modeling.

Keywords: CPCSSN, disease prediction, disease diagnosis, Hunter syndrome, MPS II disease, Mucopolysaccharidosis, factor analysis, zero-inflated Poisson, ZIP model

adfa, p. 1, 2011.

1 Introduction

Hunter syndrome or Mucopolysaccharidosis type II (MPS II) is an X-linked inherited disorder, caused by a deficiency in the activity of the lysosomal enzyme, alpha-L-iduronidase [1]. Expression of MPS II is variable with symptoms emerging at different ages [2, 3, 4, 5]. The prevalence of the disease has been estimated about 0.6-1.3 in 100,000 male births [6]. The result of delay in diagnosis or incorrect diagnosis has the potential to be tragic e.g., clinical worsening of the patient's health in terms of physical, intellectual, psychological conditions and sometimes even death; incorrect treatment, inadequate support from family members or society, and loss of confidence in the healthcare system [5]. Thus, any method that helps health practitioners with diagnosis as early as possible could have a positive influence on the patient's life.

In clinical practice, prediction models can reduce the burden of a disease by assisting patients and their physicians with a diagnosis or a prognostic outcome or with the classification of a patient according to his/her risk assessment [7]. In previous study with a small size population, the Naïve Bayes classifier (NBC) as a simple machine learning algorithm was applied to identify a group of patients with the highest likelihood of having MPS II disease with relatively common features using a real dataset from the Canadian Primary Care Sentinel Surveillance Network (CPCSSN) [8]. The performance of the NBC with other Bayesian networks was also compared. The objectives of present study were: 1. creating a synthetic dataset with larger size using a simulation-based approach. Such dataset can be used for further studies of MPS II and for free exchange of information without any need for ethical approval. The freedom from the ethics restrictions imposed will somehow eliminate the discouragement to share data and thereby encourage knowledge generation through openness of research findings [9]; 2. To determine the primary pattern of predictors and their influences into the likelihood of having the disease; 3. To construct a predictive model using the simulated dataset. The selection of attributes that are most relevant in relation to MPS II disease forecasting and the minimum number of factors that are sufficiently accounted for the largest amount of the variance in the MPS II dataset were also investigated by LASSO selection technique [10] and factor analysis.

2 Methods

2.1 Study population

The seed data for this study was a subset of the Canadian Primary Care Sentinel Surveillance Network (CPCSSN) database from 2016. The subset data was consisted of the records of 506,497 male patients from networks in Calgary, Manitoba, Ontario and Newfoundland who had at least one visit to a primary care clinic in the past 24 months, which 125 of them were previously classified as individuals with the highest likelihood of having MPS II disease using a Naïve Bayes classifier with high accuracy (Table 1) [8].

The seed data were simulated by the Monte Carlo simulation technique with a sample size of 1000, and 10,000 repetitions and unrestricted random sampling for the regression model. The simulated data were used for further analyses including the estimation of regression coefficients, estimation of population parameters and statistical inference. Table 2 shows independent features' names and their associated symptoms for the MPS II disease.

2.2 Zero-inflated Poisson (ZIP) model

A multivariable regression model was created to investigate the association between all covariates of study and the presence or absence of MPS II disease. Since, there was an excessive number of outcomes with a count of 0, the regression model was made using a ZIP probability distribution.

The ZIP distribution combines the Poisson distribution and the logit distribution [11]. The idea is that in a ZIP model, the data have two parts. In the first part, the outcome is always a zero count, while in the other part the counts follow a standard Poisson process.

Suppose that,

$$\begin{cases} yi = 0 \ \ with\ probability \ \ \omega_i \\ yi \sim Poisson\ (\lambda_i)\ with\ probability\ (1-\omega_i) \end{cases} \quad (1)$$

where *yi* is the outcome variable (MPS II disease), λ_i is the expected Poisson count for each individual and ω_i is the probability of extra zeros.

A ZIP model for the MCP II dataset, which comprises two parts, can be written as:

$$\text{Pr (yi = j)} = \begin{cases} \omega_i + (1-\omega_i)\exp(-\lambda_i) & if\ \ j = 0 \\ (1-\omega_i)\frac{\mu_i^{y_i}\exp(-\lambda_i)}{y_i!} & if\ \ j > 0 \end{cases} \quad (2)$$

In a ZIP model, the Poisson distribution is assumed to have a variance that is equal to the distribution's mean. The mean and variance of outcome for the zero-inflated Poisson are given by:

$$E(yi) = \mu = (1 - \omega)\lambda \quad (3)$$

$$Var\,(yi) = \mu + \frac{\omega}{1-\omega}\mu^2 \quad (4)$$

2.3 Variable selection and model evaluation

Prior to variable selection, variance inflation factor (VIF) was measured for all variables to detect multicollinearity among them[12]. Covariate selection was carried out by LASSO (Least Absolute Shrinkage and Selection Operator) method using all predictors including patient's age for both the Poisson distribution and the logit distribution (zero-inflation part) of ZIP model. Selection for both distributions was based on the lowest Schwarz Bayesian Information Criterion (SBC) [13]. For the selected model, the statistical significance of each coefficient was tested using the Chi-square test. Pearson Chi-square statistic was also used for testing overdispersion of ZIP model. If ZIP regression model is correctly selected and there is no overdispersion, the Pearson chi-square statistic will have an expected value of 1. If the value is significantly different from 1, then we can conclude that overdispersion exists.

Additionally, three more regression models were created using a zero-inflated binomial (ZINB), a regular binomial, and a regular Poisson probability distribution. The ZIP and ZINB are popular models for data that exhibit excess zeros such as observations from rare diseases[11]. The final ZIP model was also compared with Poisson, binomial and ZINB models using unadjusted and adjusted Akaike and Schwarz values by Vuong and Clarke tests [14, 15]. These two tests function using likelihood ratio and Kullback-Leibler information criterion.

2.4 Model validation

The model performance was assessed by the Validation Set Approach technique [16, 17], which was carried out by partitioning randomly 50% of the simulated dataset as training set, 25% as validation set and 25% as test set. The

training set was used to create the ZIP model. Then the performance of the model confirmed using validation dataset. The accuracy of the final selected model was measured on test dataset by accuracy, Kappa and other associated statistics that are often calculated from a confusion matrix for a binary classifier [18].

2.5 Comparison between simulated and real data

To compare the observed data and simulated data distributions and to evaluate the model derived from simulated data against real observations, nonparametric tests for location and scale differences were carried out using the scores of response variable (MPS II disease) using Wilcoxon and median tests. We also calculated the empirical distribution function statistics of the Kolmogorov-Smirnov (KS) test. Using asymptotic p-values, the KS test quantifies the likelihood that whether two distributions from observed and the simulated data are different or not. The null hypothesis was that there is no difference in the MPS II response status against an alternative hypothesis that the response status differs in the two datasets.

2.6 Factor analysis

Factor analysis was conducted to identify underlying factors that explain the pattern of correlations within the set of independent variables and to determine the minimum number of factors that are adequately accounted for the largest amount of the variance in the MPS II dataset. A varimax rotation was applied to obtain orthogonal factor scores and minimum eigenvalue for extraction set at 1. Factors loadings which met a minimum of 0.60 as suggested by Guadagnoli and Velicer [19] and Field [20] were considered significant and are reported.

3 Statistical software

Data extraction from SQL server, data mining including data cleansing procedure, LASSO selection, data simulation and factor analysis were carried out by SAS software (version 9.4 TS). Variance inflation analysis was performed by package usdm of R program.

4 Results

Variance inflation factors and Phi coefficient correlation analysis using all features in the model did not show any substantial correlation among features, indicating that the collinearity among predictors in the model was very small. Furthermore, the result of Pearson Chi-square test for overdispersion of ZIP model was calculated 0.0012. Considering the computed p-value of 1.0 for overdispersion test, we would fail to reject the null hypothesis of no overdispersion at the most generally used confidence levels. The zip model for MPS II disease contained two parts, a Poisson count distribution and the logistic model for predicting excess zeros. Out of 17 predictors, 14 variables were selected and remained in the final ZIP model. Table 2 shows the predictors that entered in the ZIP model and those remained in either Poisson or logit part of model (in bold). The Poisson part of final ZIP model contained 13 independent variables, i.e., all predictors except Age, Macrocephaly, Seizure and Joint and the zero-inflated part contained all predictors except Macrocephaly and Stature.

The likelihood of having MPS II disease for any individual patient can be estimated by his symptoms, the estimated parameter and the model 2. For example, the probability of having MPS II disease for a patient who is younger than 21 years old and has sleep apnea, COPD, spinal injury, hernia, respiratory infection, and carpel tunnel in his records can be estimated in four steps as follows:

Poisson part: A = -1.50911 + 0.169868 (Apnea) + 0.18549 (COPD) + 0.285727 (Spinal injury) + 0.307539 (Hernia) + 0.213357 (Respiratory) + 0.282409 (Carpal) = -0.06472 = -0.06472

Logistic part: B = 48.333826 -7.940066 (Age) - 8.754209 (Apnea) - 9.012525 (COPD) -11.966356 (Spinal injury) -12.67133 (Hernia) -6.606455 (Respiratory) -12.861689 (Carpal)

= -21.478804

Pzero = exp (B) / (1+ exp (B)) = 4.69758E-10

PCount = exp (A) × (1- P zero) = 0.94

Where A and B are the linear predictions based on the Poisson and the zero-inflated model, respectively. Pzero and PCount are the probabilities of having zero and count for MPS II disease, respectively. Thus, the probability of having

MPS II disease for a person younger than 21 who has those symptoms will be 0.94.

Table 3 reveals the results of comparisons between ZIP and three other models, i.e. ZINB, a regular Poisson and a binomial using the Vuong and Clarke tests. According to Table 3, in general, the preferred model over other models is a zero-inflated Poisson model. The positive values of the Z statistics from these tests also indicate that the ZIP model is a model, which is the closest to the correct model.

The result of ZIP model validation has been shown in Table 4. The overall accuracy, sensitivity, specificity, PPV (positive predictive value) and NPV (negative predictive value) were estimated 1.0, 0.983, 1.0, 0.892 and 1.0, respectively. Out of 2501100 observations, 77 individuals were misclassified in the dataset (0.003%): 11 patients without MPS II were predicted to incorrectly having the disease (type-I error) and 66 individuals having MPS II were incorrectly predicted without the disease (type-II error). Cohen's Kappa, which is a measure of the classifier performance as compared to chance, was estimated 0.934 (95% CI: 0.920-0.950).

To compare the true and simulated data distributions, nonparametric tests were carried out using the Wilcoxon, median and KS tests. The Wilcoxon and median tests showed that, the response for the simulated data is not significantly different than for the observed data. The asymptotic and exact p-values for the Kolmogorov-Smirnov test were 1.0 and 0.9265, respectively. This indicates that the distributions are identical for the two datasets.

Plot 1 includes Scree plot and the Variance Explained plots from the Factor analysis of MPS II disease symptoms. The Scree Plot illustrates that the eigenvalue (the variances of the factors) of the first component is approximately 1.40 and the eigenvalue of the third component is largely decreased to about 1.0. According to Factor analysis and the Scree plot these eight distinct factor patterns, which exist among patients identified with MPS II using the ZIP model. The 1st factor is characterized by high loadings of Otitis and Respiratory with a factor loading of 0.72 and 0.65, respectively. The 2nd factor was a measure of Cardiac and COPD, with a factor loading of 0.67 and 0.61, respectively. The 3rd factor was a measure of Carpel (0.71) and Apnea (0.65), which is explained by high loadings of these features. The 4th factor was a measure of only Hernia (0.69). The 5th factor was a measure of Skin with a factor loading of 0.91. The 6th, 7th and 8th factors were characterized by high loadings of Joint, Seizure and Hepatosplenomegaly each with a high loading of 0.97.

5 Discussion

In this study, we created a simulated data for Hunter syndrome containing 17 features. Because of very low prevalence of MPS II disease, such dataset cannot be created by retrospective or prospective cohort studies. The ability to produce and share synthetic datasets can shorten the idea-to-insight time from years to hours. It can also reduce research expenses and can diminish legal and ethical barriers to data sharing [21]. Moreover, when we need to determine the predictive performance of a statistical model and to estimate the risk of disease or for studies such as therapeutic drug monitoring, when only a small number of observations is available, simulated dataset can be used for validation [22]. The predictive models such as ZIP model could be used as a preselection technique to support screening persons at risk before requesting an expensive or limited diagnostic test such as Iduronate 2-sulfatase (I2S) enzyme activity, which is the gold standard test for MPS II diagnosis [23]. Using an exploratory Factor analysis, we found out that Otitis and Respiratory was the first factor extracted in the analysis and accounted for the greatest amount of variance in the data. This finding suggests that future studies should consider these 8 factors and their 11 symptoms as the most important variables of MPS II when evaluating new cases for diagnosis.

There are several potential limitations. There is a risk of bias by using synthetic data, for example, if the original seed data were biased, the simulated data would be biased too. The predictive capability of the ZIP model also needs further external investigation using a gold standard validation such as I2S enzyme activity in patients. Another limitation of this predictive model is that, like other classifiers, positive predictive value of a regression model for disease prediction depends on available features in the model, the more symptoms, better prediction. As a result, in a real world, those patients with MPS II disease, who have not developed many symptoms or our EMR system has not captured their symptoms for seed data, may not be detectable by the model.

6 Conclusion

The model performance evaluation revealed that, the model fits the MPS II data appropriately. In this study, despite some novel findings, there are at least two potential limitations that should be taken into account.

7 Acknowledgements

This project was supported in part by a grant from Shire Canada. Authors declare that no competing interests exist and Shire Canada has played no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript. Dr. MacKenzie has received grant funding, honoraria and travel support from Shire Canada.

8 Conflict of Interest

There are no conflicts of interest to report.

9 Informed Consent

Informed consent was obtained from Queen's University.

10 References

1. Bach G, Eisenberg FJ, Cantz M, Neufeld EF. The defect in the Hunter syndrome: deficiency of sulfoiduronate sulfatase. Proc Natl Acad Sci USA. 1973; 70: 2134-2138.
2. Schwartz I, Ribeiro MG, Mota JG, Toralles MBP, Correia P, Horovitz D. A clinical study of 77 patients with mucopolysaccharidosis type II. Acta Pædactrica. 2007; 96: 63–70.
3. Martin R, Beck M, Eng C, Giugliani R, Harmatz P, Muñoz V. Recognition and diagnosis of mucopolysaccharidosis II (Hunter Syndrome). Pediatrics. 2008; 121: 377–386.
4. Needham M, Packman W, Quinn N, Rappoport M, Aoki C, Bostrom A, Cordova M, Macias S, Morgan C, Packman S. Health-Related Quality of Life in Patients with MPS II. J Genet Counsel. 2015; 24: 635–644.
5. Europian Organization for Rare Diseases. Rare Diseases: Understanding This Public Health Priority; 2005. Available from: https://www.eurordis.org/IMG/pdf/princeps_document-EN.pdf.
6. A Guide to Understanding Mucopolysaccharidosis (MPS) II. The Canadian Society for Mucopolysaccharide and Related Diseases Inc.; 2017. Available from: https://www.mpssociety.ca/wp-content/uploads/2017/04/MPSIIBookletEnglish.pdf
7. Steyerberg EW. Clinical Prediction Models. Springer Science + Business Media, LLC. 2009; DOI:10.1007/978-0-387-77244-8_2.
8. Ehsani-Moghaddam B, Queenan JA, MacKenzie J, Birtwhistle RV. Mucopolysaccharidosis type II detection by Naïve Bayes Classifier: an example of patient classification for a rare disease using electronic medical records from the Canadian Primary Care Sentinel Surveillance Network. 2018; PLOS ONE https://doi.org/10.1371/journal.pone.0209018.

9. Robinson S, F FitzGibbon, J Eatock, T Hunniford, D Dixon & B J Meenan. Application of synthetic patient data in the assessment of rapid rule-out protocols using Point-of-Care testing during chest pain diagnosis in a UK emergency department. *Journal of Simulation* 2009; 3:3, 163-170, DOI: 10.1057/jos.2009.4.

10. Tibshirani, R. Regression Shrinkage and Selection via the lasso. Journal of the Royal Statistical Society. Series B (methodological). 1996; 58: 267–88.

11. Ridout M, Demétrio CGB, Hinde JP. Models for Count Data with Many Zeros. In Proceedings of the 19th International Biometric Conference. Cape Town. 1998; 179–192.

12. Yoo W, Mayberry R, Bae K, He Q, Lillard J. A study of effects of multicollinearity in the multivariable analysis. Int J Appl Sci Technol. 2014; 4: 9–19.

13. Schwarz G. Estimating the dimension of a model. Annals of Statistics. 1978; 6 (2): 461–464.

14. Clarke K. A Simple Distribution-Free Test for Non-Nested Model Selection. Political Analysis. 2007; 15: 347-363.

15. Vuong QH. Likelihood Ratio Tests for Model Selection and Non-Nested Hypotheses. Econometrica. 1989; 57(2): 307-333.

16. James G, Witten D, Hastie T, Tibshirani R. An Introduction to Statistical Learning: with Applications in R Springer. 2013; Chapter 4: 175-194.

17. Picard RR, Berk KN. Data splitting. The American Statistician. 1990; 44: 140-147.

18. Florkowski CM. Sensitivity, Specificity, Receiver-Operating Characteristic (ROC) Curves and Likelihood Ratios: Communicating the Performance of Diagnostic Tests. Clin Biochem Rev. 2008; 29: S83–S87.

19. Guadagnoli E and Velicer WF. Relation to sample size to the stability of component patterns. Psychological Bulletin 1988; 103(2): 265-275. doi: 10.1037/0033-2909.103.2.265.

20. Field A. Discovering Statistics Using SPSS. 2005. 2nd edn, SAGE, London.

21. Foraker R, Mann DL, Payne PRO. Are synthetic data derivatives the future of translational medicine? *JACC Basic Transl Sci* 2018; 3(5): 716–718.

22. van der Meerm AF and Neef C. Optimal sampling times for therapeutic drug monitoring. *Adv Pharmacoepidem Drug Safety* 2012; DOI: 10.4172/2167-1052.1000S1-004.

23. Johnson BA, van Diggelen OP, Dajnoki A, Bodamer OA. Diagnosing lysosomal storage disorders: mucopolysaccharidosis type II. Curr Protoc Hum Genet. 2013; 79:17.14.

Table 1. Performance evaluation of the seed dataset from the Naïve Bayes classifier.

	Actual		
Predicted	**No**	**Yes**	**Row Total**
No	357251	7815	365066
Yes	0	18997	18997
Column Total	357251	26992	384063
Accuracy	0.99		
Kappa*	0.91		
Sensitivity*	0.84		
Specificity*	1.0		

Table 2. Independent variables from patients of MPS II disease. The remained features in the final ZIP model are shown in bold.

Feature name	Symptom/Description
Stature	Short stature, contracture, coarse facial features, congenital Musculo-skeletal
Joint	Joint pain, joint stiffness
Apnea	Sleep apnea
COPD	COPD, airway obstruction
Hearing	Progressive hearing loss
Spinal injury	Spinal cord injury, spinal stenosis, compression, dysostosis, congenital musculoskeletal
Hernia	Umbilical hernia, inguinal hernia
Otitis	Chronic ear infections, AOM, otitis
Respiratory	Respiratory infection
Carpal	Carpel tunnel syndrome
Cardiac	Cardiac disease, heart valve problem, cardiac problem, ventricular hypertrophy
Hepatosplenomegaly	Hepatosplenomegaly, hepatomegaly, enlarged liver, splenomegaly, enlarged spleen
Skin	Pebbly skin lesion, thickened skin
Seizure	Seizure
Diarrhea	Diarrhea
Macrocephaly	Macrocephaly, enlarged head
Age	Patient's age: (1 for younger than 21or 0 for otherwise)

Table 3. Comparisons between ZIP and ZINB; ZIP and regular Poisson; and ZIP and binomial model for MPS II data using Vuong and Clarke tests. H_0: models are equally close to the true model; H_a: one of the models is closer to the true model

Test	Distribution Comparison	Vuong and Clarke Statistic	Z	Pr>|Z|	Preferred model
Vuong	ZIP and ZINB	Unadjusted	12213.57	<.0001	ZIP
	ZIP and ZINB	Akaike Adjusted	12213.55	<.0001	ZIP
	ZIP and ZINB	Schwarz Adjusted	12213.41	<.0001	ZIP
	ZIP and regular Poisson	Unadjusted	20.7538	<.0001	ZIP
	ZIP and regular Poisson	Akaike Adjusted	20.7375	<.0001	ZIP
	ZIP and regular Poisson	Schwarz Adjusted	20.622	<.0001	ZIP
	ZIP and binomial	Unadjusted	12213.57	<.0001	ZIP
	ZIP and binomial	Akaike Adjusted	12213.55	<.0001	ZIP
	ZIP and binomial	Schwarz Adjusted	12213.41	<.0001	ZIP
Clarke	ZIP and ZINB	Unadjusted	253213.5	<.0001	ZIP
	ZIP and ZINB	Akaike Adjusted	253213.5	<.0001	ZIP
	ZIP and ZINB	Schwarz Adjusted	253213.5	<.0001	ZIP
	ZIP and regular Poisson	Unadjusted	4996952	<.0001	ZIP
	ZIP and regular Poisson	Akaike Adjusted	4996952	<.0001	ZIP
	ZIP and regular Poisson	Schwarz Adjusted	-469219	<.0001	Poisson
	ZIP and binomial	Unadjusted	253213.5	<.0001	ZIP
	ZIP and binomial	Akaike Adjusted	253213.5	<.0001	ZIP
	ZIP and binomial	Schwarz Adjusted	253213.5	<.0001	ZIP

Table 4. Validation of the ZIP model for MPS II patients using test dataset.

	Predicted		
Actual	**No**	**Yes**	**Total**
No	2500476	11	2500487
Yes	66	547	613
Total	2500542	558	2501100
Accuracy	1.0 (95% CL: 1.0-1.0)		
Sensitivity	0.980 (95% CL: 0.965-0.990)		
Specificity	1.0 (95% CL: 1.0-1.0)		
PPV	0.892 (95% CL: 0.866-0.916)		
NPV	1.0 (95% CL: 1.0-1.0)		
Kappa	0.934 (95% CL: 0.920-0.950)		

PPV=Positive predictive value

NPV=Negative predictive value

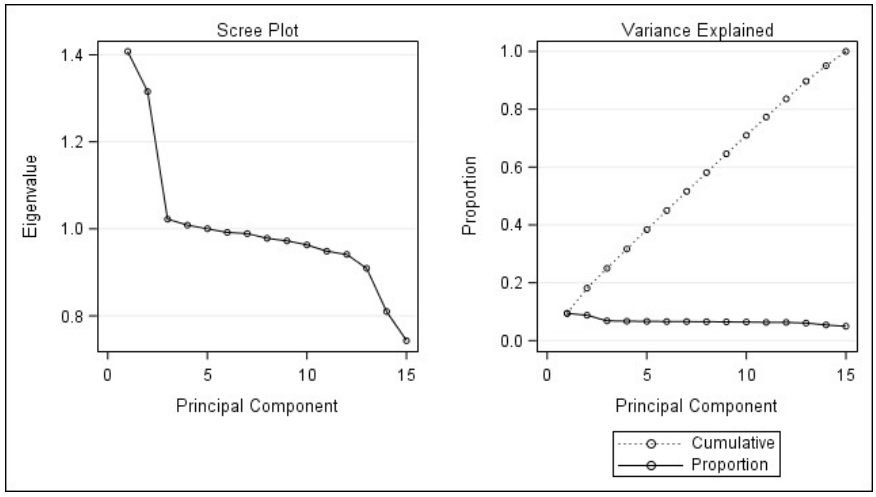


Fig. 1. Scree plot and the Variance Explained plot from the Principle Component analysis of MPS II disease symptoms. The Scree plot illustrates that the eigenvalue of the first component is approximately 1.4 and the eigenvalue of the third component is largely decreased to about 1.0. The Variance Explained plot shows that nearly 60% of total variance is explained by the first eight principal components.

Double Seasonal Holt-Winters to forecast electricity consumption in a hot-dip galvanizing process

J.Carlos García-Díaz[1] and O.Trull[2][0000-0003-2896-8606]

[1,2] Dept. of Applied Statistics, Operational Research and Quality
Universitat Politècnica de València
juagardi@eio.upv.es

Abstract. The hot-dip galvanizing process is a very common process in the automotive industry and enables steel bands to be protected against corrosion by coating with zinc. The key process in the galvanizing is the zinc bath, and especially the heating inductors, which have a high electricity consumption. The recent price rises in Spain have led industries to take more care about the electricity they consume, as well as future demand in order to manage their energy costs efficiently.
Time series forecasting is a powerful tool for forecasting future demand. This paper focuses on the modelling and exploitation of a multiple seasonality Holt–Winters model to perform these actions. An application of this method to a real example utilising data from an industrial factory in Spain is explained and discussed.

Keywords: forecasting, electricity, galvanizing, hot-dip.

1 Introduction

The energy consumption in production systems is a key factor in their economically efficient management. The quantity of energy consumed depends on the mode of production and on the instant of the process in which it is needed. As a result of these dependencies, it is necessary to control the installed capacity and the cost of the energy utilised at every stage. A recent study on the electricity market [1] determined that in Spain the price of electricity for industrial purposes has increased by 9%. This has forced industries to project future energy consumption in order to control spending. Hot-dip galvanizing is a widespread technique where steel is protected against corrosion by a zinc coating. This process requires control over the bath temperature provided by electrical heat inductors, which consume a large amount of electric power. To efficiently manage the energy consumption cost, as discussed above, it is desirable to provide a forecast of the electricity

demand required for the bath. In order to be able to produce accurate forecasts, time series techniques are commonly used, as they allow the analysis of previous data to obtain future forecasts. This paper focuses on the method of forecasting the electric consumption in a continuous hot-dip galvanizing process at a real factory located in Spain, and considers how to optimise the economic cost of this activity. The paper is organised as follows: first, the hot-dip galvanizing process is explained, focusing on the heat inductors, which are the main electrical consumers. Second, the modelling of the consumption using time series is explained. Finally, conclusions are presented in the last section.

2 Hot-Dip galvanizing process

Galvanized steel is a product that combines a good resistance to corrosion with good mechanical properties for the manufacturing in the automotive industry. The steel strips are coated with a layer of zinc that forms an alloy with the steel giving the desired properties. The coating through the hot-dip galvanizing process produces a very high added value to the steel, in a continuous and quality process [2]. Figure 1 shows an overview of the galvanizing process. In the annealing furnace, the steel strip is heated, which subsequently enters the bath of molten zinc at 460 ° C. The thickness of the coating is controlled by the air-wiping jets. Finally, the galvanized steel strip is cooled and goes through the skin-pass process [3-5].

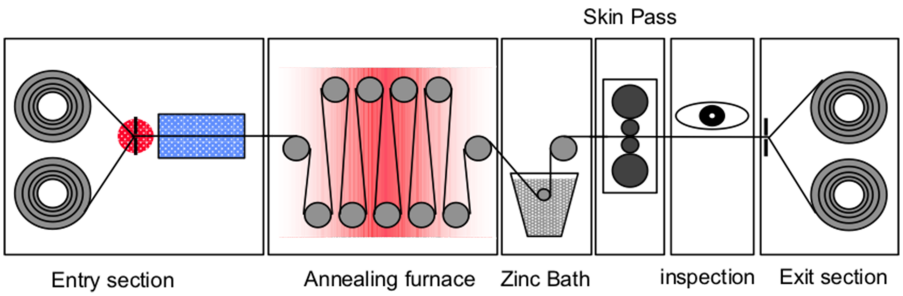


Fig. 1. Hot-dip galvanizing line.

2.1 Temperature control of the zinc bath

The supply of zinc to the process is produced by the addition of zinc-aluminium (Zn-Al) ingots that maintain the necessary zinc content in the bath. The temperature of the bath modifies with the supply of each zinc ingot. These temperature changes must be compensated by heating to maintain the bath at approximately 460°C [6]. The control of this temperature is very important to obtain a high-quality product, which is monitored with the help of three thermocouples (T1, T2, and T3), as can be seen in Figure 2. The presence of the TP12 thermocouple allows the heating temperature of the steel strip to be controlled in turn.

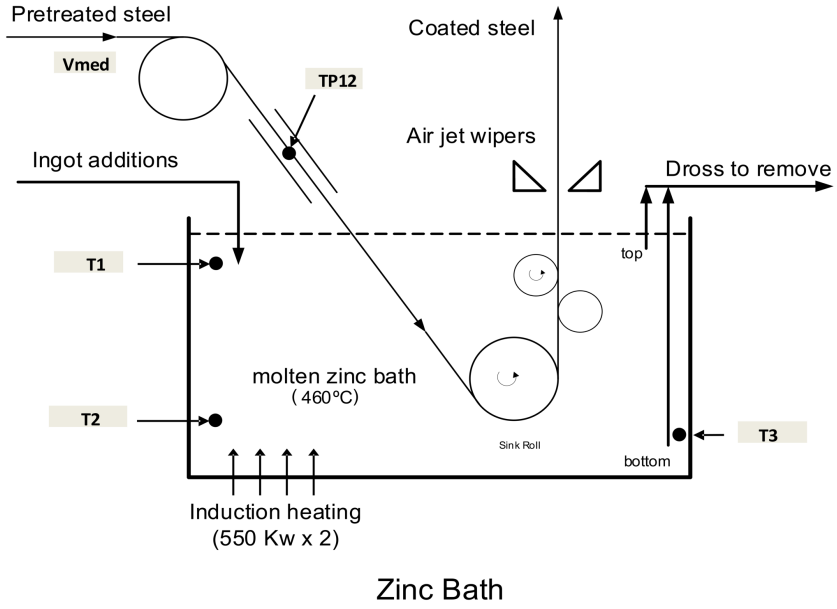


Fig. 2. Galvannealing section (zinc pot). Locations of thermocouples T1, T2, and T3 for monitoring the temperature of the bath, and the induction heating system (adapted from [2]).

2.2 Induction heating system

The heating of the bath is carried out by induction. Induction uses electromagnetic energy to heat liquid metals such as molten zinc. Induction heating is a process of heat transfer by electromagnetic induction. The temperature profile of the liquid to be heated (molten zinc) and the energy consumption are functions of the current density, the

frequency, the properties of the material, the design of the coils, the coupling between the coils and the liquid to be heated, and the characteristics of the power supply. A scheme of the induction heating system is shown in Figure 3.

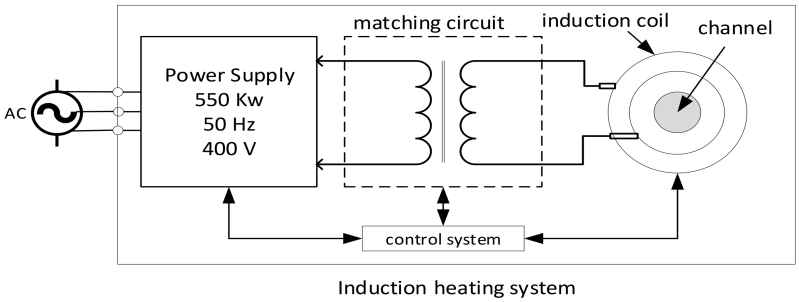


Fig. 3. Design of induction heating system.

The function of the inductors is to maintain the temperature of the zinc bath at 460 °C by induction, creating a circulation of liquid metal. In the hot galvanizing line of this study, the coating pot had a zinc capacity of 300 T, was rated at 1100 kW and equipped with two 550 kW Jet Flow inductors [7]. The two Jet Flow inductors are located on either side of the bath to supply the necessary heat flow to maintain the bath temperature, considering the heat losses from the surface and the fusion of the zinc ingot (see Figure 2). Through the use of induction heating the formation of dross is minimized, and a great homogeneity of temperatures is obtained, both in the channels and in the zinc bath. Inductors are considered as operating at higher power during the ingot melting period to compensate for the heat lost to the ingot [8]. The inductors are formed by a case of welded heavy steel construction, a magnetic circuit with a core in steel plates and two primary copper coils of high conduction cooled by air blowers, each with integral driving motors [9] (see Figure 4, a).

2.3 Channel induction furnaces: zinc pot bath-induction heating system

Heating process. The heating takes place in a small, narrow cavity called a channel. This cavity is located inside the inductor and flows through a steel core inside which the primary coils are located. The operation is similar to that of a transformer, in which the primary is formed by the electric coils and the secondary is formed by the molten

zinc. The primary induction coils are wound around an iron core and are physically separated by said steel core. In the channel acting as the secondary coil, an increase in temperature is produced, melting the zinc alloy, and the molten metal flows through the channel. The inductor is in the bottom of the zinc pot, attached to it. To maintain these temperatures, the zinc pot contains a wall of refractory material around the entire pot and inductor [9].

Bath agitation mechanism. The Foucault currents generate a field of ascending and descending forces that produce stirring of the molten zinc. The result is a rapid and uniform heating of the molten zinc bath[9] (see Figure 4, b).

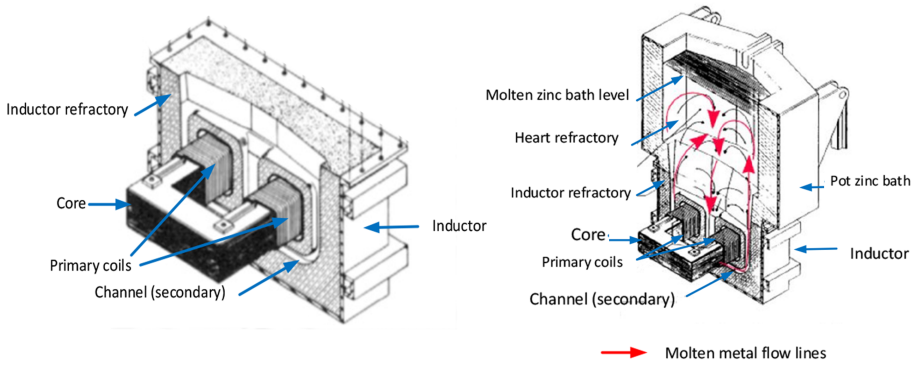


Fig. 4. Zinc pot – induction heating system (adapted from [9]). (a) Jet-Flow inductor. (b) Zinc pot bath – induction heating system.

3 Modelling and forecast of the electric consumption

The forecasting modelling was performed utilizing the data from an industrial factory in Spain (for confidentiality reasons, neither the name of the company nor its location is given here). The analysis of the electric consumption in the process holds for the hot-dipping of zinc, since it represents the greater part of the electric consumption. Measurements are made every minute, based on the temperature of the pot. Although the electricity consumption of the heaters is discrete, the inertia in the measurement produces a continuous series that will be used in the work. The series is shown in Figure 5. Time series forecasting makes it possible to analyse and treat the series described in the figure, and also provides forecasts for future consumption [10-12]. There

are many available techniques, such as exponential smoothing methods, including Holt–Winters [13-16] and state spaces [17,18], ARIMA [19], GARCH [20], etc. The first of these has been demonstrated to be a simple method to work with, while providing a high level of precision in the forecasts. Thus, it is chosen for this task. On the other hand, the strong influence of seasonalities suggests the utilization of a multiple seasonality pattern of exponential smoothing, that is, the multiple seasonality Holt–Winters [21,22].

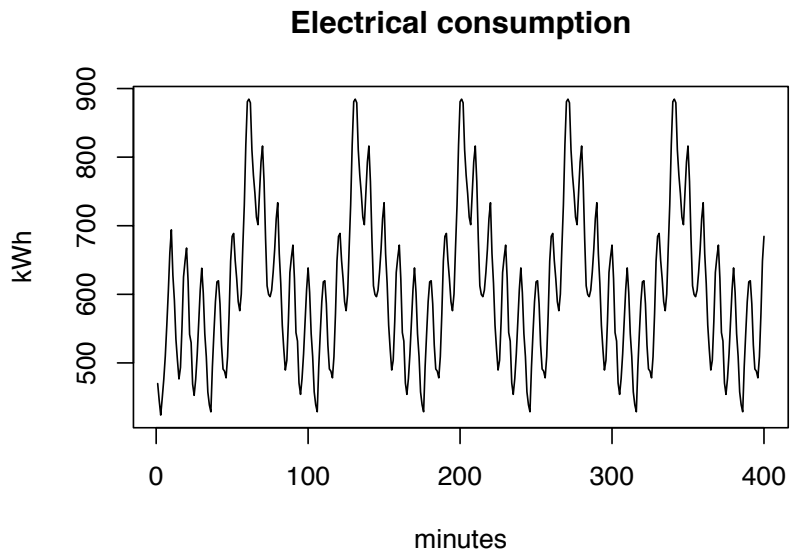


Fig. 5. Electrical consumption during dipping process.

Figure 5 evidences a typical series, which is strongly influenced by a seasonal pattern that repeats itself over the series, called the seasonality. Apparently, there are two seasonalities, both superimposed. In order to determine the length of the seasonal periods, a spectral analysis on the series is performed. This tool can detect the presence of seasonalities as well as measuring their frequencies, from which the period length is calculated. Figure 6 shows the spectral analysis for this series, and it is possible to observe the presence of two seasonalities, one that occurs every 10 minutes and another one superimposed that occurs every 70 minutes.

The multi-seasonality Holt–Winters models (nHWT), described in [23], are constituted by a series of smoothings (level, trend and seasonality) with smoothing parameters that make it possible to give the new data bigger or smaller weights compared to those observed earlier. A forecast equation gathers in this information and provides the forecasts for the following *k*-ahead time moments. These equations can relate to each other in additive or multiplicative ways, so we use a nomenclature of 3 letters to describe them. Table 1 shows the different nHWT methods. A common model is explained in (1) to (4). This model has additive trend and multiplicative seasonality methods, as well as being adjusted with AR (1), and is named as AMC.

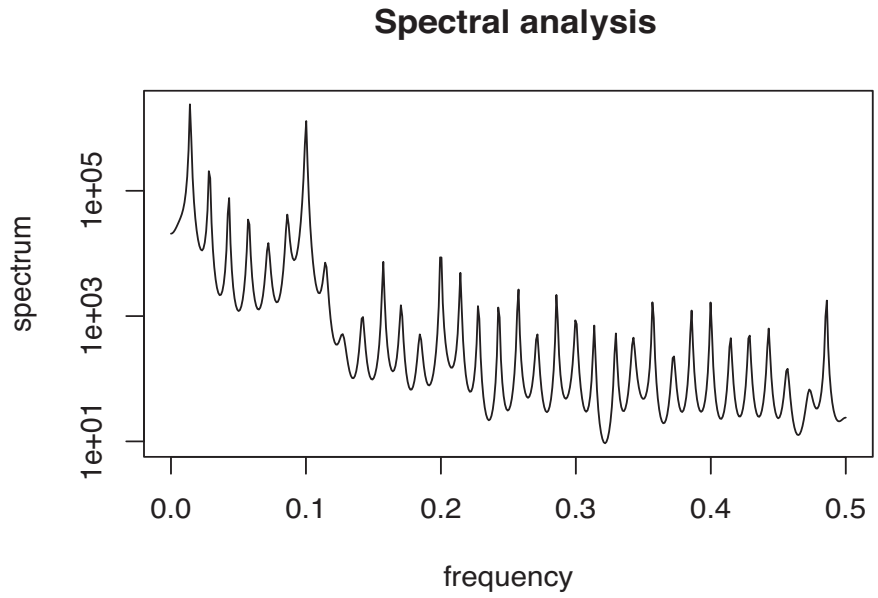


Fig. 6. Spectral analysis of the series to determine the seasonalities. A pattern with a period of 10 minutes and another with a period of 70 minutes can be clearly seen.

$$L_t = \frac{\alpha X_t}{\prod_{i=1}^{n_s} I_{t-s_i}^{(i)}} + (1-\alpha)(L_{t-1} + T_{t-1}) \quad (1)$$

$$T_t = \gamma(L_t - L_{t-1}) + (1-\gamma)T_{t-1} \quad (2)$$

$$I_t^{(i)} = \delta^{(i)} \frac{X_t}{L_t \prod_{j=1, j\neq i}^{n_s} I_{t-s_j}^{(j)}} + \left(1-\delta^{(i)}\right) I_{t-s_i}^{(i)}, \; i = 1, \ldots, n_s \quad (3)$$

$$\hat{X}_{t+k} = (L_t + kT_t) \prod_{i=1}^{n_s} I_{t-s_i+k}^{(i)} + \varphi_{AR}^{k} \varepsilon_t \quad (4)$$

The observed values are represented in X_t, whereas the forecasted are X_{t+k}. The smoothing equations L_t, T_t and $I_t^{(i)}$ for the level, trend and seasonalities have the smoothing parameters α, γ and $\delta^{(i)}$. i represents each seasonal pattern considered. The parameter φ_{AR} is the adjustment factor for the first autocorrelation error (ε_t). The values of s_i indicate the length of the seasonal patterns. In this case, $s_1 = 10$ and $s_2 = 70$. Before exploiting the model, it is necessary to determine the values of the smoothing parameters. An adjustment error indicator between the model and the observed data is used, and, through a minimisation algorithm, the parameters are obtained in the range [0,1]. The common indicators are the root of the mean squared error (RMSE) and the mean average percentage error (MAPE). Both indicators are shown in (5) and (6), where *n* is the number of observed values.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(\hat{X}_t - X_t)^2}{n}} \tag{5}$$

$$\text{MAPE} = 100\frac{1}{n}\sum_{n}\frac{|\hat{X}_t - X_t|}{|X_t|} \tag{6}$$

Table 1. Summary of the models. Notation: First letter defines the trend method (N: No trend, A: Additive, d: Damped additive, M: Multiplicative, D: Damped multiplicative); the second letter stands for the seasonal method (N: None, A: Additive, M: Multiplicative); the last one explains whether it has been AR(1) adjusted or not (blank: non adjusted, 1: adjusted with AR(1)). (e.g.: AM110,70 stands for a double (10 and 70 minutes) seasonal with additive trend and multiplicative seasonality, adjusted with the AR(1) adjustment.

Seasonality • Trend	None	Add.	Mult.	None	Add.	Mult.
	Not adjusted			AR(1) adjustment		
No trend	NN	NA	NM	NN1	NA1	NM1
Additive	AN	AA	AM	AN1	AA1	AM1
Damp. Add.	dN	dA	dM	dN1	dA1	dM1
Multiplic.	MN	MA	DM	MN1	MA1	MM1
Damp. Mult.	DN	DM	DM	DM1	DA1	DM1

We tried all the methods in Table 1, and conducted a comparison among them, with the first 330 values to fit the model (in-sample fitting), and using 70 as validation data (out-of-sample validation). The model with the lowest MAPE was used to provide forecasts. The results of this comparison are summarized in Table 2. Four models

clearly outperform the rest (models with no AR(1) adjustment are not shown, since the results were always worse than these). Finally, model $NM1_{10,70}$ is selected. The result is according to the series, with no trend. This allows the consumer to have a clear vision of future energy consumption for possible negotiation with energy suppliers, reaching agreements as described in [24]. In Figure 7 we show a sample of the forecasts. The black and blue line are the real observed data and the fit of the model, whereas the red line is the new forecast for the future. It is perceived that the predicted series follows the same patterns as the original series formed by the previously observed data, and overlaps the real data.

Table 2. Best results obtained from the fitting competition to select the model. All models include AR(1) adjustment. Both RMSE and MAPE are used for this purpose.

	MAPE of fit		MAPE on validation	
	Seasonal method		Seasonal method	
Trend method.	Additive	Multiplicative	Additive	Multiplicative
None	0.4418	0.3895	0.1767	**0.0004**
Additive	0.2480	0.1994	0.1256	0.2343
Dam. Add.	0.2542	0.5104	0.0537	**0.0004**
Mult.	0.5056	0.5199	0.0930	0.3213
Dam. Mult.	0.432	0.6449	0.0717	0.0083

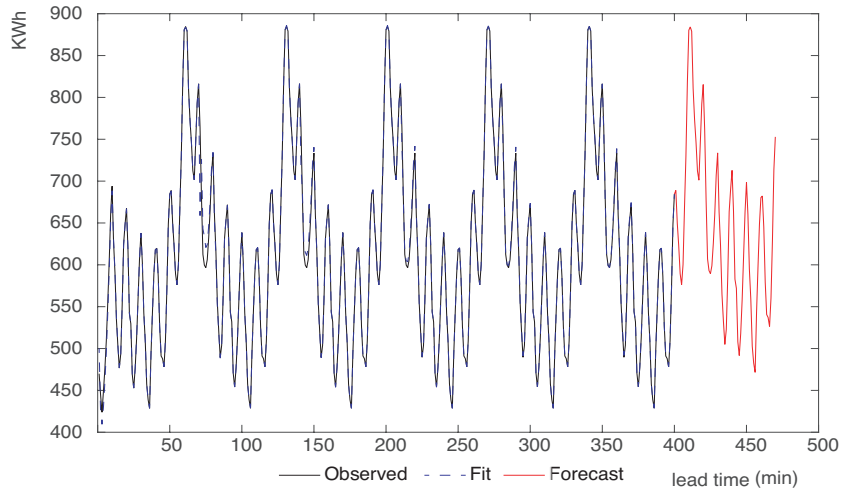


Fig. 7. Forecast of the electricity consumption.

Finally, we compared the results obtained using nHWT methods against others. Taylor et al. [25] stated that nHWT methods would outperform other methodologies for such kind of time series. Thus, we implemented the same procedure on the software R [26], using the functions for double seasonal Holt-Winters (dshw), ARIMA and state space models [27], including BATS (Box-Cox transformation and ARMA errors, Trend a Seasonality) and TBATS (Trigonometric transformation BATS). Results are summarized in Table 3.

Table 3. Accuracy comparison (MAPE) among the studied methods

Method	dshw(R)	BATS	TBATS	ARIMA	nHWT
Fit	0.314	0.887	1.659	3.225	0.2524
Forecasts	0.332	1.679	1.782	8.867	0.0004

Clearly nHWT methods outperformed the others, and provided more accurate forecasts.

4 Conclusions

In this paper we have described the process of a galvanized hot bath, and in particular focused on the study of the inductors' heat and electricity consumption. The objective was to understand the behaviour of electricity consumption, and to determine a model based on time series that can be exploited in future planning. The series shows a marked seasonal character, which led to the use of the Holt–Winters multiple seasonality model (nHWT). Thus, all the variants of these models were tested, verifying that the error committed is minimal, and determining the $NM1_{10,70}$ model as the most suitable one to use in making consumption forecasts. We made a comparison among several methods commonly used to forecast this kind of time series, and we checked nHWT method outperformed the rest.

This methodology allows more precise forecasting of electricity demand in the industry, based on which price agreements for the future can be negotiated. In this very early stage we used a selected time series. Future developments will also consider irregular time series.

5 References

1. Robinson D. Análisis comparativo de los precios de la electricidad en la Unión Europea y en Estados Unidos: una perspectiva Española [Comparative analysis of electricity prices in the European Union and the United States: a Spanish perspective]. Eurocofin, 2015.
2. Debón A, Garcia-Díaz JC. Fault diagnosis and comparing risk for the steel coil manufacturing process using statistical models for binary data. Reliab Eng Syst Saf 2012; 100: 102–114.
3. García-Díaz JC. Fault detection and diagnosis in monitoring a hot dip galvanizing line using multivariate statistical process control. Saf Reliab risk Anal theory, methods Appl 2009; 1: 201–204.
4. Ajersch F, Ilinca F, Hétu J-F. Simulation of flow in a continuous galvanizing bath: Part II. Transient aluminum distribution resulting from ingot addition. Metall Mater Trans B 2004; 35: 171–178.
5. Tang N-Y. Characteristics of continuous-galvanizing baths. Metall Mater Trans B 1999; 30: 144–148.
6. Garcia-Diaz JC, Debón A. Fault diagnosis in the steel coil manufacturing process. In: Ale BJM, Papazoglou IA, Zio E (eds) Reliability, Risk and Safety: Back to the Future. CRC Press/Balkema Taylor & Francis Group, 2010, pp. 93–100.
7. Ajax TOCCO Magnethermic Corporation.
8. Ajersch F, Ilinca F. Review of Modelling and Simulation of Galvanizing Operations. steel Res Int 2018; 89: 1700074.
9. Tama M. Development of channel induction furnaces for melting copper and brass. JOM 1974; 26: 18–25.
10. Weron R. Modeling and forecasting electricity loads and prices: A statistical approach. John Wiley & Sons, 2007.
11. Nowotarski J, Weron R. Recent advances in electricity price forecasting: A review of probabilistic forecasting. Renew Sustain Energy Rev. Epub ahead of print 2016. DOI: 10.1016/j.rser.2017.05.234.
12. Weron R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. Int J Forecast 2014; 30: 1030–1081.
13. Winters PR. Forecasting sales by exponentially weighted moving averages. Management 1960; 6: 324–342.
14. Taylor JW. Short-term electricity demand forecasting using double seasonal exponential smoothing. J Oper Res Soc 2003; 54: 799–805.
15. Taylor JW. Exponential smoothing with a damped multiplicative trend. Int J Forecast 2003; 19: 715–725.
16. Taylor JW. Triple seasonal methods for short-term electricity demand forecasting. Eur J Oper Res 2010; 204: 139–152.
17. Hyndman RJ, Koehler AB, Ord JK and Snyder, RD. Forecasting with exponential smoothing: the state space approach. Berlin Heidelberg: Springer, 2008.
18. Gould PG, Koehler AB, Ord JK, et al. Forecasting time series with multiple seasonal patterns. Eur J Oper Res 2008; 191: 207–222.
19. Box GEP, Jenkins GM, Reinsel GC, et al. Time Series Analysis: Forecasting & Control. Prentice Hall, 1994.
20. Engle RF. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica 1982; 50: 987–1007.
21. Taylor JW, de Menezes LM, McSharry PE, et al. A comparison of univariate methods for forecasting electricity demand up to a day ahead. Int J Forecast 2006; 22: 1–16.

22. Snyder RD, Ord JK, Koehler AB, et al. Forecasting compositional time series: A state space approach. Int J Forecast 2017; 33: 502–512.
23. García-Díaz JC, Trull Ó. Competitive Models for the Spanish Short-Term Electricity Demand Forecasting. In: Rojas I, Pomares H (eds) Time Series Analysis and Forecasting: Selected Contributions from the ITISE Conference. Cham: Springer International Publishing, pp. 217–231.
24. Lopes JAP, Matos MA, Saraiva JT, et al. The iberian electricity market – merging two commercial operation models into an integrated market structure.
25. Taylor, JW, de Menezes LM, and McSharry PE. A comparison of univariate methods for forecasting electricity demand up to a day ahead. Int J Forecast 2006; 22 (1): 1-16.
26. Hyndman, RJ., and Khandakar, Y. Automatic time series forecasting: The forecast package for R. Journal Of Statistical Software 2008; 27(3). https://doi.org/10.18637/jss.v027.i03.
27. De Livera, AM., Hyndman,RJ, y Snyder, RD. 2011. Forecasting time series with complex seasonal patterns using exponential smoothing Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association* 106 (496): 1513-27.

Numerical Estimation of GARCH Model through a Constrained Kalman Filter

Abdeljalil Settar, Nadia Idrissi Fatmi, and Mohammed Badaoui

LIPIM, ENSA Khouribga, University Sultan Moulay Slimane, Beni Mellal.
LaMSD, High School of Technology, University Mohamed first, Oujda, Morocco
{abdeljalilsettar,med.badaoui}@gmail.com

Abstract. In this paper, a research framework oriented towards parameters estimation of the GARCH(1,1) is proposed. The work is based on the Kalman filter and on the simultaneous perturbation stochastic approximation (SPSA) for the purpose of optimization. Kalman filter with constraints on the state variable is used in order to take into account the information eventually known a priori on volatility by eliminating a whole set of constraints imposed on the parameters that ensure the nonnegativity of the volatility which consequently allows to reduce the set of conditions on the parameters taking the asymmetric behaviours of the volatility into consideration. Simulations of this algorithm applied to the GARCH(1,1) model are presented and the efficiency of the proposed method is demonstrated.

Keywords: GARCH, Kalman Filter, Simultaneous perturbation stochastic approximation.

1 Introduction

Since they have been introduced by Engel [5] and Bollerslev [4], ARCH and GARCH models have shown remarkable effectiveness in modeling financial time series due to their ability to capture some stylish facts that characterize these series. In particular, GARCH model has provided a new stochastic-based pathway for volatility measurement based primarily on the notion of conditional variance.

The estimation techniques of GARCH models are mainly based on Ordinary Least Squares (OLS) and Quasi-Maximum Likelihood (QML) methods. Indeed, the OLS estimation was proposed by Engle [5] for ARCH models. Weiss [17] studied the theoretical properties of this estimator in the ARMA-GARCH case. On the other hand, the asymptotic results of the QML estimator were established by Ling and MacAleer [9], Francq and Zakoïan [6]. For GARCH(1,1), asymptotic properties were demonstrated by Lumsdain [10] under the strict stationarity hypothesis of the local QML estimator. Lee and Hansen [8] studied the convergence of the overall QML estimator under the assumption of second-order stationarity this faith.

However, these methods, as they have been put into effect, have been confronted

with the requirement to choose the initial values of the volatility process, which is practically unknown at first sight. Allal and Benmoumen [1] proposed an improvement in the estimation of the GARCH(1,1) model by Quasi-maximum likelihood (QML) based on the Kalman filter, the role of which is to estimate the volatility series in a recursive and optimal manner, and therefore, evaluate the function of QML without the need to set initial values a priori.
Moreover, the estimation of GARCH models by maximum likelihood leads to the production of volatility values that are not all nonnegative. Based on that, the subsequent step was the focus on the identification of the necessary and sufficient conditions that guarantee the nonnegativity almost everywhere of the conditional variance. To this end, Bolerslev [4] imposed the positivity constraint of the volatility equation parameters. Nelson and Cao [13], show that these constraints can be substantially reduced to a set of conditions based on the polynomial representation of GARCH model; necessary and sufficient for $p \leq 2$, and sufficient for $p \geq 3$. For the latter, Tsai and Chan [16] prove that the conditions of Nelson and Cao are also necessary. However, it has become apparent that this approach shows a number of limitations that make it an area for improvement. First, it is clear that the nonnegativity of the conditional variance imposes a number of inequalities that depend on $(p + q)$ parameters of the GARCH(p,q), in turn, increases the complexity of the problem in parallel with the increase of orders p and q. On the other hand, we see that the methods proposed above are intended to respond exclusively to the problem of the nonnegativity of volatility values and are not generalizable for exploiting more information available a priori on volatility, of which the positivity is a particular case, for the adjustment of parameter estimates. In addition, imposing the positivity of the parameters makes the standard GARCH model unable to capture the asymmetrical behaviour of the volatility which is characterized by the effects of sign and magnitude of the estimated parameters.
In this article, we focus our interest on estimating parameters of GARCH(1,1) model based on the Kalman filter with nonnegativity constraint on volatility. We do so, for two reasons; to divert the choice of its initial values, and to condition it directly without any recourse to the aforementioned constraints on the parameters. Thus, parameters can be fitted quite freely according to the known information (constraint) on volatility. Consequently, the set of conditions on the parameters is restricted to that of stationarity and the existence of moments.
The rest of the paper is planned as follows: In Section 2, we discuss the main properties of GARCH process in terms of stationarity and existence of moments applied to the particular GARCH(1,1). In Section 3, the stat-space representation of GARCH(1,1) is derived in addition to the volatility estimated by constrained Kalman filter. In Section 4, the Quasi log-likelihood function is constructed and optimized using the Simultaneous Perturbation Stochastic Approximation (SPSA). Furthermore, a convergence analysis of the algorithm is made under the assumptions cited in the literature. The performance of the proposed algorithm is evaluated in finite samples in Section 5, and some concluding remarks are given in Section 6 .

2 Terminology and Assumptions

Definition 1 (GARCH(p,q) process). *Let (η_t) be a sequence of independent and identically distributed (i.i.d) random variables with mean zero and variance one. (ε_t) is called the generalized autoregressive conditionally heteroscedastic process or GARCH(p,q) model if*

$$\varepsilon_t = \sigma_t \eta_t, \qquad t \in \mathbb{Z}$$

where (σ_t^2) is a nonnegative process such that

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \qquad t \in \mathbb{Z} \tag{1}$$

and

$$\omega > 0\ ,\ \alpha_i \geq 0\ ,\ i = 1, .., p\ \ and\ \ \beta_j \geq 0\ ,\ j = 1, .., q \tag{2}$$

Remark 1. The condition (2) on parameters ensures the nonnegativity of the conditional variance (1) (see Bollerslev [4]).

Proposition 1. *An equivalent ARMA(m,q) representation of the GARCH(p,q) process (ε_t) is given by*

$$\varepsilon_t^2 = \omega + \sum_{i=1}^{m} (\alpha_i + \beta_i) \varepsilon_{t-i}^2 + \nu_t - \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \qquad t \in \mathbb{Z}$$

where $m = max(p, q)$, $\alpha_i = 0\ for\ i > p$, $\beta_i = 0\ for\ i > q$ and (ν_t) represents the innovation corresponding to the process (ε_t^2) given by

$$\nu_t = \varepsilon_t^2 - \sigma_t^2 \tag{3}$$

In the following sections we will be interested in the main properties of the GARCH process in the particular case $p = q = 1$ which has the form

$$\varepsilon_t = \sigma_t \eta_t \tag{4}$$

$$\sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \tag{5}$$

with

$$\omega > 0 \tag{6}$$

and

$$\alpha \geq 0\ ,\ \beta \geq 0 \tag{7}$$

2.1 Stationarity

The strict stationarity of the GARCH(1,1) model has been studied by Nelson [11]. The second-order stationary condition of the GARCH(p,q) model is established by Bollerslev [4], of which it will be subject of the following property:

Property 1 (Second-order stationarity). A process (ε_t) satisfying the GARCH(1,1) model given by (4) and (5) is second order stationary if

$$\alpha + \beta < 1 \tag{8}$$

2.2 Moment Properties

For all $m \in \mathbb{N}^*$, the necessary and sufficient condition for the existence of the 2mth moment of the GARCH(1,1) model was provided by Bollerslev [4] as follows:

Theorem 1. *For the GARCH(1,1) process given by (4) and (5), the 2mth moment exists if and only if*

$$\mu(\alpha, \beta, m) = \sum_{i=0}^{m} C_m^i a_i \alpha^i \beta^{m-i} < 1$$

where

$$a_0 = 1,\ a_i = \prod_{j=1}^{i}(2i-1), \quad j \in \mathbb{N}^*$$

In the special case of $m = 2$, stationary fourth moment of the GARCH(1,1) process exists if and only if

$$\mu(\alpha, \beta, 2) = \sum_{i=0}^{2} C_2^i a_i \alpha^i \beta^{2-i} < 1$$

that is equivalent to

$$\beta^2 + 2\alpha\beta + 3\alpha^2 < 1 \tag{9}$$

Theorem 2. *Let (ε_t) be the GARCH(1,1) process as in (4) and (5) which parameters satisfy (8) and (9), and let's denote by μ_4 the fourth moment of (η_t), then*

$$E(\varepsilon_t^2) = E(\sigma_t^2) = \frac{\omega}{1-\alpha-\beta}$$

$$E(\varepsilon_t^4) = \mu_4 E(\sigma_t^4) = \frac{\omega^2(1+\alpha+\beta)}{(1-\alpha-\beta)(1-\mu_4\alpha^2-\beta^2-2\alpha\beta)}$$

The innovation process (ν_t) is a weak white noise process verifying

$$E(\nu_t^2) = \frac{\omega^2(1+\alpha+\beta)(\mu_4-1)}{(1-\alpha-\beta)(1-\mu_4\alpha^2-\beta^2-2\alpha\beta)}$$

Proof. (see Allal and Benmoumen [1])

3 State-Space Representation and Constrained Kalman Filter

3.1 State-Space representation of GARCH(1,1) process

The state space representation of GARCH(1,1) model as in (4) and (5), under the assumptions (8) and (9), proposed throughout this work, is obtained in the innovations from the works of Anderson and Moore [2] through the (ν_t) process described in (3) that is assumed to be Gaussian. Such representation is given by the following discrete-time equations

$$\sigma_t^2 = \omega + (\alpha + \beta)\sigma_{t-1}^2 + \alpha\nu_{t-1} \tag{10}$$

$$\varepsilon_t^2 = \sigma_t^2 + \nu_t \tag{11}$$

where (10) and (11) represent respectively the transition equation associated to the nonnegative state variable σ_t^2, and the measurement equation corresponding to the serie of observations (ε_t^2). Moreover (σ_0^2) is assumed Gaussian variable independent of $(\nu_t)_{t>0}$.The above assumptions provide the Gaussian distribution of the process $\left\{ \binom{\sigma_t^2}{\varepsilon_t^2}, t \geq 0 \right\}$.

3.2 Unconstrained volatility estimation

We consider the GARCH(1,1) state-space model in (10) and (11). Let us assume that $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ have been observed. Then, the unconstrained filtered (Updated) estimate of σ_t^2 and the variance of the associated errors are to be computed, that is

$$\widehat{\sigma}_{t/t}^2 = \widehat{\sigma}_{t/t-1}^2 + K_t(\varepsilon_t^2 - \widehat{\sigma}_{t/t-1}^2) \tag{12}$$

and

$$P_{t/t} = (1 - K_t)P_{t/t-1} \tag{13}$$

K_t is termed Kalman gain and given by

$$K_t = P_{t/t-1}(P_{t/t-1} + E\nu_t^2)^{-1} \tag{14}$$

The uncontrained predicted estimate of σ_t^2 is respectively given by

$$\widehat{\sigma}_{t/t-1}^2 = \omega + (\alpha + \beta)\widehat{\sigma}_{t-1/t-1}^2 \tag{15}$$

and the variance of the associated errors

$$P_{t/t-1} = (\alpha + \beta)^2 P_{t-1/t-1} + \alpha^2 E\nu_t^2 \tag{16}$$

The recursive relations (12)-(16) characterize the kalman filter algorithm with a choice of initial values such that

$$\widehat{\sigma}_{1/0}^2 = E(\sigma_1^2) \text{ and } P_{1/0} = Var(\sigma_1^2)$$

3.3 Constrained volatility estimation

Using the Kalman filter as above in the absence of the assumptions (6) and (7) do not ensure certainly of the almost sure nonnegativity of the conditional variance (σ_t^2). Thus, in order to take this constraint into account without having to impose the constraints (7)[1], we propose a correction of the volatility estimated in (3.2) by keeping it nonnegative through the probability density function (pdf) truncation method (Simon [14]) which consists in taking the probability density function computed by the Kalman filter (assuming that it is Gaussian) and truncates it at the constraint boundaries. The constrained conditional variance estimate then becomes equal to the mean of the truncated pdf.
In order to express the nonnegativity constraint of (σ_t^2), we suppose that at time t and for a constant N, empirically set, we have

$$\frac{1}{N} \leq \sigma_t^2 \leq N \tag{17}$$

The problem is to truncate the Gaussian pdf $\mathcal{N}(\widehat{\sigma}^2_{t/t_1}, P_{t/t-1})$ at constraints given in (17), and then find the mean $\widetilde{\sigma}^2_{t/t-1}$ and covariance $\widetilde{P}_{t/t-1}$ of the truncated pdf. These new quantities, $\widetilde{\sigma}^2_{t/t-1}$ and $\widetilde{P}_{t/t-1}$, become the constrained volatility estimate and its covariance.
We therefore initialize $i = 0$ such that

$$\widetilde{\sigma}^2_{ti} = \widehat{\sigma}^2_{t/t-1} \quad and \quad \widetilde{P}_{ti} = P_{t/t-1}$$

Now perform the following transformation:

$$\Sigma_{ti} = \frac{1}{\sqrt{\widetilde{P}_{ti}}}(\sigma^2_{t/t-1} - \widetilde{\sigma}^2_{ti}) \tag{18}$$

It can be seen that Σ_{ti} has a mean of 0 and variance 1. Furthermore inequality (17) is transformed as follows:

$$l_{ti} \leq \Sigma_{ti} \leq u_{ti}$$

With

$$l_{ti} = \frac{1 - N\widetilde{\sigma}^2_{ti}}{N\sqrt{\widetilde{P}_{ti}}} \quad and \quad u_{ti} = \frac{N - \widetilde{\sigma}^2_{ti}}{\sqrt{\widetilde{P}_{ti}}}$$

We define $\Sigma_{t,i+1}$ as the random variable that has the pdf of Σ_{ti} truncated and normalized between the limits l_{ti} and u_{ti}. Let μ_Σ and σ^2_Σ be respectively the mean and the variance of $\Sigma_{t,i+1}$. We take then the inverse of the transformation (18) to find the mean and variance of the volatility estimate after enforcement of constraint (17). Thus we obtain

$$\widetilde{\sigma}^2_{t,i+1} = \sqrt{\widetilde{P}_{ti}}\, \mu_\Sigma + \widetilde{\sigma}^2_{ti}$$

$$\widetilde{P}_{ti} = \sigma^2_\Sigma$$

[1] The assumption (6) is kept to ensure the nonnegativity of the unconditional variance.

4 Estimation of GARCH(1,1) model based on the Constrained Kalman Filter

4.1 Quasi-likelihood function

Let (ε_t) be the GARCH(1,1) model defined by (4) and (5) and let's denote by $\theta = (\omega, \alpha, \beta)'$ the parameters vector satisfying only conditions (6), (8) and (9). Let Θ be the subset of $\mathbb{R}^3$ such

$$\Theta = \left\{\theta \in \mathbb{R}^3 / \ \omega > 0 \ , \ |\alpha + \beta| \leq 1 \ , \ \beta^2 + 2\alpha\beta + 3\alpha^2 < 1\right\}$$

We propose estimating θ by using quasi-maximum likelihood since we do not make any assumption on the distribution of the *iid* variables η_t.

From the observed data $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$, the quasi log-likelihood function is given, for all $\theta \in \Theta$ by

$$L_n(\theta; \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) = -\frac{n}{2} log(2\pi) - \frac{n}{2}\left(\frac{1}{n}\sum_{t=1}^{n}\frac{\varepsilon_t^2}{\widetilde{\sigma}^2_{t/t-1}(\theta)} + log(\widetilde{\sigma}^2_{t/t-1}(\theta)\right)$$

Thus, maximizing the log-likelihood function above is equivalent to minimizing, with respect to $\theta \in \Theta$

$$l_n(\theta; \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n) = \frac{1}{n}\sum_{t=1}^{n}\frac{\varepsilon_t^2}{\widetilde{\sigma}^2_{t/t-1}(\theta)} + log(\widetilde{\sigma}^2_{t/t-1}(\theta)) \tag{19}$$

where the $\widetilde{\sigma}^2_{t/t-1}(\theta)$ are obtained recursively, for $t \geq 1$, by the constrained Kalman filter used in (3.3). Therefore, the quasi-likelihood function is completely defined since the nonnegativity of $\widetilde{\sigma}^2_{t/t-1}(\theta)$ is ensured without considering assumptions (7). On the other hand, we do not require initial values ε_0 and σ_0^2 to construct the quasi-likelihood which are not known in practice and are essential for estimating by quasi-maximum likelihood (Allal and Benmoumen [1]).

4.2 Stochastic Optimization of Quasi-Likelihood function

As mentioned before, the quasi-likelihood function has a random multi-stage character. It is constructed through the underlying process $\widetilde{\sigma}^2_{t/t-1}$ estimated by the constrained Kalman filter at each stage t, $t = 1, \ldots, n$. Thus, the optimization (minimization) of l_n should be done randomly one stage at a time over n stages after observing the $\widetilde{\sigma}^2_{t/t-1}$ in each stage t. In this case, a stochastic search algorithm is a wise choice for the minimization of the l_n function (Bhatnagar and al.[3]). Spall [15] invented the simultaneous perturbation stochastic approximation (SPSA) which relies on the approximation of the gradients using only two measurements of l_n for a parameter vector of any dimension and exhibits fast convergence. The step-by-step summary below shows how SPSA was applied to minimize the l_n function.

Step 1 : **Initialization and coefficient selection.**

- Select counter index $k = 0$;
- Give initial $\theta_0 \in \Theta$ and nonnegative coefficient a,c,A,α, and γ;
- Compute again sequences $a_k = (A + k + 1)^{-\alpha}$ and $c_k = (k + 1)^{-\gamma}$

Step 2 : **Generation of the simultaneous perturbation vector**.

- Generate a 3-dimensional random perturbation vector Δ_k. A practical choice for each component of Δ_k is to use a Bernoulli distribution, ± 1-valued with probability of $\frac{1}{2}$

Step 3 : **Evaluations of l_n function**.

- Check the existence of 2nd moment of (6)
- Obtain two measurements of the l_n function based on the simultaneous perturbation around the current $\widehat{\theta}_k$, ie. $y_k^+(\widehat{\theta}_k) = l_n(\widehat{\theta}_k + c_k\Delta_k) + \varepsilon_k^+$ and $y_k^-(\widehat{\theta}_k) = l_n(\widehat{\theta}_k - c_k\Delta_k) + \varepsilon_k^-$.

Step 4 : **Approximation of gradient** .

- Generate the simultaneous perturbation approximation of the gradient $g(\widehat{\theta}_k)$:

$$g(\widehat{\theta}_k) = \frac{y_k^+(\widehat{\theta}_k) - y_k^-(\widehat{\theta}_k)}{2c_k}(\Delta_{k1}, \Delta_{k2}, \Delta_{k3})'$$

where Δ_{ki}, $i = 1, 2, 3$ is the ith component of Δ_k vector.

Step 5 : **Update of $\widehat{\theta}_k$ estimate**.

- Check the second order stationarity condition
- Use the stochastic approximation:

$$\widehat{\theta}_{k+1} = \widehat{\theta}_k - a_k g(\widehat{\theta}_k) \tag{20}$$

Step 6 : **Iteration or termination.**

- Return to Step 2 with $k + 1$ replacing k;
- The algorithm terminates if the sequence (θ_k) converges with order of 10^{-3} or the maximum allowable number of iterations has been reached.

Remark 2.

1. A possible choice of λ and γ is : $\lambda = 0.602$, $\gamma = 0.101$.
2. The parameter A is equal to 10% (or less) of the number of iterations. (see Spall [15])

4.3 Convergence Analysis

Before presenting the simulation results of the proposed algorithm, we have to check the assumptions that ensure the convergence of the iterative expression (20). Theses assumptions were established by Spall [15] and refined by Bhatnagar and al. [3] and are used here as a starting point to prove the theoretical convergence of the SPSA method applied to the likelihood function (19). We shall carry out our analysis under the further assumptions:

(A1) The likelihood function l_n is Lipschitz continuous and is differentiable with bounded second order derivatives. Further, the map $J_n : \mathbb{R}^3 \to \mathbb{R}^3$ defined as $J_n(\theta) = -\nabla l_n(\theta), \forall \theta \in \mathbb{R}^3$ is Lipschitz continuous.

(A2) The step-sizes a_k,$c_k > 0$, $\forall k$ and $a_k, c_k \to 0$ as $k \to 0$ such

$$\sum_k a_k = \infty \text{ and } \sum_k \left(\frac{a_k}{c_k}\right)^2 < \infty.$$

(A3) $\varepsilon^+_{k\geq 0}$, $\varepsilon^-_{k\geq 0}$, are independent random vectors having a common distribution and finite second moments.

(A4) The random variables Δ_{ki}, $k \geq 0$, $i = 1, 2, 3$, are mutually independent and mean-zero, have a common distribution, and satisfy $E\left[1/\Delta^2_{ki}\right] \leq \overline{K}$,$\forall$ $k \geq 0$, for some $\overline{K} < \infty$.

(A5) The iterates (20) remain uniformly bounded almost surely,i.e.,

$$sup_k \| \theta_k \| < \infty \quad a.s.$$

(A6) The set H containing the local minima of l_n is a compact subset of $\mathbb{R}^3$.

Discussion

In order to make **(A1)** satisfied, we assume that ω is bounded, i.e, $\exists\, \overline{\omega} > 0$ such $0 < \omega \leq \overline{\omega}$. The idea is to extract from Θ the smallest compact (bounded and closed subset of Θ) on which the smoothness of l_n is guaranteed. In this regard, let's consider the set

$$\Theta_\eta = \big\{\theta = (\omega, \alpha, \beta) \in \mathbb{R}^3 / \ \eta \leq \omega \leq \overline{\omega}\ ,\ |\alpha + \beta| \leq 1 - \eta\ ,\\ 0 \leq \beta^2 + 2\alpha\beta + 3\alpha^2 \leq 1 - \eta\big\}$$

It is important to note that, Θ_η is bounded and closed set. In addition to the assumption on ω, $\Theta_\eta \to \Theta$ for small values $\eta > 0$. Thus, l_n as well as ∇l_n comply with the conditions of **(A1)** over the set Θ_η (see the Appendix). The choice of a_k, c_k and Δ_k made in the steps 1 and 2 of the SPSA algorithm above satisfies **(A2)** and **(A4)** (see Bhatnagar and al. [3]). **(A5)** ensures that the iterates by recursion (20) remain stable. Through algorithmic conditioning structures applied in step 3 and step 5, **(A5)** can be practically satisfied by imposing on the iterates after each update to remain in Θ. Let's denote by H_η the set of the local minima of l_n over Θ_η. Then, assumption **(A6)** on H_η can be easily checked from the properties of Θ_η (see the Appendix).

Theorem 3. *Under Assumptions **(A1)-(A6)**, the parameter updates (20) satisfy*

$$\theta_k \to H_\eta \quad a.s$$

Proof. (see Bhatnagar and al. [3])

5 Numerical results

In this section, numerical simulations are presented in order to assess the performance of the proposed estimation method. Our aim is to make a comparative study between the estimations obtained by quasi-maximum likelihood based on constrained Kalman filter proposed in this work, and quasi-maximum likelihood method considered in the literature. On the other hand, we show that along with the extensions of the GARCH model (EGARCH,GJR-GARCH,...), the proposed algorithm allows the standard GARCH model to capture the asymmetry effects of shocks ε_t^2 on the volatility in magnitude as well as by sign, which the estimation methods given by the literature do not ensure because of the nonnegativity conditions imposed a priori on the parameters.
First, we simulate a GARCH(1,1) model with parameters vector θ_1=(1,0.4 ,0.2), and noise process $\eta_t \sim iid(0,1)$ assumed to be Gaussian. The sample size used is 100 splits into 1000 replications. The results of this simulation is summarized in table 1, where we denote respectively by QMLE and QMLCKF, the quasi-maximum likelihood estimators, and the estimation obtained by our algorithm. For each estimators, the mean is given as the sample estimates of the parameters. In addition, the mean error and the MSE are used to compare the performance of the two approaches. Secondly, we simulate 100000 observations of an EGACH(1,1)[2] model with parameters vector θ_2=(1,-0.3,0.5,0.01), and Gaussian noise process $\eta_t \sim iid(0,1)$. The simulated data is fitted by a GARCH(1,1) according to the proposed algorithm. The results of this simulation is summarized in table 2.
The numerical results showed that the proposed algorithm is able to improve the quality of quasi-maximum likelihood estimation. Indeed, it can be observed that the proposed algorithm led to a decrease in mean error (about 9% for ω, 98% for α and 86% for β) and in MSE (about 18% for ω, 98% for α and 90% for β), obtained by quasi-maximum likelihood estimation. Furthermore, the proposed algorithm was able to capture the asymmetry effect of ε_{t-1}^2 by the negative estimate value of $\widehat{\alpha}_{QMLCKF}$ which equals to -0.4547 (Table 2). In general, it can be seen that our approach provides estimates with small absolute deviations being below 0.25 .

[2] For an EGARCH(1,1) model, the volatility process is given by

$$log(\sigma_t^2) = \omega + \alpha\varepsilon_{t-1}^2 + \gamma(\left|\varepsilon_{t-1}^2\right| - E\left|\varepsilon_{t-1}^2\right|) + \beta log(\sigma_{t-1}^2)$$

where ω, α, β and γ are real numbers.(see Nelson [12])

		QMLCKF			QML		
Parameters	True Values	Mean	Mean Error	MSE	Mean	Mean Error	MSE
ω	1	0.0934	0.9065	0.8219	9.35×10^{-7}	0.9999	0.9999
α	0.4	0.3949	0.0078	0.0006	0.3605	0.3735	0.0410
β	0.2	0.1153	0.0868	0.0077	0.2691	0.6112	0.0804

Table 1. Mean, Mean error and MSE of estimated parameters

Parameters	True Values	Estimates	Absolute Deviation
ω	1	1.2446	0.2446
α	-0.3	-0.4547	0.1547
β	0.5	0.6537	0.1537

Table 2. Mean and Absolute Deviation of estimated parameters

6 Conclusions

This work presents a numerical approach to estimate the parameters of the GARCH(1,1) model. This method is based on the constrained Kalman filter by which the condition of nonnegativity of volatility could be exploited without imposing positivity conditions of the parameters. The numerical results demonstrate the effectiveness of the proposed method and show that it is appropriate to estimate the parameters of a GARCH model taking into account the asymmetric behaviours of the volatility.

References

1. Allal, J., Benmoumen, M.: Parameter estimation for GARCH(1, 1) models based on Kalman filter. Advances and Applications in Statistics. 25(2), 115–130 (2011)
2. Anderson, B., Moore, J.: Optimal Filtering. pp. 230–238, Prentice Hall (1979)
3. Bhatnagar, S., Prasad, H.L., Prashanth, L.A.: Stochastic Recursive Algorithms for Optimization. vol. 434, pp. 44–47. Springer, London (2013).
4. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. J. Econom. 31(3), 307–327 (1986)
5. Engle, R.F.: Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. Econometrica. 50(4). 987 (1982)
6. Francq, C., Zakoïan, J.M.: GARCH models: Structure, Statistical Inference and Financial Applications. John Wiley & Sons (2010)
7. He, C., Terasvirta, T.: Properties of the Autocorrelation Function of Squared Observations for Second-order Garch Processes Under Two Sets of Parameter Constraints. J. Time Ser. Anal. 20(1), 23–30 (1999)
8. Lee, S.W., Hansen, B.E.: Asymptotic theory for the GARCH(1,1) quasi-maximum likelihood estimator. Econometric Theory. 10(1),29–52 (1994)
9. Ling, S., McAleer, M.: Asymptotic theory for a vector ARMA-GARCH model. Econometric Theory. 19(2), 280–310 (2003)

10. Lumsdaine, R.L.: Consistency and asymptotic normality of the quasi-maximum likelihood estimator in IGRACH(1,1) and covariance stationary GARCH(1,1) models. Econometrica. 64(3), 575–596 (1996)
11. Nelson, D.B.: Stationarity and persistence in the GARCH(1,1) model. Econometric Theory. 6(3), 318–334 (1990)
12. Nelson, D. B.: Conditional heteroskedasticity in asset returns: A new approach. Econometrica: Journal of the Econometric Society, 347–370 (1991)
13. Nelson, D.B., Cao, C.Q.: Inequality Constraints in the Univariate GARCH Model. J. Bus. Econ. Stat. 10(2). 229–235 (1992)
14. Simon, D.: Optimal state estimation. pp. 218–223. John Wiley & Sons (2006)
15. Spall, J.C.: Implementation of the simultaneous perturbation algorithm for stochastic optimization. IEEE Trans. Aerosp. Electron. Syst. 34(3). 817–823 (1998)
16. Tsai, H., Chan, K.S.: A note on inequality constraints in the GARCH model. Econometric Theory. 24(3). 823–828 (2008)
17. Weiss, A.A.: Asymptotic theory for ARCH models: estimation and testing. Econometric Theory. 2(1). 107–131 (1986)

Appendix: Proof of Discussion

– *Proof of **(A1)***: Let's denote by ϕ_t, ψ_t and v the functions defined on Θ and for all $t \in \{1, ..., n\}$ by :

$$\phi_t = \sigma^2_{t/t-1} \ , \ \psi_t = P_{t/t-1} \ \ and \ \ v = E(\nu_t^2)$$

From (12) and (15), we obtain for all $\theta \in \Theta$

$$\phi_t(\theta) = \omega + (\alpha + \beta)\frac{\psi_{t-1}(\theta)}{\psi_{t-1}(\theta) + v(\theta)}(v(\theta)\phi_{t-1}(\theta) + \varepsilon^2_{t-1}) \tag{21}$$

Furthermore, from (13) and (16), we have for all $\theta \in \Theta$

$$\psi_t(\theta) = \left[(\alpha + \beta)^2 \frac{\psi_{t-1}(\theta)}{\psi_{t-1}(\theta) + v(\theta)} + \alpha^2\right] v(\theta) \tag{22}$$

Using a mathematical induction on (22) for all $t \in \{1, \ldots, n\}$, it can be easly shown from (8) and (9) that $\psi_{t-1} \in C^\infty(\Theta)$ under the assumption $\psi_0 = \psi_1 = Var(\sigma_1^2)$ since $v \in C^\infty(\Theta)$ as well as $\psi_t(\theta) + v(\theta) > 0$ for all $\theta \in \Theta$. In the same way, from (3.2) and (21), we show that $\phi_t \in C^\infty(\Theta)$. In the other hand, by construction, Θ_η is closed as a union of closed intervals of $\mathbb{R}^3$, and is also bounded since for all $\theta \in \Theta_\eta$, $0 < \omega \leq \overline{\omega}$, $|\alpha| < \frac{1}{\sqrt{2}}$, and $|\beta| < 1 + \frac{1}{\sqrt{2}}$. Thus, Θ_η is a compact set on which l_n is of class C^∞. Then **(A1)** is satisfied as $\eta \to 0$.

– *Proof of **(A6)*** : To prove that, it is sufficient to assume that H_η is a finite set of Θ_η. Hence, H_η is bounded and is closed.

Using Time-Series and Forecasting to Manage Type 2 Diabetes Conditions (GH-Method: Math-Physical Medicine)

Gerald C. Hsu
eclaireMD Foundation, USA

1 Introduction

This paper describes the author's application of Time-Series Analysis and Forecasting to manage Type 2 Diabetes (T2D) conditions.

2 Method

The author utilizes the GH-Method: math-physical medicine to manage metabolic disorder diseases especially diabetes. Initially, he observed various disease phenomena. Therefore, he recorded big volume of related data, derived necessary and applicable mathematical equations, utilized suitable computational tools, including time-series analysis, spatial analysis, frequency domain analysis, and artificial intelligence. As a result, he combined them with medical domain knowledge in order to forecast the forthcoming outcomes to interpret the new findings or discoveries regarding human health. In this paper, he disregards the theoretical discussion of time-series analysis in order to focus on application and certain results from his diabetes research by using timeseries analysis and forecasting method.

3 Results

Here are some of the results from time-series analysis and forecasting:
(1) Weight: He developed a weight prediction model based on food portion, exercise, and certain metabolism respects and achieved 99.8% linear accuracy with a correlation coefficient (R) of 90% to compare with actual weight. Weight contributes ~85% of FPG formation.
(2) Fasting plasma glucose (FPG) in early morning: Using time-series analysis, he obtained R=70% between weight and FPG.
(3) Postprandial plasma glucose (PPG) at two hours after fist-bite of meal: Using time-series analysis, he obtained R= +45% between carbs/ sugar intake and PPG and R= -59% between post-meal walking and PPG. Combined carbs/sugar and walking contributes ~80% of PPG formation. He achieved 100% linear accuracy and R=85% between predicted and actual PPG.
(4) Hemoglobin A1C or HbA1C (A1C): The medical community uses A1C as the measuring yardstick to determine the severity of the patients' diabetes conditions.

There are no consistent conversion ratios available between glucose and A1C values. Therefore, the author applied statistics tools (including time-series analysis, spatial analysis, and frequency-domain analysis) and engineering approximation modeling to build up an effective A1C forecasting model. In comparison of this mathematically forecasted A1C results and lab tested A1C results (quarterly data due to insurance constraints), he achieved a linear accuracy of 96% and R=54%.

4 Conclusion

By using time-series analysis method, this clinical case study of more than four years, encompassing 1,488 days and ~500,000 big data, has demonstrated its powerful forecasting capability on weight, glucose, and diabetes control.

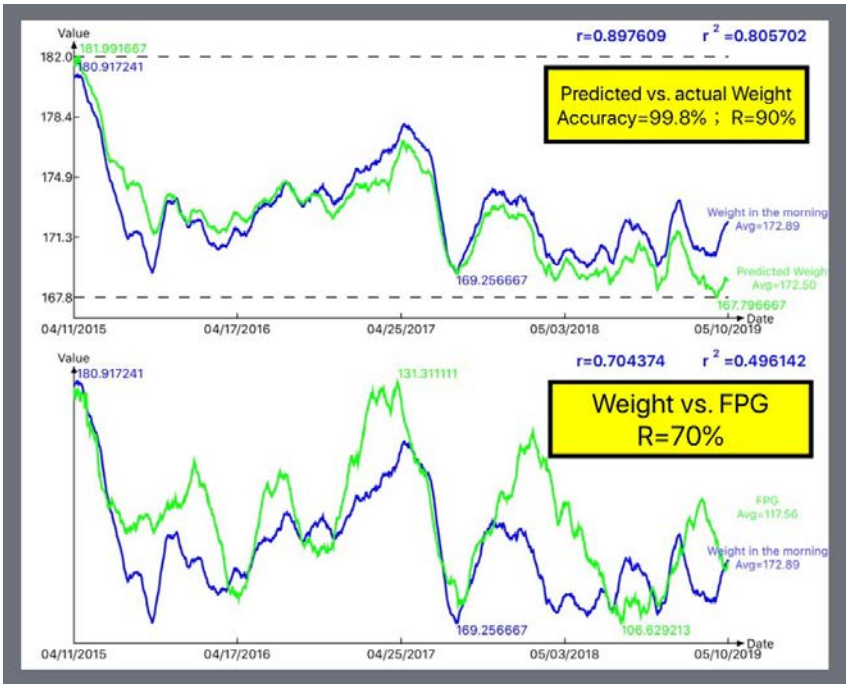


Fig. 1. Weight and FPG

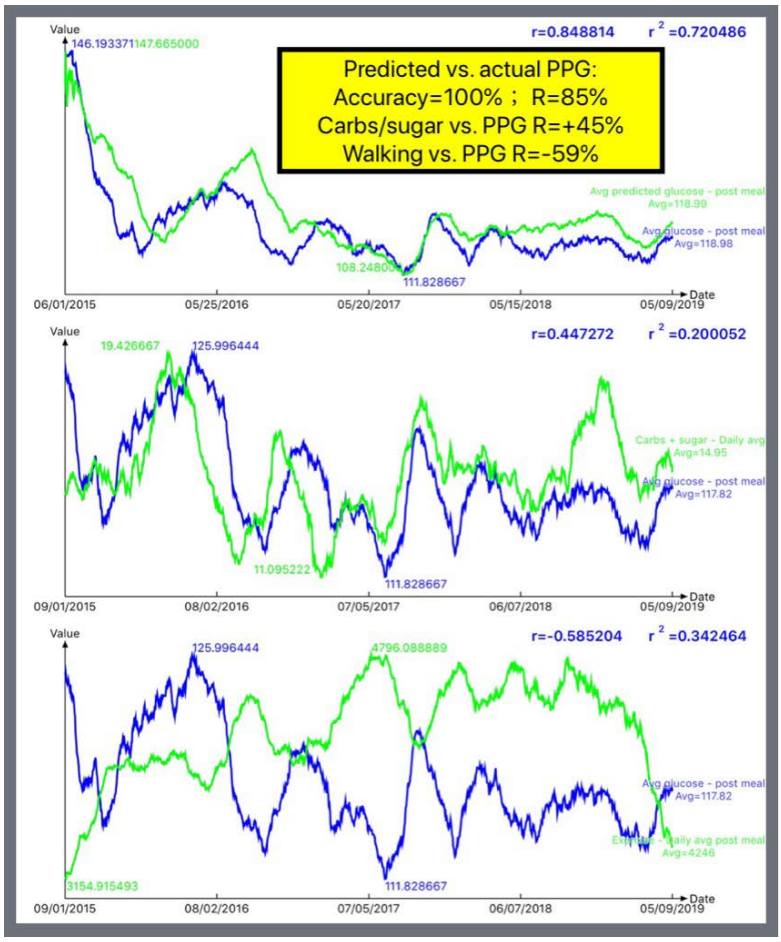


Fig. 2. Carbs/sugar (+R) and Walking (-R) vs. PPG

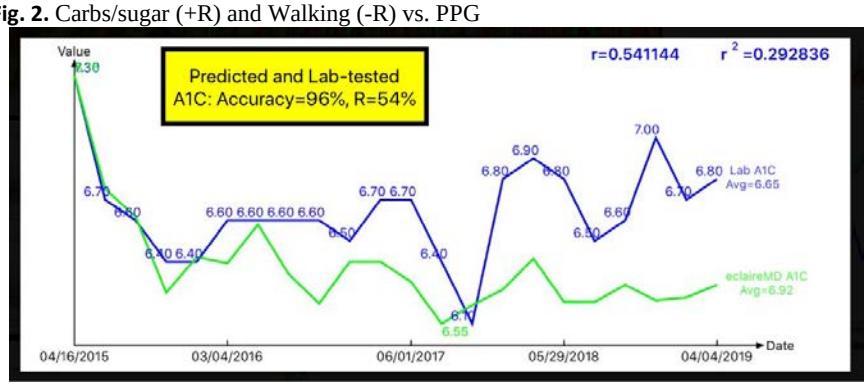


Fig. 3. HbA1C

Inflation Rate Forecasting: Extreme Learning Machine as a Model Combination Method

Jeronymo Marcondes Pinto[1] and Emerson Fernandes Marçal[2]

[1] Brazilian Ministry of Economics
jeronymomp@gmail.com
[2] Sao Paulo School of Economics
emerson.marcal@fgv.br

Abstract. Inflation rate forecasting is one most discussed topics on time series analysis due to its importance on macroeconomic policy. The majority of these papers findings point out that forecasting combination methods usually outperform individual models. In this sense, we evaluate a novel method to combine forecasts based on Extreme Learning Machine Method (Huang et al., 2004), which is becoming very popular but, to the best of our knowledge, has not been used to this purpose. We test Inflation Rate forecasting for four Latin American countries, for one, two, three, ten, eleven and twelve steps ahead. The models to be combined are automatically estimated by *R forecast* package, as SARIMA, Exponential Smoothing, ARFIMA, Spline Regression, and Artificial Neural Networks. Another goal of our paper is to test our model against classical combination methods such Granger Bates, Linear Regression and Average Mean of models as benchmarks, but also test it against basic forms of new models in the literature, like Diebold and Shin (2018), Garcia et al. (2017) and Wang et al. (2018b). Therefore, our paper also contributes to the discussion of forecast combination by comparing versions of some methods that have not been tested against each other. Our results indicate that none of these methods have an indisputable superiority against the others, however the Extreme Machine Learning Method proved to be the most efficient of all, with the smaller Mean Absolute Error and Mean Squared Error for its predictions.

1 Introduction

The inflation rate is a core indicator of economic activity. This indicator is closely monitored by policy makers, practitioners, portfolio management and economic researchers due to its importance on macroeconomic policy. Therefore inflation rate time series forecasting is a trending topic on forecasting literature.

Inflation rate forecasting was discussed on classical papers like Stock and Watson (1999) and Deutsch et al. (1994), which persists until nowadays with Zhang (2019) and Tallman and Zaman (2017), for example.

One of the main discussions on inflation forecasting literature is the role of forecasting combination on predictability improvement of models. Most of the

papers indicate that the combination of models usually increases forecasting performance.

The seminal work of Bates and Granger (1969) suggested that a simple forecast combination, such as simple or rolling weighted averages, outperform individual models. The importance of model combinations has been highlighted in recent papers, such as Hsiao and Wan (2014) and Chan and Pauwels (2018).

Timmermann (2006) reviews classical methods to combine forecasts, as generating prediction weights based on a Linear Regression or giving equal weights to all methods, like an Average Mean of all models. Bates and Granger (1969) proposed to combine forecasts based on a weighted average of each model's mean squared errors.

However, the discussion of which is the best way to combine forecasts is still open to debate, with a lot of papers suggesting new methods to obtain higher forecast accuracy. Diebold and Shin (2018) propose a forecast ensemble based on LASSO (*Lasso*), that selects and shrinks toward equal combining weights. Garcia et al. (2017) developed a study based on a Model Confidence Set (*MCS*) by Hansen et al. (2011), which allows the user to equally combine forecasts selected by MCS. Wang et al. (2018b) evaluates the performance of a forecast combination with weights calculated by an Artificial Neural Network, with a multilayer perceptron architecture (*Mlp*). All those papers develop a study with their proposed methods against some classical benchmarks. This is increasingly linked to the actual research on Machine Learning literature and its possibilities to improve the forecast.

Based on the work of Huang et al. (2004), we propose a new way to combine forecasts, with weights estimated by Extreme Learning Machine method (ELM). This method has been proved as a very efficient machine learning approach to forecasting, with good accuracy results in the literature, as discussed in Wang et al. (2018a) and Behbahani et al. (2018), and excellent algorithm performance. In this sense, this paper contributes to the forecasting literature evaluating a new method to combine forecasts.

The following time series models are used to generate forecasts to be combined or selected: exponential smoothing, SARIMA, artificial neural networks (ANNs), ARFIMA, and Spline Regression. All of these models' functional specifications are automatically provided by *forecast R* package.

We run a pseudo-real-time forecast exercise to evaluate the forecasting performance of our strategy by applying it to the Inflation Rate of the following Latin American countries: Brazil, Mexico, Chile, and Peru. We forecast this series for one, two, three, ten, eleven and twelve steps ahead. We opt to test our proposed method on Latin American countries due to its historical inflation rate instability.

We compare our forecast results to classical benchmarks, such as random walk (*RandomWalk*), Average Mean (*AverageMean*), and Linear Regression of forecasts (*LinearRegression*) like discussed in Timmermann (2006), as well as the one proposed by Bates and Granger (1969) (*GB*).

Moreover, another goal of this paper is to test some of those new methods to combine forecasts that have been published in the forecasting literature, focusing on machine learning aggregating models. Our method is compared to versions of the recent models proposed by Garcia et al. (2017), Diebold and Shin (2018), and Wang et al. (2018b). To the best of our knowledge, no one has tried to compare those approaches accuracy against each other. In addition to these models, we are going to test a combination method based on Ridge Regression (*Ridge*), as an extension of Diebold and Shin (2018) work.

The reader must be attentive over the approach in this study regarding the use of those methods. We are not necessarily using the same algorithm used by the original author, but we base on their central idea. For example, in the case of Wang et al. (2018b), the author uses a network architecture and a back-propagation schema specific to his problem, which we do not replicate here, but only the central idea of using a *Mlp* as a way of estimating the weights of the combination.

Therefore, our paper contributes to the discussion of new forecasting methods combined with Machine Learning techniques, by proposing a new method based on Extreme Machine Learning. Our paper also contributes to the evaluation of some new methods that have not been tested against each other also.

This paper is organized as follows. Section 1 discusses our proposed strategy to generate forecasts. Section 2 reports our results and evaluates our data. Section 3 discusses the merits and pitfalls of our strategy. Finally, some concluding remarks are drawn.

2 Material and Methods

2.1 Extreme Learning Machine Method (*Elm*) and the proposed framework

The Elm algorithm was proposed by Huang et al. (2004) and it's based on a single hidden layer feedforward neural network (SLNN), but it intends to solve the usual problems addressed by artificial neural networks literature, as method's speed.

For M arbitrary samples (x_i, t_i), with $x \in \mathbf{R}^n$ being the input and $t \in \mathbf{R}^m$ output of an given econometric problem, a standard way to model an SLNN with an activation function given by $g(x)$ is:

$$\sum_{i=1}^{M} \beta_i g_i(x_i) = \sum_{i=1}^{M} \beta_i g_i(w_i x_i + b_i) = o_j, j = 1, .., N. \tag{1}$$

Where $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is the vector of weights that connects input layer to hidden layer, $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{im}]^T$ is the set of weights between output and hidden nodes, o_j is the tested output and b_i is the threshold of the ith hidden neuron.

That SLNN with N hidden neurons and $g(x)$ activation function can approximate these N samples with zero error, as $\sum_{j=1}^{n} ||o_j - t_j|| = 0$, there exist β_i, w_i and b_i such that:

$$\sum_{i=1}^{M} \beta_i g_i(w_i x_i + b_i) = t_j, j = 1, .., N. \tag{2}$$

This equation can be written as follows:

$$H\beta = T. \tag{3}$$

Where: $\mathrm{H} = \begin{pmatrix} g(w_1 x_1 + b_1) & ... & g(w_N x_1 + b_N) \\ \vdots & ... & \vdots \\ g(w_1 x_N + b_1) & ... & g(w_N x_N + b_N) \end{pmatrix}$,
$\beta = \begin{pmatrix} \beta_1^T \\ \vdots \\ \beta_N^T \end{pmatrix}$ and $T = \begin{pmatrix} t1^T \\ \vdots \\ tN^T \end{pmatrix}$.
According to Huang et al. (2004), unlike common understanding that all parameters of SLNN must be tuned, experiments show that they can be arbitrarily given. Huang et al. (2004) points out that for small values for the parameters in the activation function, train an SLNN is simply equivalent to finding a least-squares solution β of the linear system $H\beta = T$:

$$min_\beta ||H(w_1, ..., w_n, b_1, ..., b_n)\beta - T||. \tag{4}$$

Based on this method, our paper proposes to solve the problem given by (4) to obtain the weights to combine different forecasts models. Therefore, our inputs must be the forecasts of different models while the output is the actual value of the predicted variable. To the best of our knowledge, no paper has used this approach.

Elm architecture is defined by a process of cross-validation applied to different sets of networks. Our method chooses the number of hidden nodes by analysis of the least mean absolute error generated on the training set. For our purposes, we tested five, ten, fifteen, twenty, twenty five, and third possible hidden nodes.

2.2 New methods to forecast combination as benchmarks

In this section we are going to expose new methods, with Machine Learning framework, that were used to combine forecasts in recent studies. Basically, in all of those methods the explanatory variable is given by each forecast to be combined while the dependent variable is the series value to be predicted.

LASSO and Ridge Regression

Diebold and Shin (2018) proposed using a LASSO-based procedure that selects and shrinks toward equal combining weights (*Lasso*). These authors aim to find a method that could select the best predictors to combine forecasts. In this study, we are using a basic form of LASSO regression as a way to develop a forecast ensemble.

Lasso estimate is given by:

$$\beta^{Lasso} = argmin_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2. \quad (5)$$

subject to $\sum_{j=1}^{p} |\beta_j| \leq t$.

Based on this maximization problem, it's possible to establish the weights for each forecast (x_{ij}), even zero.

In the same way, Ridge Regression estimate (*Ridge*) is given by:

$$\beta^{Lasso} = argmin_{\beta} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2. \quad (6)$$

subject to $\sum_{j=1}^{p} \beta_j^2 \leq t$. Where x_{ij} is going to be each of the forecasts to be combined in our experiment.

It's possible to infer that the basic difference between Ridge Regression and LASSO is maximization restriction, where it's given by the absolute value of β on the latter and it's square on Ridge Regression. In our study, we are using LASSO and Ridge Regression with a basic framework, without some specifications exposed at Diebold and Shin (2018).

Artificial Neural Network

Wang et al. (2018b) is one of the recent studies that use Artificial Neural Network as part of a framework of forecast combination. Basically, they use a Multilayer Perceptron Artificial Neural Network (*Mlp*) given by:

$$y_i = \sum_{j=1}^{m} f(w_{ij}x_j + b_i). \quad (7)$$

In this study we are using a feedforward neural network, where each input x_j feeds it's value to hidden neuron, known as hidden layers, until the final output is obtained from the neural network. During its passage by each neuron, the input value is multiplied for its respective weight w_{ij}. For more details about this method and *Mlp* architecture see Friedman et al. (2001).

In this paper we are going to use a simple *Mlp* architecture, with 3 layers and logistic activation function, which was defined by experimentation with best results.

MCS

Hansen et al. (2011) introduced the concept of Model Confidence Set (*MCS*). *MCS* is a set of models that is constructed such that it will contain the best model with a given level of confidence. The *MCS* is analogous to a confidence interval for a parameter. Garcia et al. (2017) evaluated the use of *MCS* on selecting the best combination of models to generate a forecast ensemble.

2.3 Classical methods to forecast combination as benchmarks

To evaluate any strategy, it is important to choose proper benchmarks. If a strategy is unable to outperform forecasts obtained from simple benchmarks,

it should be abandoned. Simple benchmarks serve as a lower bound to assess any strategy. For example, if the analyst wants to forecast an exchange rate, random walk is a difficult benchmark to be beaten (Rossi (2013) and Meese and Rogoff (1983)). An autoregressive model of order 1 is a difficult benchmark when forecasting a consumer price index (Castle et al. (2013) and Stock and Watson (2002)). A forecast obtained from a double difference model can be a difficult benchmark in data where the data generator process faces structural breaks (Clements and Hendry (2001)).

We are using the following classic benchmarks:

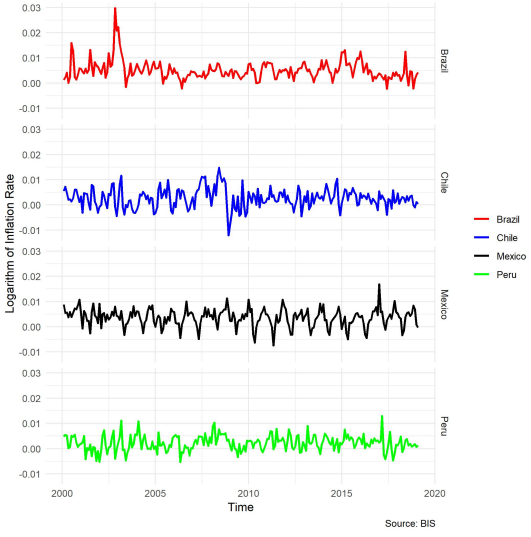
- Average Forecast Combination (Timmermann (2006));
- Linear Regression of forecasts (Timmermann (2006));
- Granger Bates Method (Bates and Granger (1969));
- Random Walk (Timmermann (2006)).

2.4 Data and Strategy

We tested our models on Inflation Rate Data for four Latin American countries: Brazil, Peru, Mexico, and Chile. This choice was made based on Latin American history of high rates of inflation during the last century. In this sense, modelling the inflation process, with the possibility to forecast it is very useful to local authorities.

All data was obtained from the Bank of International Settlements (BIS) and all series can be obtained at https://www.bis.org/. Their frequency is monthly and its dynamic evolution can be seen in Figure 1.

Fig. 1. Logarithm of Inflation Rate for Five American Latin Coutries



The strategy to our forecasting exercise is based on the following schema:

- Training set equals 50% of data;
- Validation Set equals 40% - "number of steps ahead to forecast" + 1 of data;
- Test set equals total data minus (Training Set + Validation Set).

All series were tested with Augmented Dickey-Fuller test and the result was that there is an unit root with 1% of confidence. Therefore, all of our experiment is based on the Inflation Growth rate, which is stationary at 1% of confidence.

2.5 Forecasts to be combined and *R* packages

The experiment was developed by application of all cited models and methods to the Inflation Rate series. All of the tested combination methods aim to select the best ensemble of models from a set of possible choices. Specifications for each model are selected using algorithms from *forecast R* package:

- SARIMA
- Exponential Smoothing (Ets)
- Artificial Neural Network (ANN)
- ARFIMA
- Spline Regression (Spline)

All models generated are univariate, based on using lags of Inflation Rate as information Set and no other variables.

The *forecast* package is described in detail in Hyndman and Khandakar (2008). For more details regarding the aforementioned methodologies, see Hamilton (1994) and James et al. (2013). This package chooses a particular specification based on the information available. Model performance may vary throughout the sample.

3 Findings

3.1 Pseudo real time experiment

The data gathered for the countries is used to create many variants of models to forecast Inflation Growth rate. The sample is split into three parts. The first part of the sample is used to estimate the individual models, the second is used to train and combine those estimates while the third is used to evaluate the forecast performance of the various methods over various horizons. In our exercise, we attempt to simulate a real-time operation. We use an information set that reflects, as closely as possible, the one available to agents at the time of the forecast.

For each model, forecasts are generated for one, two, three, ten, eleven and twelve steps ahead. Therefore, our initial training set includes the first 228 observations. The values we use to run our projections are not the same as those that were available to agents at that time. We run projections in our pseudo

real-time experiment with a slightly better information set. This may result in better forecasting accuracy compared to the projections generated in real-time.

To assess the predictive performance of the proposed models, a comparison of mean absolute error (*MAE*) and mean squared error (*MSQE*) generated for each method is made. Table 1 and 2 present a summary of the best models in terms of *MAE* and *MSQE* from one up to twelve steps ahead forecasts. All tables show the first and second model in terms of forecasting performance.

Table 1. Models with the lowest mean absolute error for 1, 2, 3, 10, 11 and 12 steps ahead forecast

	Brazil	Chile	Peru	Mexico
1 Step Ahead	1° *Elm* (0.088)	1° *Lasso*	1° *Lasso*	1° *Elm* (0.536)
	2° *Lasso*	2° *Ridge*	2° *Ridge*	2° *Ridge*
2 Steps Ahead	1° *Elm* (0.100)	1° *Ridge*	1° *Ridge*	1° *Lasso*
	2° *Gb*	2° *Lasso*	2° *Elm* (0.507)	2° *Elm* (0.110)
3 Steps Ahead	1° *Elm* (0.495)	1° *Ridge*	1° *Elm* (0.044)	1° *Lasso*
	2° *Gb*	2° *Elm* (0.659)	2° *Ridge*	2° *Ridge*
10 Steps Ahead	1° *Gb*	1° *Elm* (0.362)	1° *Elm* (0.014)	1° *Elm* (0.008)
	2° *Elm* (0.001)	2° *Ridge*	2° *Gb*	2° *Lasso*
11 Steps Ahead	1° *Gb*	1° *Elm* (0.465)	1° *Elm* (0.002)	1° *Elm* (0.013)
	2° *Elm* (0.018)	2° *Ridge*	2° *Gb*	2° *Lasso*
12 Steps Ahead	1° *Gb*	1° *Elm* (0.140)	1° *Elm* (0.000)	1° *Elm* (0.020)
	2° *Elm* (0.045)	2° *Gb*	2° *Gb*	2° *Lasso*

Diebold Mariano test's p-values with null hypothesis that other model has statistical dominance over *Elm* method.

It's possible to infer that *Elm* mechanisms show a good performance in comparison to the others, achieving, approximately, 60% of best results in terms of *MAE* and *MSQE* in all experiments. In addition, even in the cases where *Elm* method was not the best model, it had a good performance, being one out of two best models in almost all cases. The tables with detailed results are shown in the Appendix section.

To compute the statistic significance of these results, we use Diebold and Mariano (2002) method. We applied the Diebold-Mariano method to pair of all models tested against our *Elm* strategy. We intend to analyze if a model has statistically lesser *MAE* or/and *MSQE* than the other. Detailed results are available by request to the author.

To compare the forecast accuracy of two different methods we are using the alternative version of the Diebold-Mariano test as proposed in Harvey et al. (1997). We test the alternative hypothesis that a second method is less accurate than *Elm* strategy. For illustration purposes, all calculated p-values associated with pair of best models present in table 1 and table 2 are written in parenthesis right after *Elm* description.

Exercise performed to Brazilian data showed outstanding results for Extreme Machine Learning combination. Based on one up to twelve ahead forecast, *Elm*

Table 2. Models with the lowest mean squared error for 1, 2, 3, 10, 11 and 12 steps ahead forecast

	Brazil	Chile	Peru	Mexico
1 Step Ahead	1° *Elm* (0.088)	1° *Lasso*	1° *Lasso*	1° *Elm* (0.536)
	2° *Lasso*	2° *Ridge*	2° *Ridge*	2° *Ridge*
2 Steps Ahead	1° *Elm* (0.072)	1° *Ridge*	1° *Ridge*	1° *Lasso*
	2° *Lasso*	2° *Gb*	2° *Elm* (0.507)	2° *Elm* (0.110)
3 Steps Ahead	1° *Elm* (0.495)	1° *Ridge*	1° *Elm* (0.044)	1° *Lasso*
	2° *Gb*	2° *Gb*	2° *Ridge*	2° *Ridge*
10 Steps Ahead	1° *Elm* (0.001)	1° *Elm* (0.362)	1° *Elm* (0.014)	1° *Elm* (0.008)
	2° *Gb*	2° *Ridge*	2° *Gb*	2° *Lasso*
11 Steps Ahead	1° *Elm* (0.018)	1° *Elm* (0.462)	1° *Elm* (0.002)	1° *Elm* (0.013)
	2° *Gb*	2° *Ridge*	2° *Gb*	2° *Lasso*
12 Steps Ahead	1° *Elm* (0.045)	1° *Elm* (0.435)	1° *Elm* (0.000)	1° *Elm* (0.020)
	2° *Gb*	2° *Ridge*	2° *Gb*	2° *Lasso*

Diebold Mariano test's p-values with null hypothesis that other model has statistical dominance over *Elm* method.

strategy does not only statistically outperformed *Gb* model with three steps ahead forecast. This performance can be seen on figure 2 in our appendix.

The forecast performance of *Elm* combination on Chilean data was the worst performance for *Elm* technique. We can only reject the null that *Gb* performs better than *Elm* at 15% for 12 step ahead forecast. However, *Elm* outperformed *AverageMean* for all steps from 10 to twelve at 7%, 12% and 14%, respectively. The Mean Absolute Error and Mean Squared Error dynamic can be seen on figure 3 in the appendix section.

Our results on Mexican data show that *Elm* method outperformed all models for two, three, ten, eleven and twelve steps ahead forecast with statistical dominance dictated by Harvey et al. (1997) test's results. However, the model did not show satisfactory performance on the short run, as one step ahead forecast.

In the Peruvian case, *Elm* presented a very similar result to Mexican case. Our model statistically outperformed all models, for three, ten, eleven and twelve steps ahead forecast. Detailed results are in Table. Those results for Mexican and Peruvian case can be inferred by graphical analysis of figure 4 and figure 5, respectively.

It is also possible to evaluate the adjustment during training and testing period. All graphical adjustment analysis is described in the Appendix. This is very useful to evaluate the dynamics of the method, however, for this exercise, we opt to show only one and twelve steps ahead forecasts, focusing on short and long run.

4 Discussion

4.1 Algorithm Performance

One of the main contributions of Huang et al. (2004) is that Extreme Learning Machine allows usage of artificial neural network architecture, but with a faster computation analysis.

Our work results follow the same pattern. In our tests, *Elm* showed satisfactory performance in term of computation time. All experiments took less than three seconds to be compute. This result shows Extreme Learning Machine as a fast algorithm without loosing in terms of performance. It is worth pointing out that all of our proposed *Elm* architectures statistically surpassed *Mlp* combination approach.

5 Concluding Remarks

The inflation rate is one of the most important economic indicators to be forecasted. Due to this fact, inflation rate forecasting was and still is one of the most discussed topics on time series forecasting.

Our work analyzed a new forecast combination framework based on Extreme Learning Machine framework proposed by Huang et al. (2004) to forecast the inflation rate. We tested our model against some recent proposed combinations methods and classical benchmarks. To perform this exercise, we used a time series of Price Index Growth Rate over 4 Latin American countries: Brazil, Chile, Mexico, and Peru.

It's possible to infer that *Elm* mechanisms show a good performance in comparison to the others, achieving, approximately, 60% of best results in terms of *MAE* and *MSQE* in all experiments. In addition, even in the cases where *Elm* method was not the best model, it had a good performance, being one out of two best models in almost all cases. All those results were statistically validated by the use of Harvey et al. (1997) method to test the null if the compared benchmark has better performance than *Elm*.

It is also important to point out the algorithm speed, one of the main advantages of *Elm* over other Artificial Neural Networks architectures. All of our experiments were performed on less than 10 seconds.

Indeed our results proved that Extreme Learning Machine combinations methods have great potential, which raises the research question of what kind of different architectures could be applied to this model in ways of obtaining even better performance. In this sense, future work on this Machine Learning technique can improve our actual forecasting combination methods.

It's worth noting that our conclusions are not a general theory, with results only applied to the cases analyzed here. In this sense, it's possible to overcome some of this work's limitations by extending the analysis to more and different countries, as using different benchmarks.

Bibliography

Bates, J. M. and Granger, C. W. (1969). The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468.

Behbahani, H., Amiri, A. M., Imaninasab, R., and Alizamir, M. (2018). Forecasting accident frequency of an urban road network: A comparison of four artificial neural network techniques. *Journal of Forecasting*, 37(7):767–780.

Castle, J. L., Clements, M. P., and Hendry, D. F. (2013). Forecasting by factors, by variables, by both or neither? *Journal of Econometrics*, 177(2):305–319.

Chan, F. and Pauwels, L. L. (2018). Some theoretical results on forecast combinations. *International Journal of Forecasting*, 34(1):64–74.

Clements, M. P. and Hendry, D. F. (2001). *Forecasting non-stationary economic time series*. mit Press.

Deutsch, M., Granger, C. W., and Teräsvirta, T. (1994). The combination of forecasts using changing weights. *International Journal of Forecasting*, 10(1):47–57.

Diebold, F. X. and Mariano, R. S. (2002). Comparing predictive accuracy. *Journal of Business & economic statistics*, 20(1):134–144.

Diebold, F. X. and Shin, M. (2018). Machine learning for regularized survey forecast combination: Partially-egalitarian lasso and its derivatives. *International Journal of Forecasting*.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics New York.

Garcia, M. G., Medeiros, M. C., and Vasconcelos, G. F. (2017). Real-time inflation forecasting with high-dimensional models: The case of brazil. *International Journal of Forecasting*, 33(3):679–693.

Hamilton, J. D. (1994). Time series analysis.

Hansen, P. R., Lunde, A., and Nason, J. M. (2011). The model confidence set. *Econometrica*, 79(2):453–497.

Harvey, D., Leybourne, S., and Newbold, P. (1997). Testing the equality of prediction mean squared errors. *International Journal of forecasting*, 13(2):281–291.

Hsiao, C. and Wan, S. K. (2014). Is there an optimal forecast combination? *Journal of Econometrics*, 178:294–309.

Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., et al. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural networks*, 2:985–990.

Hyndman, R. and Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software, Articles*, 27(3):1–22.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Meese, R. A. and Rogoff, K. (1983). Do they fit out of sample? *Journal of international economics*, 14:3–24.

Rossi, B. (2013). Exchange rate predictability. *Journal of economic literature*, 51(4):1063–1119.

Stock, J. H. and Watson, M. W. (1999). Forecasting inflation. *Journal of Monetary Economics*, 44(2):293–335.

Stock, J. H. and Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business and Economic Statistics*, 20(2):147–162.

Tallman, E. W. and Zaman, S. (2017). Forecasting inflation: Phillips curve effects on services price measures. *International Journal of Forecasting*, 33(2):442–457.

Timmermann, A. (2006). Forecast combinations. *Handbook of economic forecasting*, 1:135–196.

Wang, J., Athanasopoulos, G., Hyndman, R. J., and Wang, S. (2018a). Crude oil price forecasting based on internet concern using an extreme learning machine. *International Journal of Forecasting*, 34(4):665–677.

Wang, L., Wang, Z., Qu, H., and Liu, S. (2018b). Optimal forecast combination based on neural networks for time series forecasting. *Applied Soft Computing*, 66:1–17.

Zhang, B. (2019). Real-time inflation forecast combination for time-varying coefficient models. *Journal of Forecasting*, 38(3):175–191.

A Supplementary Material

Fig. 2. Mean Absolute Error and Mean Squared Error for Brazilian forecasting exercise

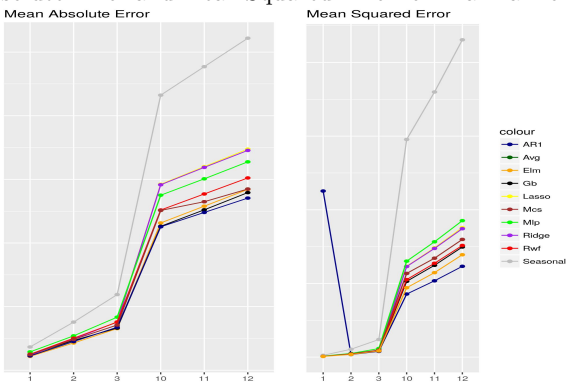


Fig. 3. Mean Absolute Error and Mean Squared Error for Chilean forecasting exercise

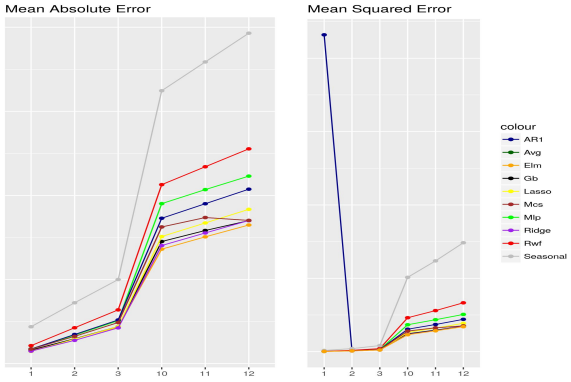


Fig. 4. Mean Absolute Error and Mean Squared Error for Mexican forecasting exercise

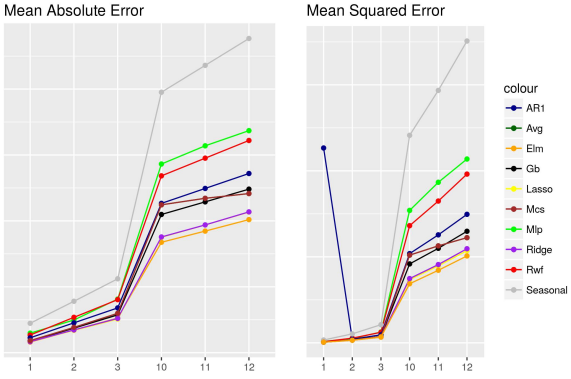


Fig. 5. Mean Absolute Error and Mean Squared Error for Peruvian forecasting exercise

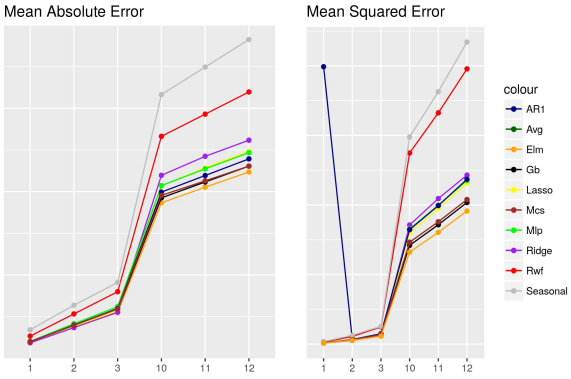


Fig. 6. Statistical adjustment of *Elm* method for Brazil, from one up to twelve steps ahead forecasts.

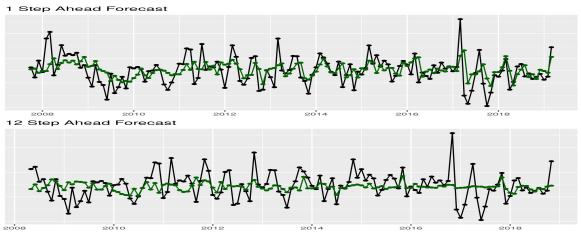


Fig. 7. Statistical adjustment of *Elm* method for Chile, from one up to twelve steps ahead forecasts.

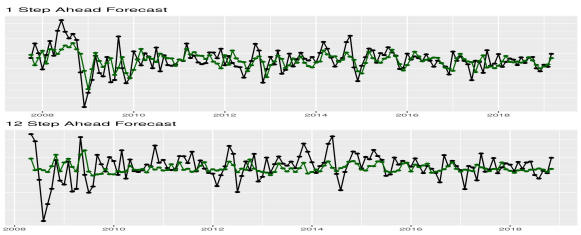


Fig. 8. Statistical adjustment of *Elm* method for Mexico, from one up to twelve steps ahead forecasts.

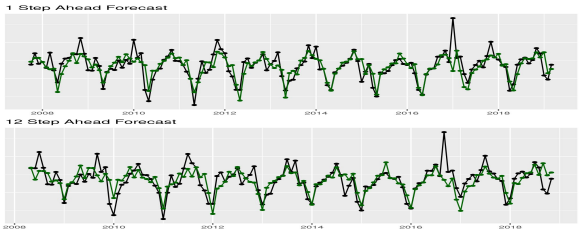
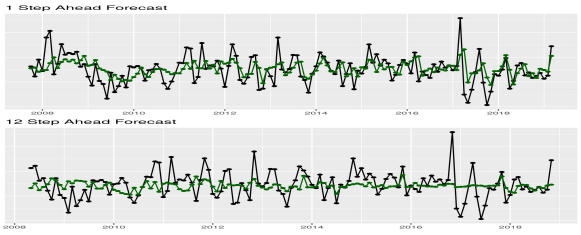


Fig. 9. Statistical adjustment of *Elm* method for Peru, from one up to twelve steps ahead forecasts.



Dynamic behavior in the fractional scope of agricultural commodities price series vis-a-vis ethanol prices

C. M. C. Inacio Jr.[a,*], S.A. David[a]

[a]*Department of Biosystems Engineering, University of Sao Paulo, 13635-900 Pirassununga, SP, Brazil - sergiodavid@usp.br*

Abstract

Having in mind that the share of ethanol as biofuel has increased in recent decades, concern about the impacts of its prices on agricultural commodity prices has gained relevance. These concerns are due to the fact that the energy and environmental benefits derived from the use of biofuels can occur at the expense of the impact of agricultural commodity prices. This is because most of the raw materials currently used to produce biofuels, such as corn in the US, sugarcane in Brazil and oilseeds in Europe, are also important commodities globally. When it comes to ethanol, it could not only influence the price level of these agricultural commodities but may also affect the volatility of these prices. Price volatility reflects the volatility of current and expected future values of production, consumption, and inventory demand. With the recognition that there are other measures of volatility associated with consumption, production or stocks, in this study, the focus is on price series and the main objective is to use numerical modelling and numerical simulation tools in the fractional scope in order to analyze the relationship between the behavior of ethanol price dynamics and the price dynamics of some agricultural commodities, such as corn, sugar and soybean. It is hoped that the results may contribute to the clarification of price relationships between ethanol and such commodities, thus allowing agents in these markets to be clearer in making decisions that involve hedging, risk exposure and investment incentives.

Keywords: Ethanol, time series, multivariate analysis, cointegration, VECM

1. Introduction

Brazil is a reference in biofuels production, with particular emphasises in the ethanol commodity. The Brazilian ethanol's production began growing in the 1970's when the oil prices decreased in the London and New York stock exchanges. Following this trend, the USA advanced in the ethanol production from corn in the mid of 2005 [1].

After the biggest discoveries in the history of Brazilian fuel market, namely the pre-salt layer located in the Santos basin, a great considerable investments in the oil sector start occured resulting in a cutting off on the ethanol investment.

One of the strategies of Brazil's sugarcane crop season for 2011-2012 was the commercial approach to the USA. The deal included at exportation of Brazil's ethanol from sugarcane to the USA that guaranteed a premium for this biofuel [2]. On the other side of the trading, Brazil imported corn-based ethanol from the USA, due to a sugarcane crop shortfall that resulted in an increase of the biofuel prices for consumers.

*Corresponding author: Tel: +55 19 35656711
E-mail addresses:sergiodavid@usp.br (S.A. David)

A strategy for recovering the ethanol production in the 2011-2012 period included the reduction of taxes in the ethanol sector and to increase the amount of gasoline in the blend with ethanol, namely from 20% to 25%. [3]. Simultaneously, the government offered to sugarcane producers subsidized loans with interest free in order to recover the earlier results. In fact, this strategy brought results to the sector, since in the following crop season of 2014 the ethanol production reached a record of 28.6 billion liters [4]. However, this also reflected by lowering the prices of the sugar in the international market, that is usually the alternative option when producting ethanol from plants.

In the recent years, the Brazilian energy sector is found to be slowing recovering, and an optimistic scenario emerged, where the internal prices are again linked to the international prices. Besides, a new investments cycle was recently announced for the sector until 2030, supported by the Renovabio program [5].

Brazil is also a major player when it comes to agricultural commodities [6, 7]. The country is recognized as one of the major exporters of agricultural products, primarily as result to its strong performance in the sector. According to the Food and Agriculture Organization of the United Nations (FAO), Brazilian agriculture products contributed to about 4% of the country's gross domestic product (GDP) and, one can see notables exportables products directly influencing this margin, such as: the sugarcane and its derivatives (ethanol and sugar), soybean, and corn (among others). It is important to hightlight that the country's agricultural area is increasing each year, implying the requirement for agricultural machinery and equipments that lead to significant impact on the use of energy, such as the ethanol.

Several studies applied the concepts of error correction models (ECM) and cointegration in agricultural and energy commodities [8, 9, 10, 11, 12, 13, 14]. Baffes et al, [10] investigated the price transmission from the international market to the internal market for 9 agricultural commodities (cocoa, coffee, corn, rice, soybean, sorghum, sugar, palm oil and wheat) over 8 countries (Argentina, Chile, Colombia, Egypt, Ghana, Indonesia, Madagascar and Mexico). Similarly, Balcombe et al [11] studied the link among corn, wheat and soybean in Brazil, Argentina and USA during the 80's and 90's. Mattos et al studied the price transmission using VECM for future markets of agricultural commodities and the ethanol commodity [12]. They evaluated the impact on the price transmission of futures prices of the Chicago Mercantile Exchange (CME) and the Brazilian Stock Exchange (BMF) in spot prices of the corn in the Brazilian internal market. Mallory et al, [13] explored the topic and analyzed long-term relations between the ethanol, corn and natural gas in the USA.

Hereafter we analyze the price transmission between the Brazilian ethanol price series and other important agricultural commodities, such as, sugar, corn and soybean. For this purpose, we adopt several mathematical tools, namely, the Bai-Perron test of breakpoints, the Cointegration test of Johansen and, the Vector Error Correction Model (VECM) exploited by the Orthogonal Impulse Response (OIR) and the the Forecast Error Variance Decomposition (FEVD).

This paper is organized as follows. In section 2, the time series (TS) are presented followed by the adopted methods presentation. In section 3, the results are discussed. Finally, in section 4, the main conclusions are outlined.

2. Data and Methodology

We consider the spot prices of the Brazilian hydrous ethanol and several other important commodities in the Brazilian agricultural GDP, such as the sugar, corn and soybean. The work aims to measure the impact of ethanol prices against the agricultural commodities and vice versa. Such evaluation is possible using multivariate models that are described in the follow-up of this paper.

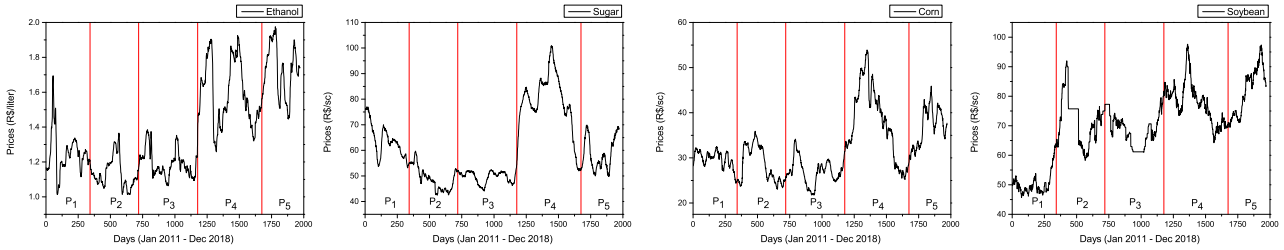
The data are obtained from the Center for Advanced Studies on Applied Economics/University of Sao Paulo (CEPEA/USP). CEPEA methodology for daily pricing these products can be found in its website (`www.cepea.esalq.usp.br`). We adopted the daily spot prices both for the ethanol and the agricultural commodities, and the time interval goes from January/2011 to December/2018.

Bearing in mind that during this period many changes occured in Brazilian energy sector, we evaluate the presence of breakpoints (BP) in the prices of the ethanol TS by means of the Bai-Perron algorithm [15]. The main idea consists in obtaining the optimal number of breaks in the TS using an information criterion, namely the Bayesian Information Criterion (BIC). The Bai-Perron algorithm is a dynamic method that estimates multiple structural changes (i.e. breakpoints) as global minimizers of the sum of squared residuals in a given TS [15]. As stated in [15], the Schwarz criterion, or simply the BIC, was applied for structural break inference by Yao [16]. The BIC value is defined as $BIC = -2LL + klog(n)$, where LL is the log-likelihood of the model, k is the number of independent parameters and n is the number of observations in the TS. Thus, the criterion is the statistics that maximizes the probability of identifying the best fitted model to the TS. Then, the model with the lowest BIC value is chosen as the best model [17]. This is commonly applied for selecting the dimension of the model by estimating the number of the breaks. Therefore, for the ethanol TS breakpoints, we obtained five possible options of structural breaks from A-E, where A is the TS with one BP and E is the maximum amount, that is, five BP in the TS. The values are listed in Table 1:

Option	BIC	Breakpoints observation number				
A	-2145.91	1177				
B	-2316.48	1176	1675			
C	-2396.85	322	1176	1675		
D	-2423.16	342	719	1177	1675	
E	-2349.23	295	590	885	1180	1675

Table 1: BIC criterion, Bai-Perron test and the corresponding breakpoints options (A-E) in the daily ethanol TS based on CEPEA methodology.

From Table 1 we verify that 4 is the optimal number of breaks in the TS. The BIC shows a slightly lower value (with confidence interval of 97,5%), which represents a better fitting for the 5 sub-periods considered in the multivariated analysis. Therefore, the sub-periods intervals are as it follows: i) Sub-period 1 (P_1) - from January/2011 to May/2012, ii) Sub-period 2 (P_2) - from May/2012 to November/2013, iii) Sub-period 3 (P_3) - from November/2013 to September/2015, iv) Sub-period 4 (P_4) - from September/2015 to October/2017 and v) Sub-period 5 (P_5) - from October/2017 to December/2018. The applied TS in this work and the corresponding sub-periods are illustrated in the Figure 1.



(a) Ethanol TS (b) Sugar TS (c) Corn TS (d) Soybean TS

Figure 1: Spot prices series of the commodities analyzed in the study and the corresponding sub-periods.

2.1. Cointegration of Time Series

The cointegration relation between two TS was firstly introduced by Granger [18]. Later, Engle and Granger [19] explored the criation of an error correction model. One simple way for explaning a cointegration relation was purposed by Murray [20] entitled "The metaphor of the Drunk and Her Dog". In order to investigate the price transmission process among the ethanol and others agricultural commodities, the cointegration hypothesis is considered. The process of adjustment is pointed by Murray [20] as the error correction model. Considering the error correction terms c and d one can write

$$x_t - x_{t-1} = u_t + c(y_{t-1} - x_{t-1}), \tag{1}$$

$$y_t - y_{t-1} = w_t + d(x_{t-1} - y_{t-1}), \tag{2}$$

where x_t and y_t are the cointegrated variables, u_t and w_t are the white noise stationary steps and, $(x_{t-1} - y_{t-1})$ is the cointegration relation between the variables x and y.

In this work, we evaluate the cointegration process among some agricultural commodities prices, namely the sugar, corn, soybean with respect to the ethanol prices. The cointegration is calculated using the Johansen test [21], and the VECM model is estimated when the cointegration between a particular commodity with respect to the ethanol is verified. Both are explained in the follow-up.

2.1.1. The Johansen test for cointegration

The Johansen test [22, 23] is applied to verify if the rank (r) of the matrix $\alpha\beta$ is equal to zero (null hypothesis). If it is not zero, then the cointegration exists. If $r = 0$ this implicate a the non-existence of the error correction term (ECT). Otherwise, in the case of r $\neq 0$, the null hypothesis is rejected and there is a cointegration relation between the analysed TS. Johansen proposed two possibles tests, namely, the Max-Eigen and the Trace tests that are based in the assumption of pure unit root. In contrast to the method for cointegration validation of Engle-Granger [19], the test purposed by Johansen allows the study of more than one cointegration relation among the variables. For this reason, the Johansen test is applied in this work with a view of analyse possibles cointegration processes along the prices of agricultural commodities and the prices of ethanol.

2.2. The Vector Error Correction Model (VECM)

The error correction model (ECM) was introduced by Sargan [24] and later imployed by Davidson [25], evolving toward what is known today as the VECM methodology. The VECM is based on the generalized vector autore-

gression (VAR), that allows an adjustment of a regression model between multiple variables in order to evaluate its relation.

Let us consider p_1 and p_2 as non-cointegrated and non-stationary TS. Then, the approach from the VAR(j) model in differences is possible as [26, 27]

$$\Delta p_t = log(p_{1t}) - log(p_{1t-1}) \tag{3}$$

$$\Delta p_{1t} = \gamma_0 + \sum_{i=1}^{j}[\gamma_1(i)\Delta p_{1t-i}] + \sum_{i=1}^{j}[\gamma_2(i)\Delta p_{2t-i}] + \epsilon_{1t}, \tag{4}$$

$$\Delta p_{2t} = \theta_0 + \sum_{i=1}^{j}[\theta_1(i)\Delta p_{1t-i}] + \sum_{i=1}^{j}[\theta_2(i)\Delta p_{2t-i}] + \epsilon_{2t}. \tag{5}$$

where γ and θ are the equation autoregressive terms of p_1 and p_2, respectively, ε_{1t} and ε_{2t} are white-noise disturbances and $i = 1, 2, ..., j$. Since a cointegration process is found among two or more TS, then the ECT can be implemented in the VAR model, which assumes the name of VECM. Note that for implementing the VECM, it is not needed that both TS are stationary. Indeed, once the β values are calculated from the ECT modeling they are adjusted for which a stationary ECT is returned and then applied in the regression equation. Thus, the ECT is represented in the VECM as $\alpha_i(\beta_0+\beta_1 p_{1t-1}+\beta_2 p_{2t-1})$ for each price equation, where $\beta_0+\beta_1 p_{1t-1}+\beta_2 p_{2t-1} = 0$ represents the equilibrium equation between the prices. Therefore, a VECM is mathematically expressed as [12, 28, 29]

$$\Delta p_{1t} = \gamma_0 + \alpha_1(\beta_0 + \beta_1 p_{1t-1} + \beta_2 p_{2t-1}) + \sum_{i=1}^{j}[\gamma_1(i)\Delta p_{1t-i}] + \sum_{i=1}^{j}[\gamma_2(i)\Delta p_{2t-i}] + \epsilon_{1t}, \tag{6}$$

$$\Delta p_{2t} = \theta_0 + \alpha_2(\beta_0 + \beta_1 p_{1t-1} + \beta_2 p_{2t-1}) + \sum_{i=1}^{j}[\theta_1(i)\Delta p_{1t-i}] + \sum_{i=1}^{j}[\theta_2(i)\Delta p_{2t-i}] + \epsilon_{2t}. \tag{7}$$

3. Results and discussion

In this section, results are presented and discussed for the cointegration test and, subsequently, the VECM model estimated for the cointegrated pairs by means of the VECM equation, OIR and FEVD. Subsection 3.1 analyses the results of the Johansen test in the different sub-periods. Subsection 3.2 discusses the VECM results adjusted for the sub-periods where the cointegration process, that is, a price transmission was found between the ethanol over the agricultural commodities and vice versa.

3.1. Cointegration from Johansen test

The Johansen test is applied based on the Max-Eigen and Trace tests as mentioned in Subsection 2.1.1. The cointegration process is evaluated for the full period (2011-2018) and for the sub-periods considered by means of the Bai-Perron test for breakpoints. In the follow-up, the results are divided into the analysed periods and the cointegration relations are discussed based on the energy and agricutural market historical of the time.

3.1.1. Full period (2011-2018)

The test results for the full period are listed in Table A.4a (see Appendix A). From Table A.4a we observe the cointegration process between the sugar with the ethanol, described by the rejection of the null hypothesis of the Johansen test. Such price transmission is expected from ethanol and sugar [9], since the markets are directly related and the sugar becomes an option against the ethanol production. For example, when the currency ratio between the Brazilian reals (R$) and the American dollar (U$) is high, the ethanol plants tends to choose the sugar production with the primarly intention of exportation, and this increases the ethanol prices in the domestic market.

Likewise, the expected price transmission between the ethanol and corn in the American market, the Brazilian prices of ethanol and corn also show a cointegration for both tests. However, this behavior was not expected, since the Brazilian ethanol is primarly produced from sugarcane, while the corn-based ethanol is in an initial state of production in the country. Agricultural production demands great amount of energy inputs, that is reflected in the fuel consumption. The results suggest that ethanol prices and the markets of corn are linked by some means, despite its production processes not being explicitly linked.

3.1.2. Sub-period 1 (January/2011 - May/2012)

Table A.4b summarizes the cointegration relation between the pairs of ethanol and each of the agricultural commodities over the period from January/2011 to May/2012. It is possible to note that for the first sub-period, cointegrations of ethanol with the sugar is still considered, and possibly influence the results in a macroscopic scale.

During the sub-period 1, Brazil imported a significant volume of ethanol from the USA, due to a crop shortfall and an increase in the sugar international prices. This resulted in an increase in the prices for both commodities during the period.

3.1.3. Sub-period 2 (May/2012 - November/2013)

The cointegration test results are summarized in Table A.4c. No cointegration process is observed for the pairs during the sub-period 2. One of the factor that helps describing this behavior is the currency exchange of the American dollar at this date. The prices of agricultural commodities usually have a negative correlation with the price of the dollar. Therefore, when the dollar gains force, the commodities become more expensive in other currencies, influencing negatively the demand. Alternatively, when the dollar becomes weaker, the commodities prices decrease in others currencies and then, as increasing demand occurs in the countries that import these commodities.

3.1.4. Sub-period 3 (Nov/2013 - Sept/2015)

From the results, it is possible to note a cointegration between corn and ethanol for the considered sub-period. However, a cointegration between ethanol and soybean was not identified in any of the studied sub-periods and could be explained by the relation between the spot and futures prices of the soyean in Brazil being influenced by the international price negotiated in the CME.

3.1.5. Sub-period 4 (Sept/2015 - Oct/2017)

The sub-period 4 results for cointegration test are hilighted in Table A.4e (see Appendix A). Likewise the sub-period 2, the sub-period 4 also demonstrated no cointegration processes among the considered pairs. As pointed

before, the interval is related to a political instability period that affected directly the energy sector of the country.

3.1.6. Sub-period 5 (Oct/2017 - Dec/2018)

Likewise previous sub-period, the results of the sub-period 5 points out to no price transmission relations in the ethanol and the agricultural commodities.

3.2. VECM Results

In this section, VECM results are hightlighted for the sub-periods and the pairs that showed a cointegration process in the Johansen test (see in Section 3.1), i.e. sub-period 1, sub-period 3 and sub-period 5. It is worth to mention that a plot scale adjustment to the FEVD model was applied in order to obtain a better visualization of the influences.

3.2.1. Sub-period 1

This section covers the VECM results of ethanol-sugar pair.

Ethanol vs Sugar.

Eq. 9 and 10 describe the ethanol price equation in the returns and the sugar price equation in the returns, respectively.

$$\begin{aligned}\Delta P_t^{ethanol} = -0.01581(-0.85386 + 1.0P_{t-1}^{ethanol} - 0.00612P_{t-1}^{sugar}) \\ +0.75274\Delta P_{t-1}^{ethanol} + 0.00017\Delta P_{t-1}^{sugar}\end{aligned} \tag{8}$$

$$\begin{aligned}\Delta P_t^{sugar} = -0.10777(-0.85385 + 1.0P_{t-1}^{ethanol} - 0.00612P_{t-1}^{sugar}) \\ +0.227705\Delta P_{t-1}^{ethanol} + 0.26472\Delta P_{t-1}^{sugar}\end{aligned} \tag{9}$$

Despite the $\alpha_{sugar} = -0.10777$ being higher than $\alpha_{ethanol} = -0.01581$, one can note from the Figure 2 that the adjustment coefficient ($\alpha_{ethanol}$) implies in bigger adjustments of the ethanol in a long-run disequilibrium. Besides, the commodity tends to reach an equilibrium in $1/\alpha_{ethanol}$ steps, i.e., $1/|-0.01581| \approx 63$ steps. Therefore, the equation that allows the analysis of the long-run equilibrium relation between the ethanol-sugar prices is described as:

$$-0.85386 + p_{t-1}^{ethanol} - 0.00612 * p_{t-1}^{sugar} = 0. \tag{10}$$

In order to measure the forecast error variance related to the prices shocks of one variable to the other in both ethanol and sugar equation, the FEVD tool is applied. In this case, it is possible to note small residuals from the ethanol prices in the sugar equation (see Figure 2), with a decreasing behavior along the steps. However, in contrast, the sugar commodity residuals are also presented in the ethanol, but increasing along the steps.

3.2.2. Sub-period 3

In the sub-period 3, the ethanol-corn pair is considered as result of the cointegration test applied for the period.

Ethanol vs Corn.

The equations of the prices of ethanol-corn pair are presented in Eq. 11 and 12.

$$\begin{aligned}\Delta P_t^{ethanol} = &-0.02512(-0.66335 + 1.0P_{t-1}^{ethanol} - 0.01953P_{t-1}^{corn}) \\ &+0.28729\Delta P_{t-1}^{ethanol} + 0.30347\Delta P_{t-2}^{ethanol} + 0.00678\Delta P_{t-3}^{ethanol} \\ &-0.00174\Delta P_{t-1}^{corn} + 0.00126\Delta P_{t-2}^{corn} + 0.00479\Delta P_{t-3}^{corn}\end{aligned} \tag{11}$$

$$\begin{aligned}\Delta P_t^{corn} = &\ 0.35541(-0.66335 + 1.0P_{t-1}^{ethanol} - 0.01953P_{t-1}^{corn}) \\ &+2.72427\Delta P_{t-1}^{ethanol} - 1.07728\Delta P_{t-2}^{ethanol} + 0.19542\Delta P_{t-3}^{ethanol} \\ &+0.12564\Delta P_{t-1}^{corn} + 0.22177\Delta P_{t-2}^{corn} + 0.22966\Delta P_{t-3}^{corn}\end{aligned} \tag{12}$$

It is possible to note from Figure 3 a mutual relation between the pair prices in which both ethanol influences in the corn prices and vice versa, which leads us to assume that imbalances in the both commodities prices influences in one another. The intensity level of the corn prices imbalance is higher than the contrary situation. From the autoregressive terms of the prices equations, we can note that ethanol prices of three days in the past influences the corn prices and vice versa. Besides, FEVD suggests mutual residual impact pattern in intensity level over the steps. In order to exemplify the long-run relation between the pair, the following equation is presented:

$$-0.66335 + p_{t-1}^{ethanol} - 0.01953 * p_{t-1}^{corn} = 0. \tag{13}$$

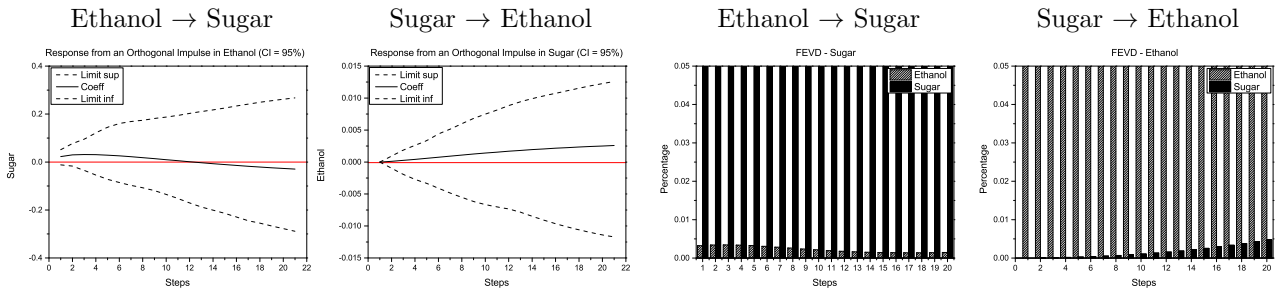


Figure 2: OIR and FEVD for the ethanol-sugar pair in sub-period 1.

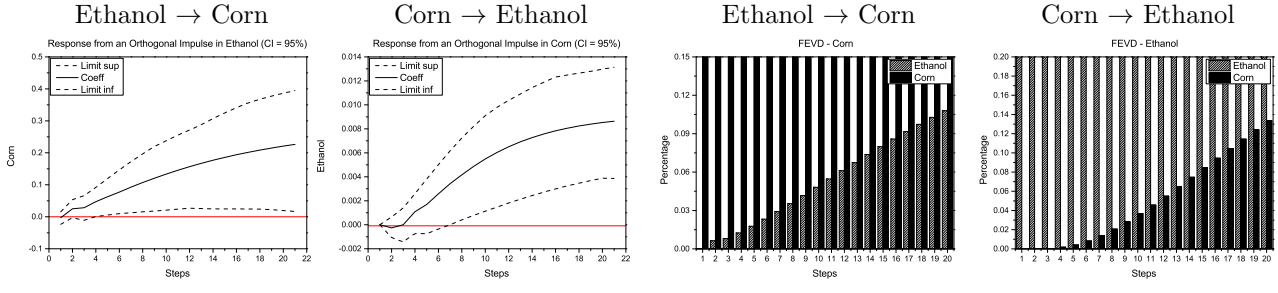


Figure 3: OIR and FEVD for the ethanol-corn pair in sub-period 3.

4. Conclusions

This study employed the multivariate analysis to investigate the ethanol prices transmission to the main Brazilian's agricultural commodities (and vice versa) by means of cointegration tests. Afterwards, VECM estimation is evaluated whenever cointegration process between ethanol and a particular commodity is verified.

The obtained results suggest a mutual price transmission from the ethanol commodity to the agricultural commodities evidenced by the ethanol-sugar pair (sub-period 1) and the ethanol-corn pair (sub-period 3), where imbalances are mutually transmitted, rather than the dominance of one to the other. The achieved results can be explained by the fact that agricultural commodities consumes energy inputs throughout its production stages. Therefore, important deviations or shocks in the prices of the ethanol can be transmitted significantly to the agricultural commodities analyzed, in the same time that production decisions such as the sugar supply can influence in the ethanol prices.

Acknowledgments

The authors wish to acknowledge the FAPESP (Sao Paulo Research Foundation), grants 2017/13815-3 and 2017/15517-0, for funding support.

Appendix A.

Results related to Johansen cointegration test, Orthogonal Impulse Response and Forecast Error Variance Decomposion are grouped into the Appendix A.

Pairs	H_0	Jan/2011 – Dec/2018	
		Max-Eigen	Trace
Ethanol - Sugar	r = 0	13.90*	17.67
	r <= 1	3.76	3.76
Ethanol - Corn	r = 0	14.05*	18.05*
	r <= 1	4.0	4.0
Ethanol - Soybean	r = 0	10.63	15.43
	r <= 1	4.8	4.8

(a) Full-period

Pairs	H_0	Jan/2011 – May/2012	
		Max-Eigen	Trace
Ethanol - Sugar	r = 0	11.52	18.61*
	r <= 1	7.09	7.09
Ethanol - Corn	r = 0	12.02	16.99
	r <= 1	4.97	4.97
Ethanol - Soybean	r = 0	11.57	13.08
	r <= 1	1.51	1.51

(b) Sub-period 1

Pairs	H_0	May/2012 – Nov/2013	
		Max-Eigen	Trace
Etanol - Sugar	r = 0	6.80	10.37
	r <= 1	3.57	3.57
Etanol - Corn	r = 0	7.31	10.77
	r <= 1	3.46	3.46
Etanol - Soybean	r = 0	7.54	11.38
	r <= 1	3.85	3.85

(c) Sub-period 2

Pairs	H_0	Nov/2013 – Sept/2015	
		Max-Eigen	Trace
Ethanol - Sugar	r = 0	13.34	15.51
	r <= 1	2.17	2.17
Ethanol - Corn	r = 0	15.02*	17.92*
	r <= 1	2.89	2.89
Ethanol - Soybean	r = 0	8.70	9.57
	r <= 1	0.87	0.87

(d) Sub-period 3

Pairs	H_0	Sept/2015 – Oct/2017	
		Max-Eigen	Trace
Ethanol - Sugar	r = 0	7.72	8.60
	r <= 1	0.88	0.88
Ethanol - Corn	r = 0	4.88	6.51
	r <= 1	1.63	1.63
Ethanol - Soybean	r = 0	8.94	13.14
	r <= 1	4.20	4.20

(e) Sub-period 4

Pairs	H_0	Oct/2017 – Dec/2018	
		Max-Eigen	Trace
Ethanol - Sugar	r = 0	8.55	11.13
	r <= 1	2.58	2.58
Ethanol - Corn	r = 0	11.03	14.45
	r <= 1	3.42	3.42
Ethanol - Soybean	r = 0	7.41	9.47
	r <= 1	2.06	2.06

(f) Sub-period 5

Figure A.4: Johansen test results. Significance levels: 10%(*), 5%(**) and 1%(***).

References

[1] D. Vedenov, M. Wetzstein, Toward an optimal U.S. ethanol fuel subsidy, Energy Economics 30 (5) (2008) 2073 – 2090 (2008). doi:https://doi.org/10.1016/j.eneco.2007.02.004.

[2] J. Goldemberg, Ethanol for a sustainable energy future, Science 315 (5813) (2007) 808–810 (2007). doi:10.1126/science.1137013.

[3] EPE, Análise de conjuntura dos biocombustíveis - ano 2013, Tech. rep., Empresa de Pesquisa Energética (EPE), Rio de Janeiro - RJ (June 2014).
URL http://epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/

[4] EPE, Análise de conjuntura dos biocombustíveis - ano 2014, Tech. rep., Empresa de Pesquisa Energética (EPE), Rio de Janeiro - RJ (May 2015).
URL http://epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/

[5] E. Farina, L. Rodrigues, A política nacional de biocombustíveis e os ganhos de eficiência no setor produtivo, Tech. rep., FGV ENERGIA, Sao Paulo - SP, Boletim Energético (March 2018).
URL http://bibliotecadigital.fgv.br/ojs/index.php/agroanalysis/article/view/78303

[6] S. A. David, J. A. T. Machado, L. R. Trevisan, C. M. C. Inácio Jr, A. M. Lopes, Dynamics of commodities prices: integer and fractional models, Fundamenta Informaticae (2017). doi:10.3233/FI-2017-1499.

[7] T. Serra, D. Zilberman, J. Gil, Price volatility in ethanol markets, European Review of Agricultural Economics 38 (2) (2010) 259–280 (12 2010). doi:10.1093/erae/jbq046.

[8] D. Quintino, S. David, C. Vian, Analysis of the relationship between ethanol spot and futures prices in Brazil, International Journal of Financial Studies 5 (2) (2017) 11 (Apr 2017). doi:10.3390/ijfs5020011.

[9] L. Kristoufek, K. Janda, D. Zilberman, Comovements of ethanol-related prices: evidence from Brazil and the USA, GCB Bioenergy 8 (2) (2016) 346–356 (2016). doi:10.1111/gcbb.12260.

[10] J. Baffes, B. Gardner, The transmission of world commodity prices to domestic markets under policy reforms in developing countries, The Journal of Policy Reform 6 (3) (2003) 159–180 (2003). doi:10.1080/0951274032000175770.

[11] K. Balcombe, A. Bailey, J. Brooks, Threshold Effects in Price Transmission: The Case of Brazilian Wheat, Maize, and Soya Prices, American Journal of Agricultural Economics 89 (2) (2007) 308–323 (05 2007). doi:10.1111/j.1467-8276.2007.01013.x.

[12] F. L. Mattos, R. L. Franco da Silveira, The expansion of the Brazilian winter corn crop and its impact on price transmission, International Journal of Financial Studies 6 (2018). doi:10.3390/ijfs6020045.

[13] D. J. Mallory, M. L.; Hayes, S. A. Irwin, How market efficiency and the theory of storage link corn and ethanol markets, Energy Economics 34 (6) (2012) 2157–2166 (2012).

[14] David, Inácio, T. Machado, Ethanol prices and agricultural commodities: An investigation of their relationship, Mathematics 7 (9) (2019) 774 (Aug 2019). doi:10.3390/math7090774.

[15] J. Bai, P. Perron, Computation and analysis of multiple structural change models, Journal of Applied Econometrics 18 (1) (2003) 1–22 (2003). doi:10.1002/jae.659.

[16] Y.-C. Yao, Estimating the number of change-points via schwarz' criterion, Statistics Probability Letters 6 (3) (1988) 181 – 189 (1988). doi:https://doi.org/10.1016/0167-7152(88)90118-6.

[17] F. A. Kingdom, N. Prins, Chapter 9 - model comparisons, in: F. A. Kingdom, N. Prins (Eds.), Psychophysics (Second Edition), second edition Edition, Academic Press, San Diego, 2016, pp. 247 – 307 (2016). doi:https://doi.org/10.1016/B978-0-12-407156-8.00009-8.

[18] C. W. J. Granger, Some properties of time series data and their use in econometric model specification, Journal of Econometrics 16 (1981) 121–130 (1981).

[19] C. W. J. Engle, R. F.; Granger, Cointegration and error correction: Representation, estimation and testing, Econometrica 55 (1987) 251–276 (1987).

[20] M. P. Murray, A drunk and her dog: An illustration of cointegration and error correction, The American Statistician 48 (1) (1994) 37–39 (1994). doi:10.1080/00031305.1994.10476017.

[21] S. Johansen, Statistical analysis of cointegration vectors, Journal of Economic Dynamics and Control 12 (2) (1988) 231 – 254 (1988). doi:https://doi.org/10.1016/0165-1889(88)90041-3.

[22] S. Johansen, Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models, Econometrica 59 (6) (1991) 1551–1580 (1991).

[23] J. Breitung, U. Hassler, Inference on the cointegration rank in fractionally integrated processes, Journal of Econometrics 110 (2) (2002) 167 – 185, long memory and nonlinear time series (2002). doi:https://doi.org/10.1016/S0304-4076(02)00091-X.

[24] J. D. Sargan, Wages and prices in the United Kingdom: A study in econometric methodology, Econometric Analysis for National Economic Planning 16 (1964) 25–54 (1964).

[25] J. E. H. Davidson, D. F. Hendry, F. Srba, S. Yeo, Econometric Modelling of the Aggregate Time-Series Relationship Between Consumers' Expenditure and Income in the United Kingdom, The Economic Journal 88 (352) (1978) 661–692 (12 1978). doi:10.2307/2231972.

[26] A. Cologni, M. Manera, Oil prices, inflation and interest rates in a structural cointegrated VAR model for the G-7 countries, Energy Economics 30 (3) (2008) 856 – 888 (2008). doi:https://doi.org/10.1016/j.eneco.2006.11.001.

[27] K. Juselius, The Cointegrated VAR Model: Methodology and Applications (Advanced Texts in Econometrics), no. 2, Oxford University Press, Oxford, 2007 (2007).

[28] R. T. Baillie, G. G. Booth, Y. Tse, T. Zabotina, Price discovery and common factor models, Journal of Financial Markets 5 (3) (2002) 309 – 321, price Discovery (2002). doi:https://doi.org/10.1016/S1386-4181(02)00027-7.

[29] R. Mahadevan, J. Asafu-Adjaye, Energy consumption, economic growth and prices: A reassessment using panel VECM for developed and developing countries, Energy Policy 35 (4) (2007) 2481 – 2490 (2007). doi:https://doi.org/10.1016/j.enpol.2006.08.019.

Patent Analysis as a Tool for Revealing Promising Trends of Technological Development

V. Avdzeyko, V. Karnyshev, E. Pascal

Tomsk State University of Control Systems and Radioelectronics (TUSUR), Lenin Ave., 40, Tomsk, 634050, Russia
http://www.tusur.ru, pio@main.tusur.ru

Abstract. The paper proposed to use the method of patent analysis for revealing promising trends of technological development. In the authors' view such approach may be useful for short-term forecasting of new products. This method is considered to be perspective and accurate due to the fact that patent information goes ahead of industrial implementation of technological solutions. The given approach is demonstrated by using the International Patent Classification (IPC) and analysis of relevant US patents based on the example of the IPC main group H02M3/00 "Conversion of DC power input into DC power output". The results are based on time series of US patents issued in from 1976 to 2017 and covers the development trends of DC to DC power converters corresponding to IPC subgroups H02M3/02 – H02M3/42.

Keywords: Patent analysis, IPC, time series, short-term forecasting,US patents, power converters, trends

1 INTRODUCTION

To succeed in a project to be implemented, it is necessary to choose the most promising trend of development in the relevant area. Forecasting and planning significantly facilitate this choice and give an opportunity to concentrate financial, material, personnel, and other resources for solving the most urgent and promising engineering problems, to shorten the time needed for R&D, design engineering, and production of new equipment, to increase the equipment operating life, and to maximize the profits from the product sales.

A way for revealing the most promising trends of engineering (technological) development is the elaboration and practical use of forecasting methods.

2 TECHNOLOGICAL FORECASTING BASED ON PATENT ANALYSIS

In recent years, much attention has been paid to examination and improvement of advanced forecasting methods. The methods are intended to reveal prospects for specific

adfa, p. 1, 2011.

trends of research and development by gathering, processing, and analyzing the data contained in patents, papers, reports, and other published works that advance the state of the art of modern equipment and technology.

Intuitive forecasting methods are increasingly inferior to methods based on the analysis of statistical information representing the background of an area under study. We share the opinion of many authors engaged in R&D forecasting [1, 2] that bibliometric methods are gaining acceptance. These methods are based on the property of scientific and technical information (articles, patents, theses, conference proceedings, etc.) to reflect and be ahead of scientific and technological gains in industry

The bibliometric methods used for technological forecasting include patent, publication, and citation-index analyses and also methods for evaluating the significance of inventions and innovations. They have gained acceptance as they provide the opportunity to directly relate the dynamics of scientific and technological information to the progress in science and technology and to display the state of the art in various fields of science and technology and the trends in their development.

In predictive assessment of technology development, patent information, which contains a large amount of specific technical data (objectives, keywords, assignees, authors, etc.) is used most frequently [2]. An additional advantage of patent information is, in our opinion, the use of the International Patent Classification (IPC).

Patent statistics serves as a reliable and stable indicator of trends in the development of various technological areas. Campbell et al. [3] showed that patent data can be considered a forecasting tool for decision-making at the national, industry-specific, and corporate levels. Mogge [4] pointed out that statistical analysis of international patent data is a valuable tool for corporate technology forecasting and planning. Patent analysis is currently one of the best ways to detect engineering and technological changes, as it allows the occurrence of new products to be predicted at least 6-18 months before their market appearance.

3 USE OF THE INTERNATIONAL PATENT CLASSIFICATION

The above forecasting approaches and methods are associated with considerable data processing costs and loss of information during processing [5]. These shortcomings can be significantly reduced and even eliminated by using the IPC [6]. A method of technological forecasting was proposed [7] which uses the IPC for patent analysis. Using the IPC and ranking relevant patents according to the filing or registration date allows one to reveal trends in the development of technologies under investigation [8].

3.1 Patent analysis using the database of the US Patent and Trademark Office

The US Patent and Trademark Office (USPTO) has one of the largest seeded databases. In contrast to the databases of the European Patent Organization and other national agencies, it provides direct access to full-text patent descriptions.

Taking into account the peculiarities of the access to patent information provided by the USPTO, a software approach was proposed [9, 10] that allows one to build up lists of US patents according to a given set of patent numbers, IPC subgroups, and keywords, to create local seeded databases of US patents, to obtain information on quantitative distribution of patents over the IPC subgroups, to form patent data series, process them, and plot the results for the period since 1976 to the present.

In this paper, we consider a way to reveal promising trends of technology development by using the IPC and analysis of relevant US patents based on the example of the IPC main group H02M3/00: Conversion of DC power input into DC power output.

3.2 Analysis of DC/DC converters

DC/DC converters are designed to match the DC voltage of a power supply line (power source) to the input voltage of a load.

According to the IPC, DC/DC converters can be made to operate without conversion (subgroups H02M3/02 – H02M3/20) and with (H02M3/22 – H02M3/44) intermediate conversion into AC.

Converters without intermediate DC/AC conversion (direct DC/DC converters) are used, for example, when specifications do not require galvanic isolation between the primary power supply (battery) and the load, or when one or more auxiliary power supplies or converter units of a control circuit need to be directly connected to the supply line. Recently, electric circuits of this type have found wide application in microelectronic devices.

Converters with intermediate DC/AC conversion (DC/AC/DC converters) are used to provide galvanic isolation between the power supply line and the load. In this case, the sequential DC/AC plus AC/DC conversion is performed with an increased frequency to provide high weight and size parameters of the converters.

To perform patent analysis for DC/DC converters, a database of US patents belonging to the IPC main group H02M3/00 was created for the period from 1976 to 2017. Figure 1 presents diagrams showing the number of registered US patents for direct DC/DC converters and for DC/AC/DC converters. The total number of patents for the period from 1976 to 2017 for the main group was 14,309. The number of patents for direct DC/DC converters and for DC/AC/DC converters was 6,863 and 7,840, respectively.

Until 2002, the number of patents for both types of converters increased, and from 2002 to 2006, there was a sharp decrease in their number. Since 2006 to 2017, there was an increase in the number of granted patents: from 42 to 1409 for direct DC/DC converters and from 181 to 837 for DC/AC/DC converters.

The sharp decrease in the period from 2002 to 2006 can be explained by a number of reasons:

- an almost 20% decrease in the total number of issued US patents in these years;
- a sharp increase in the industrial production of converters with various input and output parameters (which proves the relation of patent activity to the output and range of industrial products);

- registration of the claimed patents in irrelevant groups and subgroups of the IPC, modification of the IPC itself, as well as by many other factors the study of which is beyond the scope of this work.

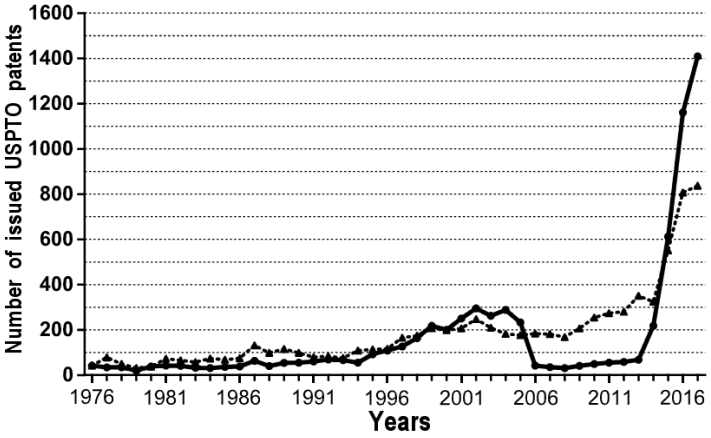


Fig. 1. Number of USPTO patents issued for direct DC/DC (solid line) and DC/AC/DC converters (dashed line).

As we dealt with short-term and medium-term forecasting, the analysis of the trends of converter development was carried out for the period from 2007 to 2017.

3.3 Analysis of converters without intermediate DC/AC conversion

In accordance with the IPC, direct DC/DC and DC/AC/DC converters can be made as static converters (SCs), dynamic converters (DCs), or as a combination of static and dynamic converters, or as a combination of rotary converters and other dynamic and/or static converters.

In view of the fact that only static converters are currently used in practice, an analysis of devices made on their basis has been carried out for the last 10 years since the beginning of the increase in the number of registered patents (Fig. 2).

For this period, 3,737 and 4,231 patents have been issued for direct DC/DC and DC/AC/DC converters, respectively. Thus, these converter types are equivalent in number of patents. However, while in 2014, DC/AC/DC converters went beyond direct DC/DC converters in number of patents, the number of patents issued in 2015-2017 for direct DC/DC converters (3546) is almost 30% more than for DC/AC/DC converters (2716).

Let us consider the element base of semiconductor devices used to design the converter types under comparison. In accordance with the IPC, direct DC/DC converter circuits can be based on thyristors (H02M3/125 – 3/142) or transistors (H02M3/145 – 3/158). Only 11 patents were issued for the thyristorized models in 2007-2017 and 2,869 for the transistorized models.

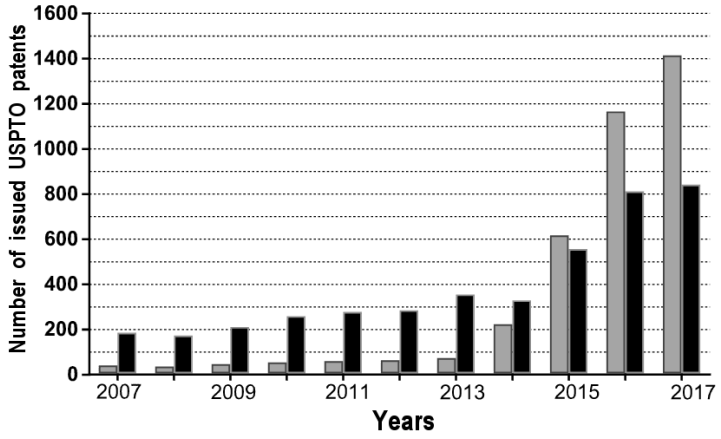


Fig. 2. Patent issuance dynamics for direct DC/DC (black bars) and DC/AC/DC (grey bars) converters.

It follows that thyristorized converters are not promising, which seems to be due to their large dimensions and the need for additional "quenching" devices. Let us consider transistorized converters of different circuit design.

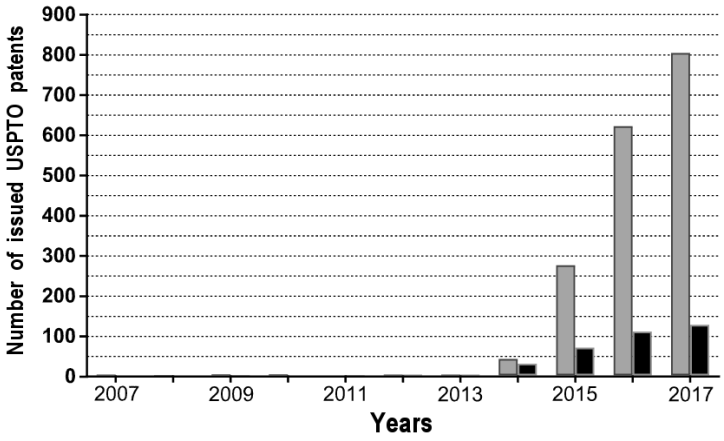


Fig. 3. Transistorized converter circuits with digital control and one semiconductor device (black bars) and with several semiconductor devices used as a terminal control unit for a single load (grey bars).

Figure 3 shows diagrams of the number of patents issued for converter circuits with automatically controlled voltage or digitally controlled current (H02M3/157) and for circuits made using several semiconductor devices as a terminal control unit for a single load (H02M3/158).

In 2017, the number of patents issued for circuits with several semiconductor devices switched by a given algorithm to control (stabilize) the output parameters of the converters was 6.4 times that issued for circuits with one semiconductor device.

3.4 Analysis of converters with intermediate DC/AC conversion

Similarly, we analyze converters with intermediate DC/AC conversion. Like with direct DC/DC circuit designs, the transistorized circuits are significantly superior to the thyristorized circuits. Over the past 10 years, 3,941 patents have been issued for transistorized converters (H02M3/325, H02M3/335 - H02M3/338) and only 30 for thyristorized ones (H02M3/305 - H02M3/315), indicating the promise of using transistors as semiconductor devices in the output units of converters. The search for promising converter circuits is illustrated by Fig. 4, which shows the number of patents issued for converters with push-pull circuits (H02M3/337) and with self-oscillating circuits (H02M3/338). In 2017, 102 patents were issued for push-pull converters and only 12 patents for self-oscillating converters.

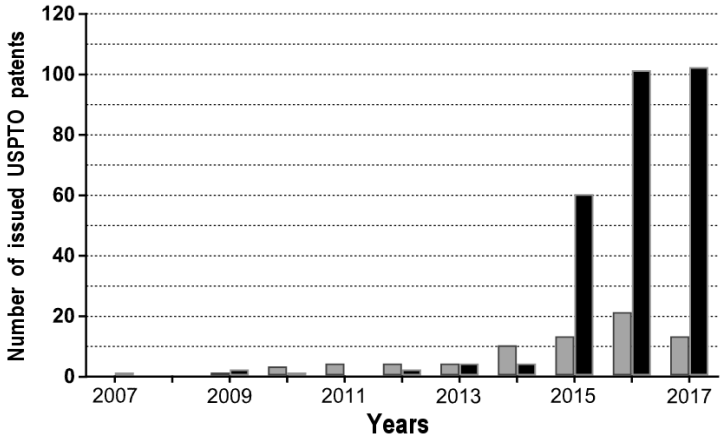


Fig. 4. The number of patents issued for push-pull (black bars) and self-oscillating converters (grey bars).

4 CONCLUSIONS

1. The analysis performed on the example of one group of the IPC not only has shown the capabilities of the patent analysis method for revealing promising trends of technological development, but also has substantiated the possibility to develop forecasts for further development based on the results obtained.
2. Classification of specific research areas in terms of the IPC makes it possible to use patent analysis for revealing both promising and dead-end trends in the development of equipment and technology.

3. The advantage of converter circuits without intermediate DC/AC conversion is based on advanced solutions in microelectronics.
4. Circuits with intermediate DC/AC conversion are advisable to use in converters with increased output power or with a significant difference between the voltage of the primary power source and the required output voltage of the converter.
5. Direct DC/DC converters with self-controlled voltage or current are more promising to design based on several switched semiconductor devices.
6. DC/AC/DC converters based on push-pull circuits have a significant advantage over self-oscillating converters.

ACKNOWLEDGMENT

The work was supported in part by the Russian Foundation for Basic Research (Project No. 18-07-01270 A).

REFERENCES

1. Ayse Kaya Firat, Wei Lee Woon, Stuart Madnick. "Technological Forecasting – A Review," Working Paper CISL# 2008-15, September 2008.
2. V. Coates, M. Faroque, R. Klavins, K. Lapid, H.A. Linstone, C. Pistorius, A.L. Porter. "On the future of technological forecasting," Technol. Forecast. Soc. Change 67 (1) (2001), pp. 1-17.
3. R.S. Campbell. "Patent trends as a technological forecasting tool," World Patent Information. Vol. 5 (1983), pp. 137-143.
4. M.E. Mogee. "Using patent data for technology analysis and planning," Research Technology Management. Vol. 34, pp. 43-49 (1991).
5. M. Fattori, G. Pedrazzi, R. Turra. "Text mining applied to patent mapping: a practical business case," World Patent Information. Vol. 25 (2003), pp. 335-342.
6. S. Jun. "IPC code analysis of patent documents using association rules and maps-patent analysis of database technology," Communications in Computer and Information Science. Vol. 258 (2011), pp. 21-30.
7. J. Kim, M. Hwang, Jeong Do-Heon, H. Jung. "Technology trends analysis and forecasting application based on decision tree and statistical feature analysis," Expert Systems with Applications. Vol. 39 (2012), pp. 12618-12625.
8. M. Bengisu, R. Nekhili. "Forecasting emerging technologies with the aid of science and technology databases," Technological Forecasting & Social Change. Vol. 73, (2006), pp. 835-844.
9. V.I. Avdzejko, V.I. Karnyshev, R.V. Meshcheryakov, Parnyuk L.V. "Forecasting of development trends and spotting breakthrough technologies in the field of converter equipment," Izvestiya Vysshikh Uchehnykh Zavedenii. Elektromekhanika (Russian Electromechanics). Vol. 60 (2017), No. 2, pp. 51-56 (in Russian)
10. [10] V.I. Avdzejko, V.I. Karnyshev, R.V. Meshcheryakov. "Forecasting of power electronics development directions based on international patent classification time series," Elektrotekhnicheskie I Informatsionnye Kompleksy I Sistemy. Vol. 12 (2016), No. 2, pp. 23-28 (in Russian)

Time series analysis of rainfall from climate models under the future warming scenarios over western Himalayan region

Sudip Kumar Kundu[1] and Charu Singh[2]

[1] Project Assistant, Centre for Atmospheric and Oceanic Sciences, Indian Institute of Science, Bangalore- 560012, KA, India
[2] Scientist/Engineer – SE, Marine and Atmospheric Sciences Department, Indian Institute of Remote Sensing, Dehradun- 248001, UK, India

Emails: sudipkrkundugeoh@gmail.com,
charu@iirs.gov.in

Abstract. Due to widely reported global warming, events like melting glaciers, rising sea level and changing weather patterns are being observed across the globe. The present study deals with the time series analysis of rainfall under various future warming scenarios namely RCP 4.5 and RCP 8.5 (2076-2100) with respect to the reference historical period (1976-2000) during the principal monsoon season (i.e.1st June to the 30th September) over the western Himalayan region. To serve this purpose, we have considered the daily monsoon season rainfall data based on the archives of five coupled climate models which participated in Climate Model Inter-comparison Project Phase 5 (CMIP5). Considering huge inter-model variations amongst the CMIP5 models, here results are also reported based on the multi-model mean (MMM). Characteristics of the rainfall pattern in term of time series during the historical time period are compared with that of the two future warming scenarios viz; RCP 4.5 and RCP 8.5 by means of statistical techniques. There is a possibility of an increase in the mean monsoon season rainfall at daily scale under RCP 4.5 and RCP 8.5 over that region compared to the historical MMM (HMMM). Both of the western Himalayan Indian states Uttarakhand (UK) and Himachal Pradesh (HP) would experience more enhancements in the daily intensity of rainfall under the warmest future warming scenario (i.e. RCP 8.5). Based on the present analysis, intense rainfall events are anticipated over this mountainous region which could act as an initiator for various meteorological hazards over the western Himalayan region in future.

Keywords: CMIP5, rainfall, monsoon season, time series, RCP 8.5

adfa, p. 1, 2011.

1 Introduction

The main characteristics of monsoon climate over the Indian region are wet summer and dry winter. In summer, southwest monsoon blows from Indian Ocean and carries moisture, causes summer rain in India. On the other hand, northeast monsoon originating over landmass of Siberia and become cold and dry, therefore rainfall doesn't take place in winter season. India receives more than 75 % of total annual rainfall during its principal monsoon months June, July, August and September (JJAS). Considering the huge population in India, monsoon season rainfall plays an important role in the country's socio-economic aspects especially in the industry and agricultural sectors (Meher et al., 2017). Himalayan region has been treated as 'Water tower of Asia' (Choudhary & Dimri, 2017) as it is the huge source of water in the form of liquid (carried by different drainage system) and solid (snowfall and ice on the mountain) both. As a great physiographic divide, it becomes an obstacles and forces rain-bearing southwest monsoon to provide most of the moisture before crossing it northward which resulted heavy precipitation in the Indian side (Ray et al., 2011). The climatic condition of western Himalayan region is governed by the south-west monsoon from the month June to September and westerly disturbance from November to March (Kumar et al., 2016). Thereafter, the monsoon season rainfall over this region is related to the southwest branch of Indian summer monsoon. Now a day's, extreme weather conditions such as heavy precipitation, cloud burst, flash flood, landslide and extreme avalanches are the regular happening incidents in the region of western Himalayan (Britannica.com, 2019). In that region severe effect on water availability has been created due to the overall changing pattern of rainfall which leads to water stress as well as drought (Ray et al., 2011). Recent study indicates that the rainfall over that region becomes more intense in such a way that more precipitation takes place over a short time period. As a result, higher intensity and incidence of floods, especially flash flood happens in the river basins. Ray et al., (2011) also indicate that rainfall in western Himalayan region becomes more unpredictable as there is a considerable variation between the duration of monsoon and the amount of rainfall in the different places of that region. There is negligible change observed in winter precipitation but a detectable decreasing trend has been observed in monsoon precipitation (Bhutiyani et al., 2010). At this stage, it is very crucial to investigate the rainfall pattern under the climate change scenarios over the western Himalayan region understand the physical mechanism associated with these changes. Therefore, climate models have been treated as the primary tools to investigate the response of the climate system for various forcing and make predictions of future climate on the seasonal as well as decadal time scales for upcoming century and the beyond (Flato et al., 2013). To project the future climate, it is mandatory to know about scenarios which can represent the equally feasible futures under the different amount of greenhouse gases (GHG) emissions. According to the AR5 of IPCC, the simulations from CMIP5 will become high precedence for most of the major climate modeling centers on the field of research agendas. It has been aimed for better understanding of climate and also to project future climate change (Taylor et al., 2007). Chaturvedi et al, (2012) noted that the ensemble mean of CMIP5 climate models are able to represent observed climate very closely

than any other individual model. In that context, the present study is planned for the time series analysis of rainfall from some selected CMIP5 models during JJAS under RCP 4.5 and RCP 8.5 compare to reference historical period over the western Himalayan region.

2 Study region

The study region for the present study is western Himalayan region (Fig. 1) which encompasses three Indian states namely Jammu & Kashmir (J&K), Himachal Pradesh (HP) and Uttarakhand (UK). The geographical extension of this region is lying in between 28° N to 37° N latitude and 72° E to 82° E longitude. The height of this region ranges from 170 m to 7861 m and consists of some well-known Himalayan peaks like Nanda Devi (7,816 m), Trisul (7,120 m), Kamet (7,756), Kinnaur Kailash Peak (6,500 m) etc.

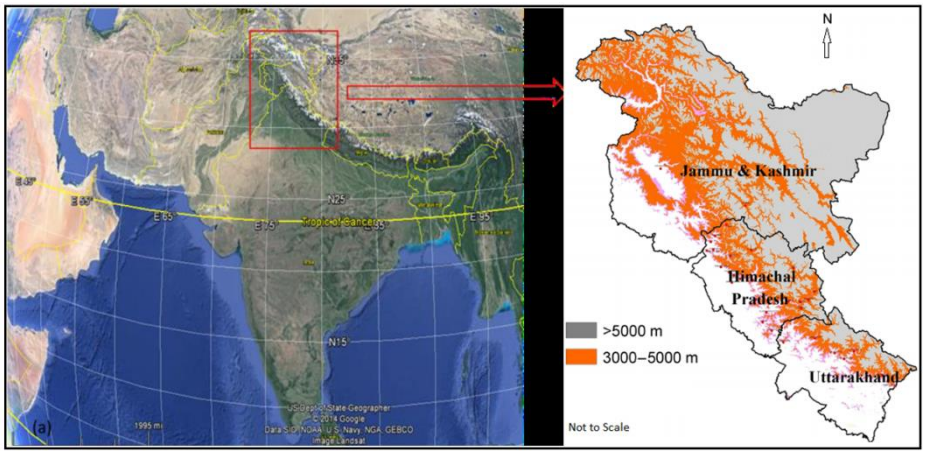


Fig. 1. Location of the states of Uttarakhand, Himachal Pradesh and Jammu and Kashmir in western Himalayan region (source www.google.com)

3 Data sets and Methodology

3.1 Used datasets

The present study is based on the daily rainfall data from some selected CMIP5 models (details are discuss in table 1) for the period 1976 to 2000 (reference historical period) and 2076 to 2100 (RCP 4.5 and RCP 8.5) over the western Himalayan region. We have used five CMIP5 models namely MRI-CGCM3, CESM1-CAM5, CMCC-CM, MIROC5 and MPI-ESM-LR along with their MMM as those models have better ability to simulate Indian summer monsoon according to Sabeerali et al., (2013). The IMD gridded rainfall data with resolution 0.5˚×0.5˚ (latitude × longitude) has been

utilized for the validation purpose. The daily rainfall data extracted from selected CMIP5 models have been compared with the IMD gridded rainfall data during 1976 to 2000.

Table 1. Details of five CMIP5 models utilized in the present study

Models	Contributing Institute	Extension		Resolution	
		Longitude (lon) in deg	Latitude (lat) in deg	lon (deg)	lat (deg)
MRI-CGCM3	Meteorological Research Institute, Japan	0 to 358.88	-89.14 to 89.14	1.12	1.12
CESM1-CAM5	Climate and Global Dynamics Laboratory (CGD) at the National Center for Atmospheric Research (NCAR), U. S	0 to 358.75	-90 to 90	1.25	0.94
CMCC-CM	Centro Euro-Mediterraneo per I Cambiamenti Climatici, Italy	0 to 359.25	-89.43 to 89.43	0.75	0.75
MIROC5	Atmosphere and Ocean Research Institute (The University of Tokyo), National Institute for Environmental Studies, and Japan Agency for Marine-Earth Science and Technology, Japan	0 to 358.59	-88.93 to 88.93	1.41	1.40
MPI-ESM-LR	Max Planck Institute for Meteorology (MPI-M), Germany	0 to 358.13	-88.57 to 88.57	1.86	1.87

3.2 Methodology

To investigate the rainfall intensity at daily scale under the future warming scenarios over the western Himalayan region the present study has been carried out based on the time series analysis from five selected CMIP5 models along with their MMM during JJAS (fig. 2). To serve that purpose, rainfall intensity has been projected in terms of time series during 2076 to 2100 under both the RCPs compare to historical reference period 1976 to 2000 over both the western Himalayan states UK and HP. The rainfall data are extracted and plotted at daily scale from CMIP5 models along with their MMM and IMD gridded data during 1976 to 2000 for the validation purpose. Thereafter, the daily rainfall data from CMIP5 models along with their MMM have also been plotted to project the future rainfall intensity under RCP 4.5 and RCP 8.5 during 2076 to 2100 for both the Himalayan states. Statistically we have fitted

trend line for all the time series analysis to find out the decreasing and/ or increasing trend of rainfall intensity during JJAS. Here, we have considered maximum intensity of rainy season (MRS) which denotes the maximum rainfall for a day during the JJAS, expressed by mm/day. The corresponding day of the maximum intensity rainfall (DMRS) during rainy season has also projected under the RCP 4.5 and RCP 8.5 compare to HMMM.

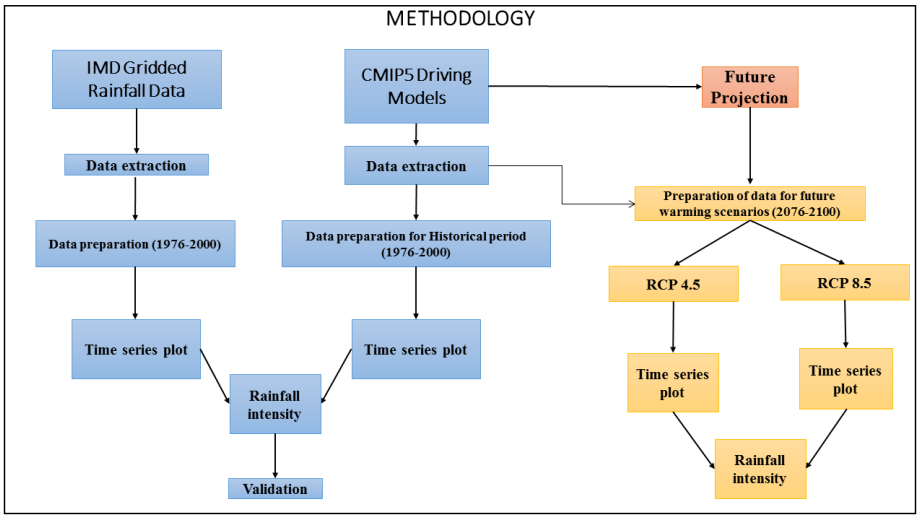


Fig. 2. Flow chart for research methodology

4 Results and Discussions

The time series of daily mean monsoon season rainfall from five CMIP5 models have been represented along with IMD rainfall data during 1976 to 2000 over both the state UK and HP separately. Thereafter, we have projected the future rainfall from those CMIP5 models along with MMM under the RCP 4.5 and RCP 8.5 during 2076 to 2100 over both the state UK and HP. The time series has also been done to project the MRS as well as DMRS during JJAS under RCP 4.5 and RCP 8.5 from 2076 to 2100 compare to historical reference data.

4.1 Validation of rainfall data

During the 1976 to 2000, the daily mean rainfall of JJAS (fig.3) from IMD gridded data indicates an increasing trend where the rainfall varies from 5 to 10 mm/day over UK. But no individual CMIP5 models are able to simulate daily mean monsoon season rainfall during 1976 to 2000. Although, the models MIROC5 and MPI-ESM-LR are closely capture the rainfall data compare to IMD, it show negative trend. CESM1-CAM5 overestimates the rainfall data where MRI-CGCM3 and CMCC-CM underestimate the same over UK. In case of HP, the daily mean monsoon season rainfall from

IMD gridded data indicates an increasing trend with varying in between 5 to 10 mm/day during 1976 to 2000. The CMIP5 model CESM1-CAM5 is able to capture rainfall data very closely to the IMD rainfall data with increasing trend. Rest four CMIP5 models including MIROC5 and MPI-ESM-LR failed to capture daily mean monsoon season rainfall data during 1976 to 2000 over HP. So, it is advisable to use MMM of five CMIP5 models rather than individual for future projection.

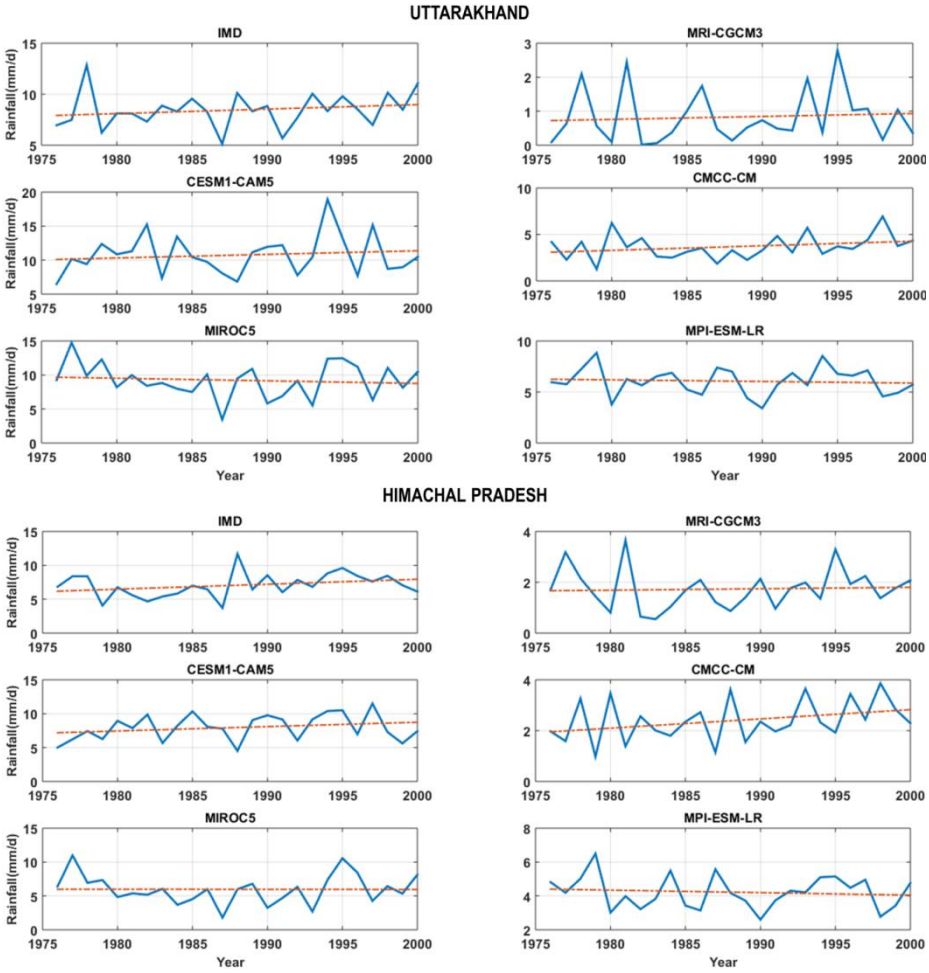


Fig. 3. Time series plots from five selected CMIP5 models along with IMD data using daily mean monsoon season rainfall during JJAS for the period 1976–2000 over Uttarakhand and Himachal Pradesh

4.2 Projection of mean monsoon season rainfall data

The daily mean monsoon season rainfall have been projected from five CMIP5 models along with their MMM under the RCP 4.5 (fig. 4) and RCP 8.5 (fig. 5) during 2076 to 2100 over both the state UK and HP. Both the states will experience an increasing trend of monsoon season rainfall according to MMM of five CMIP5 models under RCP 4.5. The mean monsoon season rainfall may be varies in between 4 to 8 mm/day and 4 to 6 mm/day over the state UK and HP respectively.

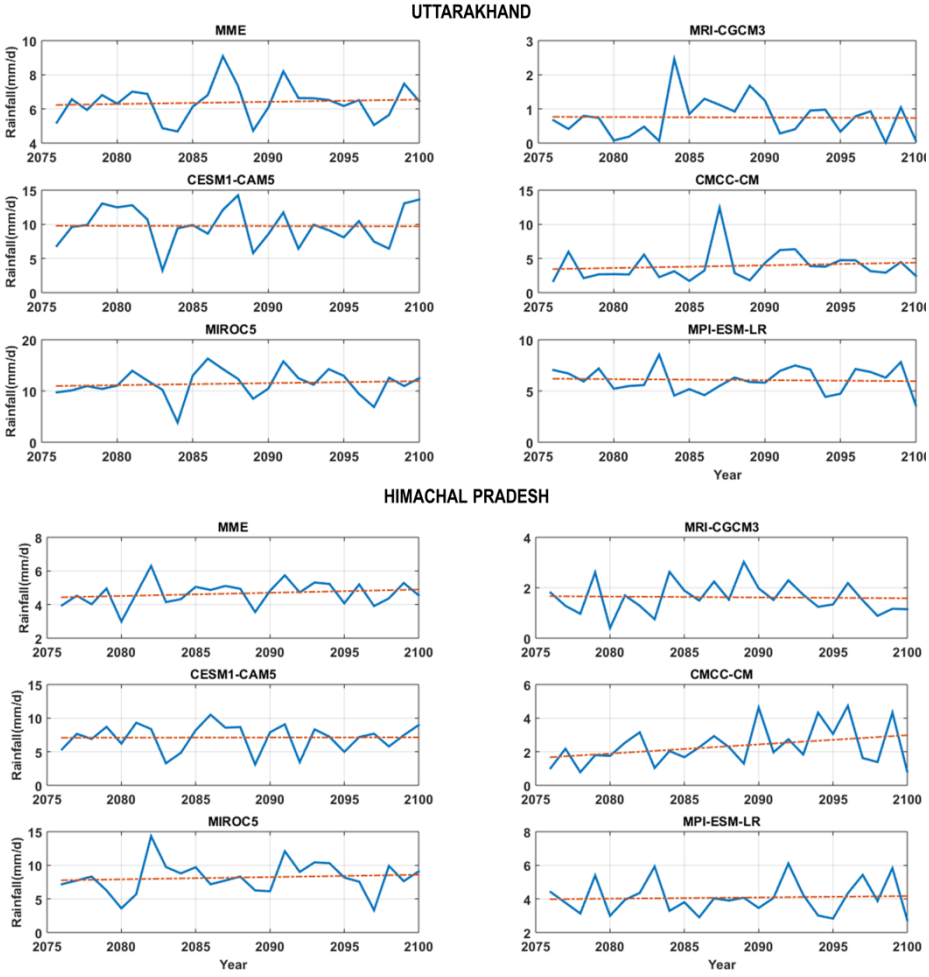


Fig. 4. Time series plots from five selected CMIP5 models along with their MMM using daily mean monsoon season rainfall during JJAS under RCP 4.5 for the period 2076–2100 over Uttarakhand and Himachal Pradesh

Under the warming scenarios RCP 8.5, both the states will experience negative trend of monsoon season rainfall during 2076 to 2100. However, the mean monsoon season

rainfall may be varies from 7 to 9 mm/day and 4 to 8 mm/day over the state UK respectively which is higher than the RCP 4.5 scenarios

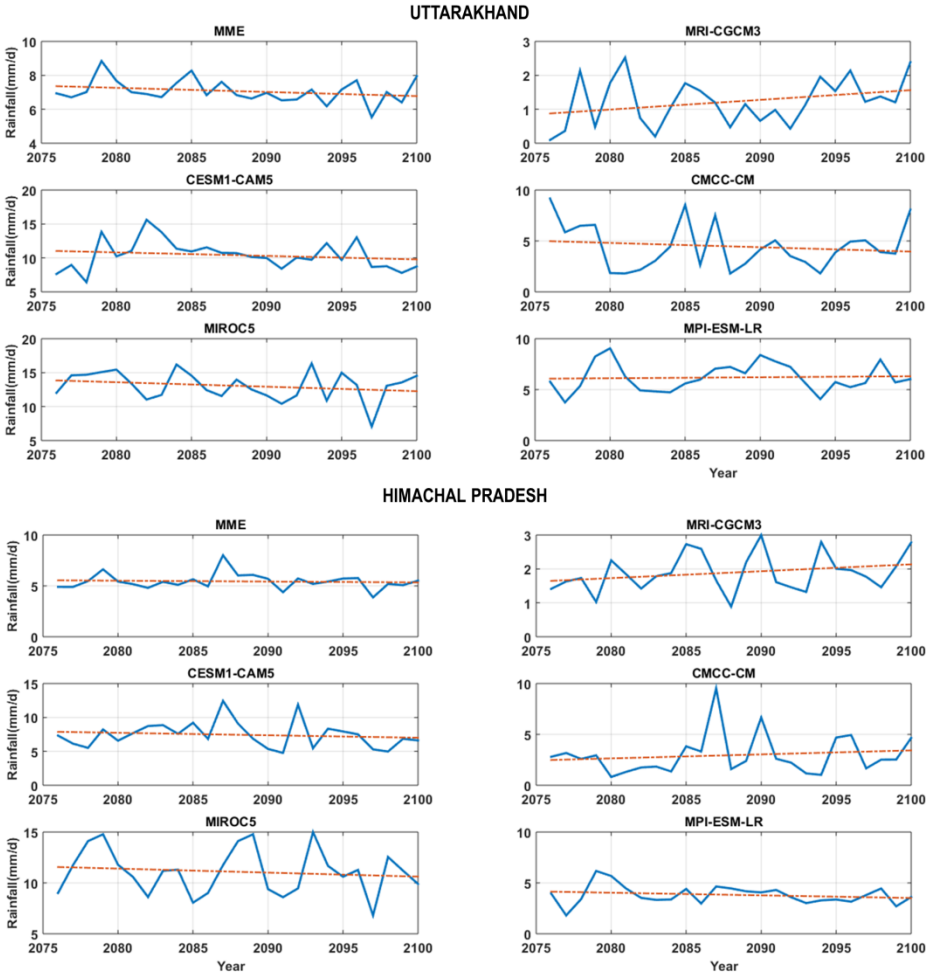


Fig. 5. Time series plots from five selected CMIP5 models along with their MMM using daily mean monsoon season rainfall during JJAS under RCP 8.5 for the period 2076–2100 over Uttarakhand and Himachal Pradesh

4.3 Projection of rainfall intensity during monsoon season

The one-day maximum intensity rainfall during JJAS has also been projected during 2076 to 2100 under the warming scenarios RCP 4.5 and RCP 8.5 (fig. 7) compare to historaical reference period (fig. 6) over both the states UK and HP.

The MRS of monsoon season rainfall from IMD and MMM of five CMIP5 models both indicate increasing trend during 1976 to 2000 over both the states UK and HP.

The MMM of CMIP5 models are able to simulate MRS closely to the IMD data over UK which is varies in between 45 to 55 mm/day during 1976 to 2000.

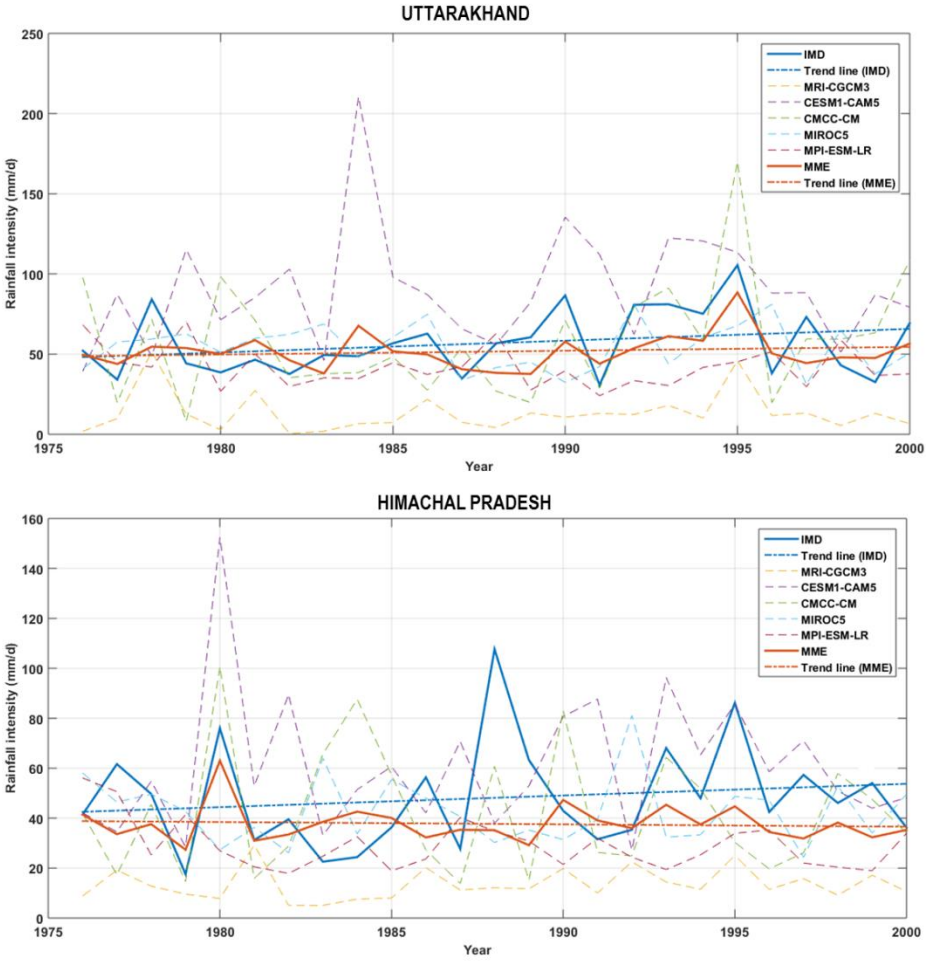


Fig. 6. Time series plots from five selected CMIP5 models along IMD data using one-day maximum intensity rainfall during JJAS for the period 1976–2000 over Uttarakhand and Himachal Pradesh

The MMM of five CMIP5 models capture increasing trend of MRS with higher intensity rainfall over both the state (except UK) under RCP 4.5 and RCP 8.5 during 2076 to 2100. The UK state will experience negative trend of MRS under RCP 8.5 during 2076 to 2100, although the rainfall intensity will be increased under RCP 8.5 than RCP 4.5 over both the states.

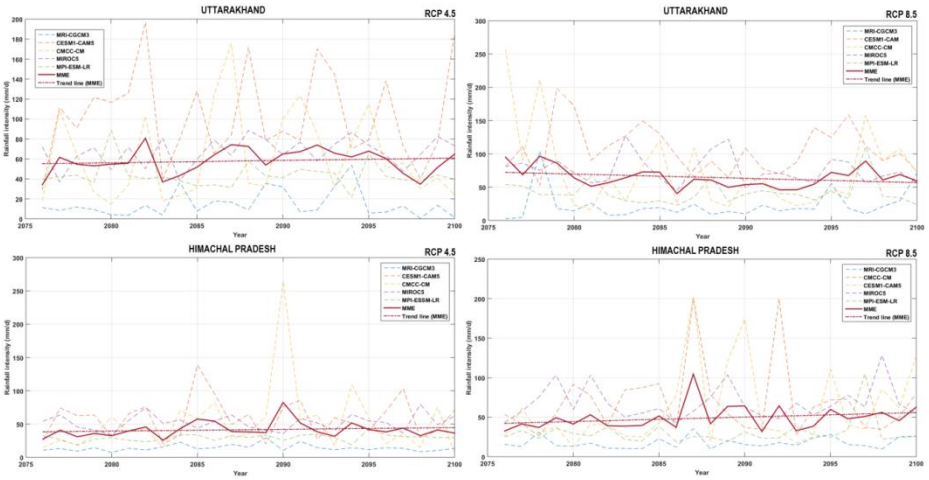


Fig. 7. Time series plots from five selected CMIP5 models along with their MMM using one-day maximum intensity rainfall during JJAS under RCP 4.5 and RCP 8.5 for the period 2076–2100 over Uttarakhand and Himachal Pradesh

It can also be said from the time series analysis that, the DMRS during JJAS will be delayed under the warming scenarios RCP 4.5 and RCP 8.5 over both the state UK and HP (except RCP 4.5).

5 Conclusions

Climate models are built on well-established physical principles but it has a better prediction power on the temperature pattern than the precipitations (Farber, 2007). Climate projections are more uncertain for mountain regions compared to plains mainly due to complex orographic dependent climatic regime. Choudhary & Dimri, (2017) noticed that climate modeling is very difficult in Himalayan region due to the spatial discrepancy in orography and different climatic regimes. The present study utilized five CMIP5 models along with their MMM for the time series analysis of rainfall during 2076 to 2100 (RCP 4.5 and RCP 8.5) compare to historical reference period 1976 to 2000. The CMIP5 models are also been validated with the IMD gridded rainfall data during 1976 to 2000 and found to match well with MMM than its individual. It can be projected from the time series analysis that there is an increasing trend of daily mean monsoon season rainfall under the RCP 4.5 and RCP 8.5 compare to historical data. There is also a possibility of increase in the one-day maximum intensity rainfall over both the states UK and HP under the future warming scenarios, however more enhancement in the MRS will take place under the warmest future warming scenario (i.e. RCP 8.5).

Acknowledgements Present work is a part of the EOAM project. Dr. V. Venugopal (IISc, Bangalore) is thankfully acknowledged for discussion. Authors would like to

thank Group Head MASD, Dean (Academics) and Director of IIRS for support and encouragement. The CMIP5 data has been taken from https://esgf-data.dkrz.de/search/esgf-dkrz/. We thank IMD for developing and providing the gridded rainfall data set for research purpose.

References

1. Bhutiyani, M. R., Kale, V. S., & Pawar, N. J. (2010). Climate change and the precipitation variations in the northwestern Himalaya: 1866-2006. *International Journal of Climatology*, *30*(4), 535–548. https://doi.org/10.1002/joc.1920
2. Britannica.com. (2019). *Himalayas - Climate | mountains, Asia*. 1–3.
3. Chaturvedi, R. K., Joshi, J., Jayaraman, M., Bala, G., & Ravindranath, N. H. (2012). Multi-model climate change projections for India under representative concentration pathways. *Current Science*, *103*(7), 791–802. https://doi.org/10.2307/24088836
4. Choudhary, A., & Dimri, A. P. (2017). Assessment of CORDEX-South Asia experiments for monsoonal precipitation over Himalayan region for future climate. *Climate Dynamics*, *0*(0), 1–22. https://doi.org/10.1007/s00382-017-3789-4
5. Farber, D. a. (2007). Climate Models: A User's Guide. *UC Berkeley, Public Law and Legal Theory Research Paper Series*, *901*(1030607), 1–46. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1030607#%23
6. Flato, G., Marotzke, J., Abiodun, B., Braconnot, P., Chou, S. C., Collins, W., … Rummukainen, M. (2013). Evaluation of Climate Models. *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*, 741–866. https://doi.org/10.1017/CBO9781107415324
7. Kumar, N., Yadav, B. P., & Bist, S. (2016). *Precipitation in the Himalayas*. *16*(96).
8. Meher, J. K., Das, L., Akhter, J., Benestad, R. E., & Mezghani, A. (2017). Performance of CMIP3 and CMIP5 GCMs to simulate observed rainfall characteristics over the western Himalayan region. *Journal of Climate*, *30*(19), 7777–7799. https://doi.org/10.1175/JCLI-D-16-0774.1
9. Ray, M., Doshi, N., Alag, N., & Sreedhar, R. (2011). Climate Vulnerability in North Western Himalayas. *Climate Vulnerability in North Western Himalayas*.
10. Sabeerali, C. T., Dandi, A. R., Dhakate, A., Salunke, K., Mahapatra, S., & Rao, S. A. (2013). *Simulation of boreal summer intraseasonal oscillations in the latest CMIP5 coupled GCMs*. *118*, 4401–4420. https://doi.org/10.1002/jgrd.50403
11. Taylor, K. E., Stouffer, R. J., & Meehl, G. a. (2007). A Summary of the CMIP5 Experiment Design. *World*, *4*(January 2011), 1–33. https://doi.org/10.1175/BAMS-D-11-00094.1

On the Evaluation of Similarity for Time Series

Silvia Maria Ojeda[1], Juan Carlos Bellassai Gauto[2] and Marcos Alejandro Landi[2,3]

[1] FAMAF-Universidad Nacional de Córdoba, Argentina
[2] CIEM-CONICET, Córdoba, Argentina
[3] Instituto de Altos Estudios Espaciales Mario Gulich- CONAE, Córdoba, Argentina
juancbellassai@gmail.com

Abstract. The search and detection of similarities is a central problem in the analysis and processing of time series databases. The issue is relevant, for example, in problems of classification of time series and in situations in which a predictive process must be evaluated, or when it is necessary to compare two or more prediction methods. Many of the works oriented to the evaluation of similarity in time series have focused on the notion of dynamic distortion, with good results in the quantification of similarity, but with a high computational cost. As a result, the interest in the development of new similarity indexes and the improvement of existing similarity measures remains in force; even more considering the remarkable increase and availability of time series databases and the urgency that applications demand daily. The expectation about the new proposals is that they are able to quantify quickly and not only effectively the similarity between time series, in response to different application problems. Therefore, an interesting alternative is to investigate about simple mathematical formulation measures, which have proven useful for measuring the similarity in two-dimensional scenarios and assess their adaptation to measure similarity between time series. One of the proposals to measure similarity between two-dimensional scenarios is the SSIM similarity index, defined to quantify similarity between digital images. The development was presented by Wang et al. in 2004 and has shown excellent results to evaluate the similarity between two digital images. SSIM has the advantage over other proposals, its simple mathematical formulation. In effect, this index is calculated from the product of three factors: the luminance, the contrast and the correlation between the images to be compared. These factors represent, respectively, simple relations between the means, the contrast and the correlation between the images. In this work, we adapted the SSIM index for images to the problem of evaluating the similarity in time series, obtaining a temporal similarity index called SSIMT.

Keywords: Time Series, Classification, Clustering..

1 Introduction

The validation of predictions is a central problem in the time series forecasting process to determine the performance of two or more predictions methods. One of the

main validation methods, is the comparison of prediction with an observed time series. Therefore, the measure and detection of similarities is a key topic for time series forecasting. Currently, many works oriented to the evaluation of similarity in time series have focused on the notion of dynamic distortion [1], [2], with good results in the quantification of similarity [3], but with a high computational cost [4]. Also, elastic similarity measures are widely used to determine if two time series are similar to each other, with excellent off-line results [1]. However, in the online configuration, where the available data continuously increases over time and not necessarily in a stationary way, the results are not as good as would be desired. This is due to the computational complexity of elastic similarity measures and their lack of flexibility to accommodate different non-stationary intervals, which makes them incompatible with the system requirements [1]. As a result, the interest in the development of new similarity indexes and the improvement of existing similarity measures remains in force; even more considering the remarkable increase and availability of time series databases and the urgency that applications demand daily.

In recent years new approach's based on the correlation behavior and the proximity between two time series have been proposed [5], [6]. In particular, *D* index proposed by Chouakria and Nagabhushan showed an high performance in the classification of time series [5]. This proposal combines the correlation and the proximity between the series in a multiplicative way, introducing a tuning constant controlling the weight of each quantity in the final product. This index is given as following:

Let $x = (u_i, \ldots, u_p)$ and $y = (v_i, \ldots, v_p)$ be two time series of p values observed at the time instants $(t_1, \ldots, t_p)$. The *D* index between *x* and *y* is defined as:

$$D(x,y) = f_k(\text{cort}(x,y)) * \delta_{conv}(x,y)$$

where

$$\text{cort}(x,y) = \frac{\sum_{i=1}^{p-1}(u_{i+1} - u_i) * (v_{i+1} - v_i)}{\sqrt{\sum_{i=1}^{p-1}(u_{i+1} - u_i)^2} + \sqrt{\sum_{i=1}^{p-1}(v_{i+1} - v_i)^2}}$$

f_k is an adaptive tuning function and δ_{conv} is a distance measure such as, Euclidean, Frechet, or Dinamic Time Warping distance, that summarizes the closeness of sequences *x* and *y*. There are many possible ways to choose a function f_k. Here, we follow the guidelines given in [5], according to which f_k is considered an exponential adaptive tuning function given by:

$$f_k(t) = \frac{2}{1 + \exp(kt)'}$$

The implementation of *D* index with the Euclidean distance has a low computational cost. Also, this is a flexible measure to compare two sequences with different behaviors, comparing them in terms of both correlation and dissimilarity. However, the performance of the *D* index to measure the similarity between two time series, strongly depends on the calibration of the k coefficient [5]. Currently, there is not a

standard method to calibrate k, which represents a major drawback to use of *D* index. Therefore the aims of this study were: To define a simple method to quickly quantify the similarity between two time series, using an approach based on the correlation behavior and the proximity between them, and evaluate in different contexts the performance of the proposed method, using simulated data and real data.

2 Methods

2.1 Notation

Usually an image *I* is represented by an *X* matrix of *n* rows and *m* columns, with non-negative coefficients. This matrix, called digital image, is a discretization of the function *f* ($f: D \subset R^2 \rightarrow R$) that defines the intensity of the image *I*: Another possible representation of *I* is from vectors, through the transformation bijective: $Vec(X) = (k_1^T, k_2^T, \dots, k_m^T)^T$, where k_i^T denotes the *i*th column of *X*. Thus, according to the vector representation, the Image *I*, is represented by a vector *x* of dimension $N = nm$, whose coordinates are positive real numbers. Formally, from the vector representation, the set of all images is expressed as: $R_+^N = \{x = (x_1, x_2, \dots, x_N) / \ x_i \in R_+\}$, where $R_+ = \{r \in R / r > 0\}$.

2.2 SSIM Formulation

Among the most successful proposals to assess similarity and quality between images, it is inevitable to mention the SSIM index, presented by [7]. This work, perhaps marks a before and after on the subject, since unlike previous works, it focuses on the intention to evaluate the similarity between two spatial processes incorporating contextual information, relative to them, with excellent well-documented results [7]–[10]. One feature of this coefficient is its superior performance when incorporating human visual perception with respect to the well-known root-mean-square error (RMSE) and peak-signal to-noise ratio (PSNR) indices [8]. Let *x*, *y* en R_+^N; the SSIM index is given by:

$$SSIM(x, y) = l(x, y)^{\alpha}. c(x, y)^{\beta}. s(x, y)^{\gamma}$$

where α, β and γ are parameters that are associated with the weight of each multiplicative coefficient,

$$l(x, y) = \frac{2\bar{x}\,\bar{y} + c_1}{\bar{x}^2 + \bar{y}^2 + c_1} \quad \text{(luminance)}$$

$$c(x, y) = \frac{2S_x S_y + c_2}{S_x^2 + S_y^2 + c_2} \quad \text{(contrast)}$$

$$s(x, y) = \frac{S_{xy} + c_3}{S_x S_y + c_3} \quad \text{(correlation)}$$

$\bar{x}$, $\bar{y}$, S_x, S_y, S_{xy} represent the sample means of *x* and *y*, the sample variances of *x* and *y*, and the sample covariance between *x* and *y*, respectively. The constants c_1; c_2

and c_3 are all positive. The constants c_1; c_2 and c_3 characterize the saturation effects of the visual system to guarantee stability when the denominators are close to zero. In practice, the values of these constants are very small. Here, we consider the balanced case, i.e., $\alpha=\beta=\gamma=1$.

The SSIM coefficient is not a metric in a strict sense; however, this is not an obstacle to the use of these components to construct a valid metric. In [11], the authors studied this problem and suggested suitable quantities to define normalized and generalized metrics based on the important components of the SSIM coefficient.

In this work, we adapted the SSIM index for images to the problem of evaluating the similarity in time series, obtaining a temporal similarity index called SSIMT. The expression of SSIMT coincides with that of SSIM, only that now x and y are two time series of the same length. We also defined two robust versions of SSIMT, called SSIMM and SSIMR. The first was obtained substituting in the mathematical formulation of SSIMT the means $\bar{x}$ and $\bar{y}$ by the respective medians $\tilde{x}$ and $\tilde{y}$ of the series. The second, called SSIMR, was defined from SSIMT, substituting the mean by the trimmed mean.

2.3 Experiments

We carried out three experiments in order to test the performance of the three proposed index. In the first two experiments we compared pairs of different simulated series, while in the third experiment we tested the performance of the methods using real time series obtained from satellite image. The new proposals were evaluated by an algorithm of Monte Carlo type classification. This algorithm is detailed below for a generic index E:

Let N be a natural number.

1. Generate d sets, $G_1, G_2,\ldots,G_d$, of s series of length n. Each set of series is defined from a different model.
2. Divide each set, $G_1, G_2,\ldots,G_d$, into two groups: Reference Group and Validation Group, with s_1 and s_2 cardinality, respectively. Then, mix the series of all validation groups and define G as the group that results from the union of the d validation groups.
3. For each series x in G, and for each reference group, calculate the average dissimilarity between x and each series of the reference group, by means of the index E. Then, assign x to the reference set with a minor average dissimilarity.
4. Define a vector of d components, V_1, containing in the ith component the number of series (of the ith set) that were resulted well-classified.
5. For $j = 2, \ldots N$, repeat steps 1 through 4. In the jth iteration, define a vector called V_j analogously to how V_1 was defined.
6. Define V as a vector of d components that results from averaging coordinate to coordinate $V_1, V_2, \ldots V_d$. V indicates on average, and by set, the number of series in G correctly classified, according to their reference group.
7. Define the performance of index E as the proportion of series in G correctly classified.

In the first experiment, four autoregressive stationary models of order 1 (d = 4) were considered (Fig. 1). For this case, a set of 30 series of length n = 200 was generated for each model, where each set had 20 reference series and 10 as validation series. The classification algorithm was applied with N = 100 iterations for SSIMT, SSIMM, SSIMR and D index. We implement the D index using the Euclidean distance, and k = 3.1, to maximize the performance of the selected distance [5].

$$X_t = aX_{t-1} + e_t \quad \text{AR(1) model}$$

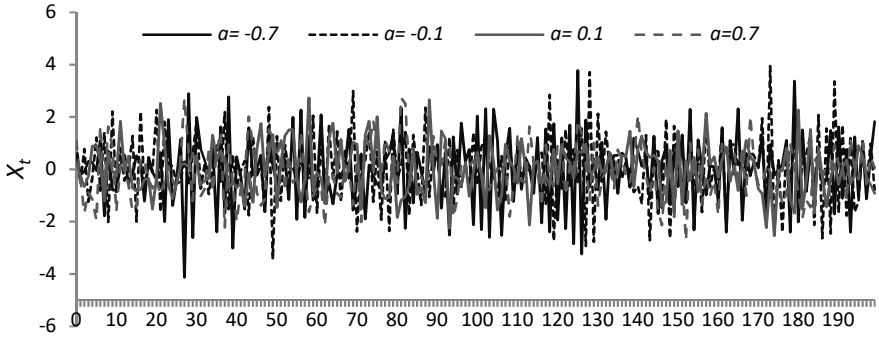


Fig. 1. Example of AR(1) time series with different a coefficient.

In a second experiment were define three models (d = 3) and was generate a set of 30 series of 200 length for each model. The first set of series was define from a process AR(1) stationary, of mean zero and variance one. The series of the second model were obtain by a process MA(1) with mean zero and variance four, while the series of the third model were generate from a process ARIMA (1,1,0). With the previous specifications, we made a total of N = 100 iterations for the algorithm of classification, where in each iteration, every set of series was divided in two groups, *RG* y *VG*, with 20 and 10 series respectively. Finally, the performance of each index was calculated. The indices used were SSIMT, SSIMM, SSIMR and D index.

$$X_t = e_t - \theta e_{t-1} \quad \text{MA(1) model}$$

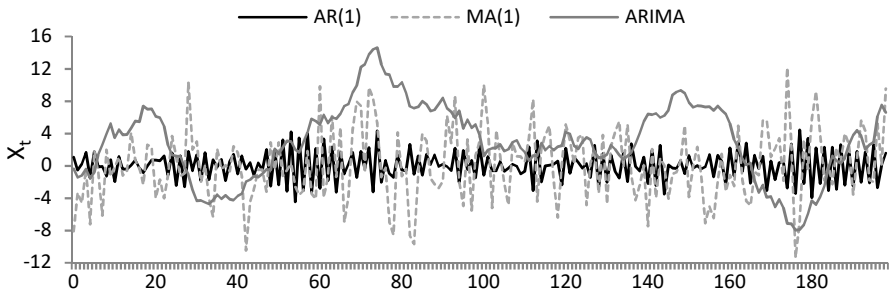


Fig. 2. Example AR(1), MA(1) and ARIMA time series.

A third experiment was carried out using time series of NDVI satellite data (Normalized Difference Vegetation Index) obtained from the MODIS sensor (Moderate Resolution Image Radiometer). NDVI data are the most commonly used information to study the ecosystem functioning [12], since they have has a strong correlation with the amount of biomass [13] and the photosynthetic activity [14]. The database used contains 285 series divided into five groups (d = 5) of different types of vegetation cover: Chaqueño forest, Serrano forest, Serrano grassland, Patagonian forest and Monte forest. Each group of vegetation cover contains 57 series of length n = 23 (one year). For this experiment, we applied the algorithm previously proposed, using 10 series of each group as reference series and 47 as validation series. This experiment was repeated using time series of 5 consecutive year.

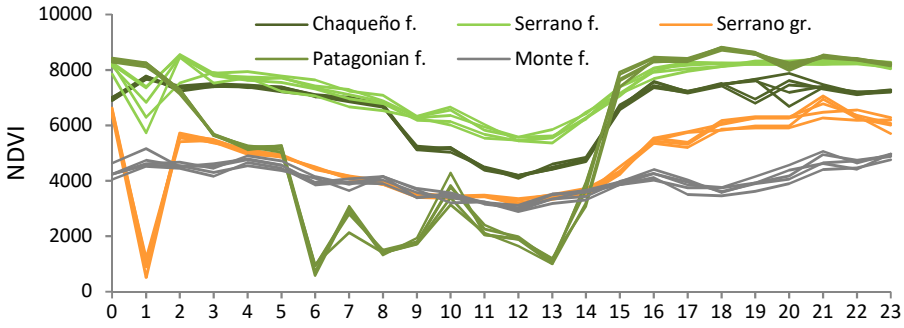


Fig. 3. Example of NDVI time series of one year length.

Finally, a second analysis for the clustering of the NDVI series was proposed, using the PAM (Partitioning Around Medoids) function in the software R (Package cluster 2.1). The basic PAM algorithm is fully described in chapter 2 of [15]. Knowing the true vegetation cover class of each series, we set up a confusion matrix and we saw if the indexes managed to group and correctly identify the series according to its type of vegetation cover. In this analysis we use series with length n = 23 (one year),

this being a desired length in the practice of analyzing this type of series. This experiment was repeated using time series of 5 consecutive year.

3 Results

The performance obtained in the different experiments showed that SSIM is a good method to compare time series. First experiment using AR1 models showed that the *D* index had the lowest performance with an overall accuracy of 25.2% (Table 1). Notably, results showed that the index had a cero accuracy to classify time series generated with coefficients a= -0.7 and 0.7. In contrast the SSIMT, SSIMM and SSIMR indices showed higher performance than *D* index, with a overall accuracy that ranges from 58.8 to 67.8%.

Table 1. Average percentage accuracy of the classification process

	D	**SSIMT**	**SSIMM**	**SSIMR**
AR(1) (a=-0.7)	0.00	91.0	92.7	91.0
AR(1) (a=-0.1)	54.0	25.6	26.5	61.4
AR(1) (a=0.1)	46.7	29.3	35.9	29.3
AR(1) (a=0.7)	0.0	89.3	88.0	89.3
Overall accuracy	25.2	58.8	60.8	67.8

Second experiment once more showed that the *D* index had the lowest performance with an overall accuracy of 36.4% (Table 2), while the different versions of SSIM index had an overall accuracy that ranges from 95 to 99.9%. We also noted that *D* index had an accuracy close to 0% to classify MA(1) and ARIMA time series, and that the four indexes had an accuracy of 100% to classify the AR(1) time series.

Table 2. Average percentage accuracy of the classification process

	D	**SSIMT**	**SSIMM**	**SSIMR**
AR(1)	100	100	100	100
MA(1)	0.00	98.50	86.70	86.50
ARIMA	0.09	98.60	98.50	98.60
Overall accuracy	36.40	99.90	95.10	95.00

The results of the third experiment performed with NDVI series showed that the four index have high performance, with an overall accuracy that ranges from 94.5 to 99.9 (Table 3). The results obtained in the clustering analysis with PAM function, showed a similar results, since the four index have high performance, with overall accuracy that ranges from 99.4 to 99.9.

Table 3. Average percentage of accuracy for the classification of NDVI time series

	D	**SSIMT**	**SSIMM**	**SSIMR**
Patagonian Forest	100	100	100	100
Serrano Forest	97.90	97.20	78.70	87.20
Chaqueño Forest	100	100	100	100
Monte Forest	100	100	100	100
Serrano Grassland	95.70	87.20	93.60	87.20
Overall accuracy	97.80	99.90	94.50	94.90

Table 4. Average percentage accuracy of the clustering process of NDVI time series

	D	**SSIMT**	**SSIMM**	**SSIMR**
Patagonian Forest	100	100	100	100
Serrano Forest	99.65	98.25	97.89	98.25
Chaqueño Forest	100	100	100	100
Monte Forest	100	100	100	100
Serrano Grassland	99.30	98.95	99.30	98.95
Overall accuracy	99.79	99.44	99.44	99.44

4 Conclusions

The results presented here showed that although the SSIM index was developed to measure similarity between images, it can be used as an index of similarity between time series (in this case called SSIMT). SSIMT and the two robust versions of the SSIMT proposed (SSIMM and SSIMR), showed better results than the *D* index developed by Chouakria and Nagabhushan [7], which is an index with a high performance [9], [16].

In the classification of real series (third experiment) the four compared indices showed a similar performance, with a percentage higher than 94% of correctly classified series. However, in the case of simulated series we observed large differences in the performance for the *D* and the proposed indices. Particularly, in the second experiment we observed notable differences, since the proposed indices correctly classified more than 90% of the series, while the *D* index correctly classified less than 40%. These differences in yield between D and the proposed indices may be due to the fact that the NDVI series have a predictable behavior determined by the seasonal photosynthetic activity of the vegetation, as opposed to the stationary behavior of the selected AR (1) and MA (1) models, and the non-stationary profile of the simulated ARIMA (1,1,0) model. Therefore, this suggests that the indices presented have greater capacity than the D index to assess similarity when the time series to be compared are stationary or non-seasonal. Considering that the D index and the proposed indices use different factors to assess the similarity or dissimilarity in distance, dispersion and correlation between two time series, it is necessary to conduct a thorough study to

understand the impact of each factor on the performance of each index, as well as the relationship among these factors.

As a future task, we propose to study the behavior of the indexes defined by implementing them in sliding mobile windows through the series. This idea would allow a global evaluation of the SSIMT, SSIMR and SSIMM indexes, based on their local evaluation. Although the computational cost would increase, a lower cost would be expected from the proposed indices in comparison to other initiatives, due to the ease of calculation.

5 References

[1] I. Oregi, A. Pérez, J. Del Ser, y J. A. Lozano, «On-line Elastic Similarity Measures for time series», *Pattern Recognit.*, vol. 88, pp. 506-517, abr. 2019.

[2] H. Li, C. Guo, y W. Qiu, «Similarity measure based on piecewise linear approximation and derivative dynamic time warping for time series mining», *Expert Syst. Appl.*, vol. 38, n.º 12, pp. 14732-14743, nov. 2011.

[3] F. Gullo, G. Ponti, A. Tagarelli, y S. Greco, «A time series representation model for accurate and fast similarity detection», *Pattern Recognit.*, vol. 42, n.º 11, pp. 2998-3014, nov. 2009.

[4] Q. Cai, L. Chen, y J. Sun, «Piecewise statistic approximation based similarity measure for time series», *Knowl.-Based Syst.*, vol. 85, pp. 181-195, sep. 2015.

[5] A. D. Chouakria y P. N. Nagabhushan, «Adaptive dissimilarity index for measuring time series proximity», *Adv. Data Anal. Classif.*, vol. 1, n.º 1, pp. 5-21, feb. 2007.

[6] R. Vallejos y S. Ojeda, «Image Segmentation and Time Series Clustering Based on Spatial and Temporal ARMA Processes», en *Advances in Image Segmentation*, P.-G. Ho, Ed. InTech, 2012.

[7] Z. Wang y A. C. Bovik, «A universal image quality index», *IEEE Signal Process. Lett.*, vol. 9, n.º 3, pp. 81-84, mar. 2002.

[8] Zhou Wang y A. C. Bovik, «Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures», *IEEE Signal Process. Mag.*, vol. 26, n.º 1, pp. 98-117, ene. 2009.

[9] W. Lin y C.-C. Jay Kuo, «Perceptual visual quality metrics: A survey», *J. Vis. Commun. Image Represent.*, vol. 22, n.º 4, pp. 297-312, may 2011.

[10] Ke Gu, Guangtao Zhai, Xiaokang Yang, Wenjun Zhang, y Min Liu, «Structural similarity weighting for image quality assessment», en *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, San Jose, CA, USA, 2013, pp. 1-6.

[11] D. Brunet, E. R. Vrscay, y Zhou Wang, «On the Mathematical Properties of the Structural Similarity Index», *IEEE Trans. Image Process.*, vol. 21, n.º 4, pp. 1488-1499, abr. 2012.

[12] I. Gitas, G. Mitri, S. Veraverbeke, y A. Polychronaki, «Advances in Remote Sensing of Post-Fire Vegetation Recovery Monitoring - A Review», en *Remote*

Sensing of Biomass - Principles and Applications, L. Fatoyinbo, Ed. InTech, 2012.

[13] N. I. Gasparri, M. G. Parmuchi, J. Bono, H. Karszenbaum, y C. L. Montenegro, «Assessing multi-temporal Landsat 7 ETM+ images for estimating above-ground biomass in subtropical dry forests of Argentina», *J. Arid Environ.*, vol. 74, n.º 10, pp. 1262-1270, oct. 2010.

[14] N. Pettorelli, J. O. Vik, A. Mysterud, J.-M. Gaillard, C. J. Tucker, y N. Chr. Stenseth, «Using the satellite-derived NDVI to assess ecological responses to environmental change», *Trends Ecol. Evol.*, vol. 20, n.º 9, pp. 503-510, sep. 2005.

[15] L. Kaufman y P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. Hoboken, N.J: Wiley, 2005.

[16] Zhou Wang y A. C. Bovik, «Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures», *IEEE Signal Process. Mag.*, vol. 26, n.º 1, pp. 98-117, ene. 2009.

On-The-Fly Dynamic Ensembles for Time Series Forecasting

Ahmed R. Elshami[1], Aliaa Youssef [1] and Mohamed W. Fakhr [1]

[1]Arab Academy for science and technology, Heliopolis, 2033 Cairo, Egypt
armhels@gmail.com
aliaay@yahoo.com
Waleedf@aast.edu

Abstract. This paper proposes two approaches for dynamic creation of prediction ensembles. Instead of using fixed regions in the space, we propose the "on-the-fly" idea which is employed in 2 approaches. In the first approach, global models are trained, and a dynamic validation set is created on-the-fly for each test vector based on random forest (RF) defined neighborhood. The dynamic ensemble weights are obtained from the models performance on this validation set. In the second approach, the test vector RF-neighborhood is divided randomly in half, where the first part is used to train the models and the other is used to validate and calculate the dynamic weights. 5 prediction models are used and the proposed approaches are tested on the monthly time series data from M3 competition with 1045 different time series, and with multi-step horizons ranging from 1 to 18. The proposed approaches show very competitive results with other machine learning and statistical forecasting techniques while significantly better results than the state of the art dynamic ensemble methods.

Keywords: Time series forecasting, Random forest, Local learning, Dynamic ensemble, Regression, Machine learning.

1 Introduction

Dynamic ensemble technique in time series prediction has shown promising performance in difficult multi-step forecasting problems[1]. They usually depend on static neighborhood regions and also on nearest neighbor calculations [[1],[2],[3]]. The former may lead to suboptimal regions and the former requires heavy calculations for each test vector. In this paper we propose a method to perform time series prediction based on random forest (RF) defined neighborhood [4]. The proposed method is based on finding all the training examples that share the same leaves in the RF with the test vector. This paper proposes 2 approaches to create the dynamic ensemble; In the first approach, for each test vector we create an on-the-fly dynamic validation set composed of its RF defined neighborhood. Five machine learning models are trained globally (namely; support vector machine, neural network, K-nearest neighbor, Gaussian processes regression and linear regression). Normalized weights are assigned to each of

them based on their performance on the dynamic validation set for each test vector. In the second approach we create a larger RF-based neighborhood, where we take randomly half of it to train the 5 models and the other half for validation to find their normalized weights.

2 Random Forest Based Neighborhood

RF is used to find the nearest neighborhood between test vector and training examples [4], (see Fig. 1). For each test vector, all the training examples that share the same leaves with the test vector are sorted based on the number of times they share the same leaves. A threshold based on percentage of appearance in all trees is then applied (10% for the 1st and 20% in the 2nd approach).

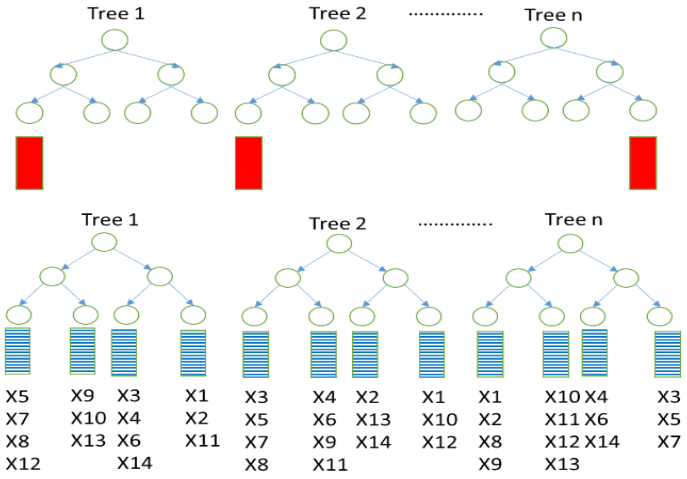


Fig. 2. (top test example assigned to a leaf in each tree) (bottom the corresponding training example assigned in each leaf)

3 First Approach: On-The-Fly Validation

5 global prediction models are trained, then for each test vector, the on-the-fly dynamic validation set based on RF neighborhood is created. We assign normalized weights for the global predictors in the ensemble by checking accuracy of each predictor on the local validation set of that test vector.

4 Second Approach: On-the-Fly Training and Validation

The RF neighborhood size is doubled, and for each test vector, random 50% of its training neighbors are used to train the 5 on-the-fly prediction models while the other 50%

is used to validate them and the validation accuracy is used to find the ensemble weights.

5 Experimental Results

The proposed approaches are tested on the monthly M3 competition dataset with a multi-step-ahead configuration (with horizons from 1 to 18 as was done in [5]). The work in [6] considered only time series that are more than 80 data points, this yields to 1045-time series. A recent study in [7], [8]compare their results with [6], since all of them uses the same 1045 datasets. The proposed approaches produce highly competitive SMAPE results compared with the state of the art results in[6], [7], [8], and with better results than the dynamic ensemble technique discussed in [[1],[3]]

References

1. Sergio AT, de Lima TPF, Ludermir TB (2016) Dynamic selection of forecast combiners. Neurocomputing 218:37–50. https://doi.org/10.1016/J.NEUCOM.2016.08.072
2. Mendes-Moreira J, Soares C, Jorge AM, Sousa JF (2012) Ensemble approaches for regression: A survey. ACM Comput Surv 45:. https://doi.org/10.1145/2379776.2379786
3. Yao C, Dai Q, Song G (2018) Several Novel Dynamic Ensemble Selection Algorithms for Time Series Prediction. Neural Process Lett 1–41. https://doi.org/10.1007/s11063-018-9957-7
4. Elshami AR, Youssef A, Fakhr MW (2018) Multi-step Ahead Time Series Prediction via Bagging Trees Based Neighborhood
5. Makridakis S, Hibon M (2000) The M3-competition: Results, conclusions and implications. Int J Forecast 16:451–476. https://doi.org/10.1016/S0169-2070(00)00057-1
6. Ahmed NK, Atiya AF, El Gayar N, El-Shishiny H (2010) An empirical comparison of machine learning models for time series forecasting. Econom Rev 29:594–621. https://doi.org/10.1080/07474938.2010.481556
7. Helmi A, Fakhr MW, Atiya AF (2018) Multi-step ahead time series forecasting via sparse coding and dictionary based techniques. Appl Soft Comput J 69:464–474. https://doi.org/10.1016/j.asoc.2018.04.017
8. Makirdakis S, Spiliotis E, Assimakopoulos V (2017) The Accuracy of Machine Learning (ML) Forecasting Methods versus Statistical Ones : Extending the Results of the M3-Competition

Assessing Wavelet Analysis for Precipitation Forecasts Using Artificial Neural Networks in Mediterranean Coast

Javier Estévez, Xiaodong Liu, Juan A. Bellido-Jiménez and Amanda P. García-Marín
University of Córdoba.Projects Engineering Area / Napier Edinburgh University. School of Computing

Abstract

Precipitation is one of the most important variables needed in different hydrological models: infiltration, soil loss, droughts, overland flow production, floods, etc. To predict its behavior is complex due to it is highly intermittent over time. Because of the adequate time-frequency representation of wavelet techniques, they are being widely applied to different hydrological resources applications. In this work, wavelet analysis has been applied in order to forecast monthly precipitation data in Mediterranean Coast (Málaga, Southern Spain) using Artificial Neural Networks (ANN) models. Several mother wavelets have been evaluated at different decomposition levels for rainfall predictions using a standard multilayer perceptron architecture. The results obtained indicate that the Daubechies wavelet transforms of order 5 (db5) used at level 3 are the most appropriate for this case study, deriving the more effective performance of all the models assessed.

A robust Hodrick-Prescott filter for smoothing high-frequency time series

Ilaria Lucrezia Amerise * and Agostino Tarsitano

Dipartimento di economia, statistica e finanza - Università della Calabria, Rende (CS), Italy

Abstract. The Hodrick-Prescott (HP) filter is a widely used but also criticized mathematical tool for removing the trend-cyclical component from time series data. Here we propose a simpler use of the filter: detecting outliers in high-frequency time series and, where necessary, replacing them with less aberrant values. The main tool is a re-formulation of HP filter in terms of absolute values, which can be very effective in constructing reliable reference time series affected by outliers from a variety of sources. The method pursues a balance between fidelity to the observed data and smoothness of the reference curve relative to which the deviations are measured. Experiments with SARMAX models show that the proposed method is a valid data preparation tool especially important in a difficult and often controversial area such as lengthy time series.

Keywords: Data preprocessing, smoothing time series, outliers treatment.

1 Introduction

The Hodrick-Prescott (HP) filter is used frequently in macroeconometrics to describe the trend-cyclical component of time series. The usual objective of such studies is to find a reference curve, which is more sensitive to long-term than to short-term fluctuations. The current paper has a different focus. Time series that consist of on-the-minute, hourly, daily or weekly observations inevitably show unexpected spikes (peaks and troughs) that appear to be grossly inconsistent with neighboring values. Since occasional large disturbances may have serious consequences for model identification, parameter estimation and prediction intervals, it is important to attenuate their adverse effects before data are used. Our aim here is to develop a robust variant of the HP filter as a tool for synthesizing in a reference curve the "deep tendencies" underlying a whole set of phenomena. The reference curve is intended to remove or reduce potentially troublesome behavior in a time series, even though, in the preliminary stage, we ignore the specific model that is eventually to be applied to the data. The relevant basic instrument is a re-formulation of HP filtering in terms of absolute values,

* Corresponding author.

In order to be operational, we assume that time series consist of a general component and other irregular aspects that are superimposed upon a reference curve.

$$p_t = \widehat{p}_t + u_t, \qquad t = 1, 2, \cdots, n \tag{1}$$

where $p_t \geq 0$ is the value observed at period t and n is the length of the time series. The values $\widehat{\mathbf{p}} = (\widehat{p}_1, \cdots, \widehat{p}_n)$ belong to the reference curve, which represents common, undisturbed growth variations of the process. The residuals u_t are assumed to have zero mean, finite variance and to be not necessarily uncorrelated.

We take this as our point of departure: the reference curve represents the fundamental components of the time series at hand. Therefore, the first statistical task in our approach is the choice of the reference values $\widehat{\mathbf{p}}$ that approximate the observed values as well as possible while, at the same time, penalizes curvilinearity of the smoothing behavior. Once the reference curve has been constructed, it can be used to establish minimally tolerable bounds that observations should not cross otherwise will be replaced by more plausible (however arbitrary) values. The tacit idea is that identification of extreme fluctuations has to be carried out before any forecast technique is implemented.

The paper is organized as follows: in Section 2, we present the least absolute deviations filtering (LADS) and show how a valid smoothing can be carried out by using linear programming. The method is fully automatic and very robust. Section 3 deals with the detection and mitigation of outliers in time series. Section 4 examines the construction of simultaneous prediction intervals derived from Box-Jenkins models. The final section discusses our findings and points out some extensions and improvements for further applications of the proposed method.

2 Robust Hodrick-Prescott filter

We obtain the reference curve by solving the following problem: given a real $0 \leq \lambda \leq 1$, find the values of $\widehat{\mathbf{p}} = (\widehat{p}_1, \cdots, \widehat{p}_n)$ that minimize the convex combination:

$$Q\left(\widehat{\mathbf{p}}, \lambda\right) = \lambda \frac{F(\widehat{\mathbf{p}})}{F_{max}} + (1 - \lambda) \frac{S(\widehat{\mathbf{p}})}{S_{max}}, \quad 0 \leq \lambda \leq 1 \tag{2}$$

with

$$F\left(\widehat{\mathbf{p}}\right) = \sum_{t=1}^{n} \left|\widehat{p}_t - p_t\right|, \quad S\left(\widehat{\mathbf{p}}\right) = \sum_{t=3}^{n} \left|\Delta^2 \widehat{p}_t\right|. \tag{3}$$

where Δ denotes the difference operator $\Delta \widehat{p}_t = \widehat{p}_t - \widehat{p}_{t-1}$. HP filtering is usually attributed to [4], but in fact it dates back to [5] building on the graduation method developed by [11]. The use of the least absolute deviations in the HP filter instead of the conventional least squares is appropriate because the former are less sensitive to big fluctuations in values (outliers) than the latter. See [9] and [1].

F_{max} in (2) denotes the maximum of $F(\widehat{\mathbf{p}})$, which occurs when all the second-order differences are equal to zero. In this case, the reference curve is determined by fitting to the observations $\mathbf{p}$ a line by the least absolute deviations. The constant S_{max} is the maximum possible value of $S(\mathbf{p})$, which occurs when the reference values are equal to observed values and hence $S_{max} = \sum_{t=3}^{n} |\Delta^2 p_t|$. The constants F_{max} and S_{max} re-scale the objective function $Q(\widehat{\mathbf{p}}, \lambda)$ to the $[0, 1]$ interval, so that λ consistently balances smoothness against the goodness-of-fit for different time series.

The rationale of (2) is the trade-off between $F(\widehat{\mathbf{p}})$ that is inversely related to goodness-of-fit and $S(\widehat{\mathbf{p}})$ that is directly related to the roughness of the reference curve. If $\lambda \to 1$, then the dominant component will be the normalized city block metric of the residuals and $\widehat{\mathbf{p}}$ will increasingly resemble the observed values more closely, no matter how irregular they may be. As $\lambda \to 0$, $\widehat{\mathbf{p}}$ approaches the line $\widehat{p}_t = b_0 + b_1 t,\ t = 1, 2, \cdots, n$ regardless of the fit component. Apart from these extreme cases, however, the solution of (2) is a serious concern because there does not appear to be an easy, efficient method to account for in an integrated manner the two conflicting components.

2.1 A cost-parametric linear programming solution

In order to simplify the solution of problem (2) we can replace the absolute differences between smoothed and observed values in the $F(.)$ component at period t, say $f_t = \widehat{p}_t - p_t$, with the sum of two non-negative deviations:

$$|f_t| = |\widehat{p}_t - p_t| = f_t^+ + f_t^-,\ f_t^+ = \begin{cases} f_t & \text{if } \widehat{p}_t \geq p_t \\ 0 & otherwise \end{cases}\ f_t^- = \begin{cases} -f_t & \text{if } \widehat{p}_t < p_t \\ 0 & otherwise \end{cases} \tag{4}$$

The same can be done for the component $S(.)$, that is, $|\Delta^2 \widehat{p}_t| = |s_t| = s_t^+ + s_t^-$. At this point, we can formulate (2) as a cost-parametric linear programming problem

$$\min_{\widehat{\mathbf{p}} \in R^n} (\mathbf{c} + \lambda \mathbf{a})^t \mathbf{x} \tag{5}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}, \qquad \mathbf{x} \geq \mathbf{0}_{2(2n-2)}, \qquad 0 \leq \lambda \leq 1. \tag{6}$$

where $\mathbf{x}^t = [(\mathbf{f}^+)^t | (\mathbf{f}^-)^t | (\mathbf{s}^+)^t | (\mathbf{s}^-)^t]$ is a $2(2n-2)$ row vector of "decision variables" and $\mathbf{c}$ and $\mathbf{a}$ are $2(2n-2)$ partitioned vectors of "costs" such that

$$\mathbf{c}^t = \left[\mathbf{w}_1^t \ |\mathbf{w}_1^t \ |\mathbf{0}_{n-2}^t \ |\mathbf{0}_{n-2}^t\right], \qquad \mathbf{a}^t = \left[-\mathbf{w}_1^t \ | -\mathbf{w}_1^t \ |\mathbf{w}_2^t \ |\mathbf{w}_2^t\right] \tag{7}$$

The symbols $\mathbf{f}^+$ and $\mathbf{f}^-$ denote $n \times 1$ vectors and $\mathbf{s}^+$, $\mathbf{s}^-$ are $(n-2) \times 1$ vectors. The weights $\mathbf{w}_1$ are given by $w_{1,t} = 1/F_{max}, t = 1, 2, \cdots, n$ and the weights $\mathbf{w}_2$ are given by $w_{2,t} = 1/S_{max}, t = 3, \cdots, n$. The symbol $\mathbf{0}_{n-2}$ represents an $(n-2) \times 1$ column vector with all components equal to zero. Finally, the matrix $\mathbf{A}$ is an $(n-2) \times 2(2n-2)$ partitioned matrix

$$\mathbf{A} = \left[\mathbf{D} | -\mathbf{D} | \mathbf{I}_{n-2} | -\mathbf{I}_{n-2}\right] \tag{8}$$

where $\mathbf{I}_{n-2}$ denotes the $(n-2)$ identity matrix and $\mathbf{D}$ is a $(n-2)\times n$ banded matrix, *i.e.* the non-zero elements are in a band centered on the main diagonal

$$d_{i,j}=\begin{cases}\binom{(-1)^{2+j-i}}{2j-i} & i=1,2,\cdots,2;\ j=i,i+1,\cdots,i+2\\ 0 & \text{otherwise}\end{cases} \tag{9}$$

The right-hand side of the equality constraints in (5) is given by $\mathbf{b}=\mathbf{Dp}$. The matrix $\mathbf{D}$ is such that the elements of the vector $\mathbf{b}$

$$\begin{pmatrix} b_i=\sum_{j=i}^{i+2}(-1)^{2+j-i} \\ 2j-ip_j \quad i=1,2,\cdots,n-2\end{pmatrix} \tag{10}$$

are the second differences of the observed values. The matrix $\mathbf{A}$ is assumed to be of full row rank. Smoothed values are obtained from the decision variables of the optimal solution as: $\widehat{p}_t=p_t+(f_t^+-f_t^-)$. [7] show that the set of admissible values of λ can be partitioned into a finite number ν of subintervals.

$$\widehat{p}_t\,(\lambda)=\begin{cases}p_t, \quad t=1,\cdots,n \text{ if } \lambda=\lambda_0=0\\ (1-\lambda_i)\,\mathbf{w}_1^t\,(\mathbf{f}^+ + \mathbf{f}^-)+\lambda_i\mathbf{w}_2^t\,(\mathbf{s}^+ + \mathbf{s}^-) \text{ if } \lambda\in[\lambda_{i-1},\lambda_i), i=1,\cdots,\nu\\ \sum_{j=0}^{1} b_j t^j,\ t=1,\cdots,n \text{ if } \lambda=\lambda_\nu=1\end{cases} \tag{11}$$

The central relationship in (11) means that an optimal basic index set for some fixed value of λ would remain optimal for a range of λ. The parametric linear programming procedure is not difficult to implement (see, for example, [12]).

2.2 Choice of the smoothing constant

The choice of λ is as important as it is arbitrary. One way to proceed is to solve problem (5) for various values for a fixed set of values such as $\lambda\in L=(0.01,0.05,0.10,\cdots,0.90,0.95,0.99)$ and then deciding which λ value constitutes a good choice on the basis of visual comparisons. It can be shown that $Q[\widehat{\mathbf{p}}(\lambda)]$ is a positive and concave function of λ.

In Figure 1, we report the relationship between λ and $Q[\widehat{\mathbf{p}}(\lambda)]$, which has been obtained by evaluating (5) for each λ in L. The example we consider consists of the income of farms in Australia (m.) for each financial year from 1948/49 to 1957/58 reported in [5] for a total of $n=10$ observations. The series has been chosen because of its large fluctuations.

The curves reveal an inverted U-shaped relationship between the minimum of $Q(\widehat{\mathbf{p}}_{m,\lambda})$ and λ for each order of differencing. This behavior indicates that, as λ growths, the minimum of $Q(\widehat{\mathbf{p}}_{m,\lambda})$ increases, and after reaching a turning point, will diminish. As an operative strategy, we consider the element of the grid where the turning point is located as the best λ because, here, smoothness turns from a positive into a negative effect on goodness-of-fit. Thus, in the case in example, a value near $\lambda=0.4$ seems appropriate. Needless to say, it is a rather cumbersome and expensive way of finding the smoothing constant, but it has the merit of being completely automatic.

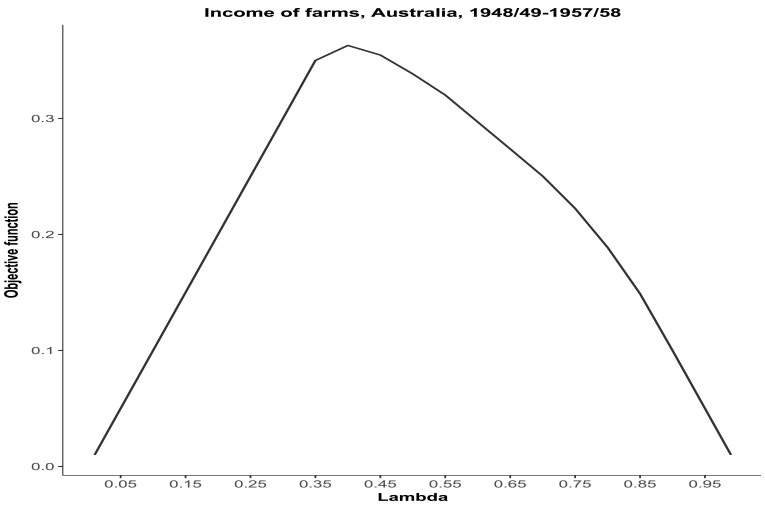


Fig. 1. Relationship between the loss function and the smoothing parameter.

Figure 2 illustrates the time series, the robust HP (with ($\lambda = 0.4$) and the ordinary HP filter with a smoothing coefficient of $\lambda = 1.01$ (in the unnormalized scale).

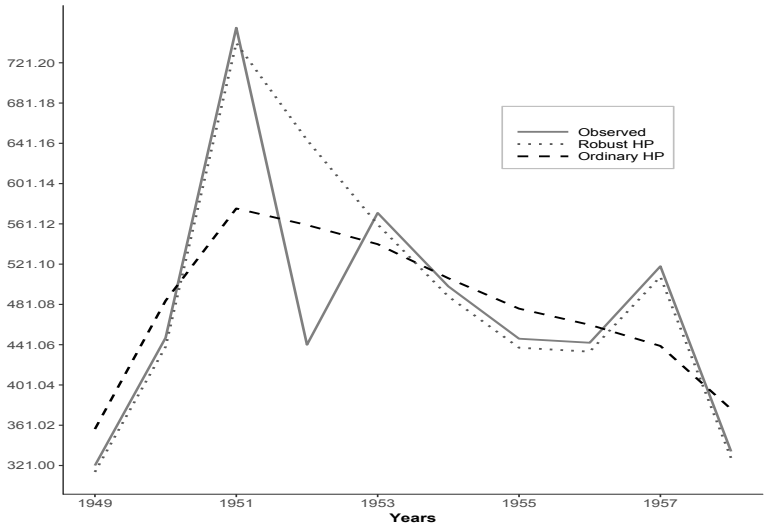


Fig. 2. Observed and interpolated income of farms, Australia

[5] compares the quasi-linear trend obtained with the ordinary HP filter to a polynomial trends of degree $2, 3, 4$ and concludes that the quasi-linear trend seems the more suitable one, as it is more successful in smoothing out the fluctuations in the data. The robust HP filter overlaps the observed time series, with the exception of year 1951 where a deep valley is ignored. All the other peaks are preserved.

3 Dealing with outliers

The method proposed in the preceding section can be particularly useful in the treatment of outliers when there is no a priori information that can be used to identify and remove unreliable measurements. A reasonable strategy to detect outliers is the analysis of the difference between original and reference values $\widehat{u}_t = \widehat{p}_t - p_t$ $t = 1, 2, \cdots, n$ by looking for points that are poorly interpolated by the reference curve.

In this regard, it is necessary to define lower and upper thresholds for $|\widehat{u}_t|$ which delimit the regions of tolerable variations. A natural choice is

$$\tilde{\mu} - K\tilde{\sigma} < \ \widehat{u}_t \ < \tilde{\mu} + K\tilde{\sigma} \qquad t = 1, 2, \cdots, n \tag{12}$$

where $K > 0$ is a positive multiplier, $\tilde{\mu}$ is a robust measure of central tendency and $\tilde{\sigma}$ is a robust measure of dispersion (robustness is required because the mean and the variance are vulnerable to the influence of outliers).

In our applications, we use the Sen rank weighted mean

$$\tilde{\mu} = \left[\binom{\nu}{2j+1}\right]^{-1} \sum_{i=1}^{\nu} \binom{i-1}{j}\binom{\nu-i}{j}\widehat{u}_{(i)} \tag{13}$$

where $\widehat{a}_{(i)}$ is the i-order statistic with $0 < j < (\nu–1)/2$. See [10]. Our choice is $j = 2$ if $\nu > 5$, otherwise $\tilde{\mu} = median(|\widehat{u}_t|), |u|_t > 0$. The integer $\nu \leq n$ is the number of residuals u_t that are different from zero. This restriction is necessary because the residuals arising from the solution to the linear programming problem discussed in Section 2.1 contains a certain number of zero values, which if fully included in the computation of the two statistics, would reduce their robustness.

The statistic used as a robust scale estimator is the first quartile of the sorted pair-wise differences between all residuals.

$$\tilde{\sigma} = 2.21914 \left\{\left||\widehat{u}_i| - |\widehat{u}_j|\right|; i < j, |\widehat{u}_i|, |\widehat{u}_j > 0\right\}_{(q)} \tag{14}$$

where $q = \binom{n}{2}/4$. See [6].

The factor K appearing in (12) establishes the conservative/liberal attitude of the filter in rejecting outliers. Liberal choices (large values) of K effectively turn the filter off since no modifications are, in fact, suggested. Conversely, conservative choices (small values) of K can lead to the refusal of most of the observations. We have adopted the traditionally four-sigma rule, that is, $K = 4$, which, in a random sample from a standard Gaussian distribution, would reject the 0.0063% of the units. This means that only 6 observations out of $100'000$, may be expected to lie beyond a distance of $\pm 4\sigma_p$ from the reference curve.

3.1 Replacement of outliers

If a residual $\widehat{u}_t$ surpasses one of the warning limits, then the corresponding value is considered an outlier. This does not imply that the suspicious value should be automatically eliminated. The sharp decision of whether to keep or reject an observation is, to some degree, wasteful. While the removal of aberrant values may improve the performance of fitting and forecasting models, it may end up suppressing some important feature of the time series. If too many of them are deleted and imputed, for example, using an average of the remaining data, the forecasting technique may be adapted to an unrealistic time series without sufficient information about peaks and valleys. We argue that, it would be better to down-weight dubious observations rather than reject them.

Indeed, when considered the relevance of the spikes in time series, we do not want to drastically smooth out such maxima. On the other hand, local minima should be, at least partially, preserved because they could represent particular conditions that need to be accounted for in time series setting. Thus, we propose replacing of suspect outliers with an average of actual and reference values

$$\tilde{p}_t = \gamma p_t + (1-\gamma)\,\widehat{p}_t, \quad \text{with} \qquad 0 < \gamma < 1 \tag{15}$$

where t runs over all the periods with values falling outside the bounds (12). The greater is γ, the closer is the averaged value $\tilde{p}_t$ to the observed outlier p_t and the smaller is the contribution of the reference value $\widehat{p}_t$. As γ decreases, the strategy (15) yields average values which lie more and more closer to the reference time series, thus strengthening the role of the smoothing procedure.

The empirical work undertaken so far has been rather limited. While awaiting additional studies, we suggest $\gamma = 0.25$, which enables the smoothed time series to maintain the shape (if not the magnitude) of maxima (with surrounding peaks) and minima (with surrounding valleys).

4 Effectiveness of the robust HP filter

There are various indicators that can be used to quantify the impact of outlier cleaning and hence of the HP filter. In this section, we assess the robustness of HP method by analyzing its effects on the accuracy of simultaneous prediction intervals (PIs) derived from Box-Jenkins seasonal models.

We will evaluate the consequences of leaving outliers in the data by splitting a time series into two parts: the "training" period, which ignores a number of the most recent time points, and the "validation" period, which is comprised only the ignored time points and constitutes a separate part of the time series. The training period is used to identify and estimate the model. The validation period is used to test the smoothing procedure.

We assume that the time series $p_t, t = 1, 2, \cdots, n$ is adequately represented by a multiplicative seasonal autoregressive moving average with external regressors

process (SARMAX)

$$p_t - \left(\beta_0 + \sum_{j=1}^{m} \beta_j X_{t,j} \right) = [\phi^*(B)]^{-1} \theta^*(B)\, a_t \tag{16}$$

where $a_t, t = 1, 2, \cdots$, are independent and identically distributed random variables with mean zero and finite variance σ_a^2, B is the backward shift operator and $\phi^*(B)$ and $\theta^*(B)$ are defined as

$$\begin{cases} \phi^*(B) & = 1 - \phi_1^* B - \phi_2^* B^2 - \cdots - \phi_{p^*}^* B^{p^*} \\ \theta^*(B) & = 1 - \theta_1^* B - \theta_2^* B^2 - \cdots - \theta_{q^*}^* B^{q^*} \end{cases} \tag{17}$$

where p^* and q^* are the orders of the AR and MA polynomials, respectively. For stationarity and invertibility, it is assumed that the roots of $\phi^*(B)$ and $\theta^*(B)$ lie outside the unit circle, with no single root common to both polynomials. The $X_{t,j}, j = 1, 2, \cdots, k$ are k variables observed on day t influencing the dependent variables; β_j is a parameter measuring how the price p_t is related to the j-th variable $X_{t,j}$. In order to keep the estimation problem tractable, the exogenous variables are all deterministic functions of time, *e.g.* calendar variables or orthogonal polynomials in time. Of course, in the case of binary variables one of the categories must be omitted to prevent complete collinearity.

4.1 Simultaneous prediction intervals

In order to assess the effectiveness of our robust smoother, we compare the capacity of prediction intervals (PIs) to contain all H future values, both in the presence and absence of smoothing.

The scope of prediction intervals is to determine two confidence bands

$$P\left[\bigcap_{l=1}^{H} \left(C_{l,\alpha}^1 \leq p_{n+l} \leq C_{l,\alpha}^2 \right) \right] = 1 - \alpha \ . \tag{18}$$

such that the probability of consecutive future values $p_{n+l}, l = 1, 2, \cdots, H$ lying simultaneously within their respective range is $(1-\alpha)$. The limits in (18) are $C_{l,\alpha}^1 = p_{n,l} - c_\alpha \sigma_l$ and $C_{l,\alpha}^2 = p_{n,l} + c_\alpha \sigma_l$. The quantity σ_l^2 is the residual variance at the l-th lead time

$$\sigma_l^2 = \sigma_a^2 \sum_{l=0}^{l-1} \psi_l^2 \qquad l = 1, 2, \cdots, H \ . \tag{19}$$

c_α is the α-th quantile of the joint probability distribution of forecast errors

$$G^{-1}(c_\alpha) = 1 - \alpha \qquad \text{with } G(c_\alpha) = Pr\left(|z_l| \leq c_\alpha,\ l = 1, 2, \cdots, L \right) \tag{20}$$

where $z_l = e_{n,l} / \sqrt{\sigma_l^2}, l = 1, \cdots, H$ are the standardized forecast errors. The computation of c_α requires an explicit hypothesis about the distribution of the

forecast errors. More specifically, we assume that G is the H-variate Gaussian distribution.

$$Pr\left(|z_l| \leq c_\alpha, l{=}1, \cdots, H\right){=}\int_{-c_\alpha}^{+c_\alpha} \cdots \int_{-c_\alpha}^{+c_\alpha} f\left(z_1, \cdots, z_L\right) dz_1 \cdots dz_H. \quad (21)$$

Simultaneous PIs guarantee that the H individual intervals include the respective expected price with a confidence level of $(1{-}\alpha)$. If a_t is a Gaussian process and the ϕ, θ and σ_a^2 coefficients are known, then $(z_1, \cdots, z_H)$ have an H-variate Gaussian distribution with mean vector $\mathbf{0}_H$ and correlation matrix and correlation matrix

$$\boldsymbol{\Sigma} = (\rho_{i,j}) = \frac{\sum_{l=0}^{i-1} \psi_j \psi_{j-i+l}}{\sqrt{\sum_{l=0}^{i-1} \psi_l^2} \sqrt{\sum_{l=0}^{j-1} \psi_l^2}} \qquad i < j \ . \quad (22)$$

where $\psi_i, i = 0, 1, \cdots, n$ are the parameters of the infinite moving-average representation of the process. If the errors are independent and identically distributed, then $\boldsymbol{\Sigma} = \sigma_a^2 \mathbf{I}_H$. Only in this case would be legitimate using marginal PIs

$$p_{n,l} \pm z_{\alpha/2}\sigma_l, \qquad l = 1, \cdots, H. \quad (23)$$

where $z_{\alpha/2}$ is the upper α-th quantile of the univariate standard Gaussian distribution. However, the hypothesis of independent or even uncorrelated forecast errors is illusory and has no validity in practical situations. Hence, unless the observed values $p_{n+l}, l = 1, \cdots, H$ develop according to a known pattern, the probability that a given sequence lies completely inside all H marginal PIs would be less than $100(1{-}\alpha)$, especially if H is large. This is the reason why we have focused our efforts on simultaneous PIs.

4.2 Evaluation of PIs

The most important characteristic of PIs is their actual coverage probability (PIAC). We measure PIAC by the proportion of true values of the validation period enclosed in the bounds

$$PIAC_\alpha = 100H^{-1} \sum_{l=1}^{H} c_{l,\alpha} \quad \text{where } c_{l,\alpha} = \begin{cases} 1 & \text{if } p_{n+k} \in \left[C_{l,\alpha}^1, \quad C_{l,\alpha}^2\right] \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

If $PIAC_\alpha \geq (1{-}\alpha)$ then future values tend to be covered by the constructed bands, but this may also imply that the estimates of the variances in the forecast errors are positively biased. A $PIAC_\alpha < (1{-}\alpha)$ indicates under-dispersed forecast errors with overly narrow prediction intervals and unsatisfactory coverage behavior. All other things being equal, narrow PIs are desirable as they reduce the uncertainty associated with forecast-based decision-making. However, high accuracy can be easily obtained by widening PIs. A complementary measure that quantifies the sharpness of PIs might be useful in this context. Here, we use the score function.

$$ASW_\alpha{=}\frac{1}{H}\sum_{l=1}^{L} R_{l,\alpha} \quad \text{with } R_{l,\alpha}{=}\left(\frac{1{-}\alpha}{2}\right)\frac{\left(C_{l,\alpha}^2 - C_{l,\alpha}^1\right)}{p_{n+l}}, \quad l{=}1, \cdots, H. \quad (25)$$

This expression reflects a penalty proportional to the narrowness of the intervals. The penalty increases as α decreases, to compensate for the tendency of prediction bands to be broader as the confidence level increases. Of course, the lower ASW_α is, the more accurate PI will be.

4.3 Empirical Analysis

A remarkable characteristic of hourly time series of electricity prices is the presence of price spikes. The frequency of spikes changes chaotically over time and the their amplitude can vary by up to several standard deviations from standard prices. Under these conditions, selecting or developing an appropriate model for forecasting may be much more challenging than in other settings. In this section, we perform an experimental evaluation of our method. We prefer using real data rather than conjecture of what should be a model of the system. In doing so, we intend to preclude one method from having an unfair advantage merely because the data used for the comparison are generated under the same process on which the method is based.

The purpose of this section is to examine $144 = 24 \times 6$ hourly time series of prices, one for each hour of the day and each zone of the Italian electricity market. Due to transmission capacity constraints, Italy is partitioned into six zones: North, Center-North, Center-South, South, Sardinia and Sicily with a separate price for each zone. When there is no transmission congestion, arbitrage opportunities force the prices in each zone to be equal. All the time series are long $1'976$ days, but the last three weeks ($H = 21$) are reserved for assessing the accuracy of PIs.

Parameters are estimated by optimizing the log-likelihood function of (16), provided that $p,\ q,\ P,\ Q$ are known and errors are Gaussian random variables. Since we ignore the order of the polynomials, the estimation is repeated for different values of p, q, P and Q. The search of the best SARMAX model is carried out within the bounds $0 \leq p, q, P, Q \leq 3$ which include 256 distinct processes to be explored for each time series. The search for the best model is carried out in non-stepwise automatic mode using the *auto.arima* function of the R package $forecast$ with parameters constrained to be stationary. It should be pointed out that PIs tend to perform poorly when the residuals are not Gaussian. In consequence, even under the most favorable conditions, the PIs in (18) are approximate PIs.

To compute (21), we apply the method proposed by [3]. Table 1 shows the results at the confidence levels $(80, 85, 90, 95)$. Columns labeled "none" display the actual prediction interval coverage rate (PIAC) and the average width (ASW) when time series have not undergone a pre-processing stage. Columns labeled "RHP", display the analogous results obtained after applying the robust Hodrick-Prescott filter with $K = 4$ and $\gamma = 0.25$. In the initial general examination, we note the consistency of the behavior of PIAC and ASW with the latter decreasing as the former increases, for each zone, either in the presence or absence of filtering. Naturally, this is a confirmation of the expected behavior

of the score function that measures the PI performances. What appears immediately clear is the notable difference between the various zones, reflecting the climatic diversity of the zones and price differentials. It is no coincidence that the most negatively affected zones are the problematic large islands of Sicily and Sardinia, which suffer from poor interconnections and frequent congestion.

Table 1. Improvement in the accuracy of prediction intervals.

		$PIAC_\alpha$		ASW_α	
Zone	$(1-\alpha)\%$	None	RHP	None	RHP
1	80	87.66	81.56	13.65	9.13
	85	91.41	86.13	11.29	8.85
	90	94.37	92.47	8.42	5.91
	95	96.93	95.24	4.87	2.83
2	80	89.83	82.08	15.33	10.72
	85	92.00	86.12	12.61	8.53
	90	95.15	91.19	9.43	7.11
	95	97.92	96.16	5.45	3.64
3	80	93.58	82.32	17.73	11.18
	85	94.96	89.01	14.63	8.97
	90	96.73	93.27	10.87	6.11
	95	98.32	97.67	6.35	3.82
4	80	95.55	81.82	17.48	10.56
	85	96.35	87.05	14.54	9.17
	90	97.13	93.24	10.84	6.16
	95	98.12	96.05	6.27	4.51
5	80	99.10	83.92	28.26	22.27
	85	99.10	87.83	23.37	20.80
	90	99.10	93.78	17.39	14.72
	95	99.10	98.54	10.11	8.75
6	80	98.71	84.22	22.85	19.44
	85	98.71	87.91	18.91	17.85
	90	98.90	94.01	14.04	12.16
	95	98.90	98.71	8.09	6.92

To have an idea of the effects of RHP in reducing the impact of price spikes on forecasting, we compare the narrowness of the prediction intervals reported in the columns headed ASW_α at the first level. Figures shown in column "RHP" are systematically and significantly lower than those shown in column "none". Additionally, more precise forecasts are obtained without appreciably reducing the coverage rate. The performances of SARMAX models, combined with the RHP, appear to be moderately satisfactory with respect of the improved accuracy and efficiency of the prediction intervals. The main result is that, in the absence of smoothing, SARMAX models consistently yield PIs with greater than nominal coverage rates.

The robust smoother corrects the coverage rates, but not in a way to alter the impression of over-dispersed forecast errors. This is an unwanted conservatism,

primarily due to inflated estimates of the forecast error variances, which, in turn, can be attributed either to unsuspected behavior of the time series in the validation period, or to the length of the forecast horizon or, ultimately, to the weakness of the usual Box-Jenkins approach, when applied to electricity price time series.

5 Concluding remarks

The robust HP filter is not only effective in favoring optimal conditions for the application of SARMAX models, but also have neutral or inhibiting effects on outliers when the tuning constants are appropriately chosen. Naturally, we do not claim that it achieves the best, or even a satisfactory, result under all circumstances, or even under most. However, it does have the advantage that it reduces the width of simultaneous prediction intervals deriving from SARMAX models while keeping the coverage rate close to the nominal level. As such, our robust filter adds a very promising methodology to the data analysis toolbox within the area of time series analysis.

References

1. Chan, F. Y., Chan, L. K., Yu, M. H. A generalization of Whittaker-Henderson graduation. Trans. of Soc. Act., 36, 183–211 (1984)
2. Cogley, T., Nason, J. M. Effects of the Hodrick-Prescott filter on trend and difference stationary time series. Implications for business cycle research. J. Econ. Dyn. Control, Vol. 19, 253–278 (1995)
3. Genz, A. Numerical computation of multivariate normal probabilities. J. Comput. Graph. Stat., 1, 141–149 (1992)
4. Hodrick, R. J., Prescott, E.C. Postwar U.S. business cycles: An empirical investigation. Discussion Paper No. 451, Department of Economics, Carnegie Mellon University, 1980
5. Leser, C. E. V. A simple method of trend construction. J. R. S. S. B, 23, 91–107 (1961)
6. Rousseeuw, P.J., Croux, C. Alternatives to the median absolute deviation. J. Am. Stat. Assoc., 88, 1273–1283 (1993)
7. Saaty, T., Gass, S. Objective function (part 1). J. Oper. Res. Soc., 2, 316–319 (1954)
8. Schlicht, E. Estimating the smoothing parameter in the so-called Hodrick-Prescott filter. J. Japan Statist. Soc. 35, 99–119 (2005)
9. Schuette, D. A. A linear programming approach to graduation. Trans. Soc. Act., 30, 407–431 (1978)
10. Sen, P. K. On some properties of the rank-weighted means. J. Ind. Soc. Agr. Stratist., 16, 5–61 (1964)
11. Whittaker, E. T. On a new method of graduation. Proc. Edin. Math. Soc., 41, 63–75 (1922)
12. Yao, Y., Lee, Y. Another look at linear programming for feature selection via methods of regularization. Stat. Comput. 24, 885–905 (2014)

Big-Learn 2.5: Using Lucidworks and SolrJ to Improve Online Search in Big Data Environment

K. AOULAD ABDELOUARIT[1,] B. SBIHI[2] and N. AKNIN[3]

[1, 2, 3] TIMS Research Unit, LIROSA Laboratory
Abdelmalek Essaadi University
Tetuan, Morocco
[1] abdelouarit.karim@gmail.com
[2] bsbihi@hotmail.com
[3] noura.aknin@uae.ac.ma

Abstract. In this article, we present an implementation of the Big-Learn system in its version 2.5. It is a Big Data solution for online search, used by a learner in a context of distance learning or information searching. The main objective of this solution is to process massive data and evaluate the reliability and quality of the information returned by the online search system, in order to improve the learning process and thus allow a reliable consumption of data. The version 2.5 of the Big-Learn tool is based on the combination of two systems: Lucidworks and SolrJ. Thus, the massive data generated by the Big Data layer will be processed at the Spark technique of the Lucidworks tool, in addition to their indexing by the Lucene engine and finally their exploitation by the SolrJ tool from the Solr interface.

Keywords: Big Data, e-Learning, Online Search, Lucidworks, Lucene, SolrJ, Spark, Solr.

1 Introduction

The Big Data phenomenon made data on the Internet difficult to manage with traditional database or information management tools [1]. Indeed, the current online search engines do not provide adequate and consistent data results with the expectations of its users and especially the learners who use the e-learning platform. They are left with a heterogeneous set of data that they cannot consume or process for educational purposes and in particular in distance education or online search [2].

This article follows our previous work on the study of a complete solution for the processing of massive and heterogeneous data returned by online search engines [3]. Our proposed solution is based on a tool called "Big-Learn" to integrate several types of massive data into a data layer to facilitate access and optimal search with adequate and consistent results according to the expectations of the learner, the user of the online search [4]. Thus, the formula of the solution chosen in version 1 is to use the Hadoop-MapReduce technology to store and process the massive data generated from the Big Data layer [5]. This data is then indexed by the Lucene engine. At the end, this indexed data is easily accessible via a flexible search interface using the Solr

Framework [6]. The version 2 of the solution has replaced the Hadoop layer with the use of Apache Spark technology given its performance and its real-time and more efficient processing of massive data.

The purpose of this document is to present the Big-Learn system in its new version 2.5, which consists in using a combination of two systems to improve the processing and storage of massive data as part of the online search used in an e-learning environment used on massive data.

The following paragraph is a reminder of version 2.0 of the Big-Learn system based on Spark and Solr for the online search process used by the learner in the Big Data environment. Thus, we describe the application and technical architecture in this version. Then we expose the new version 2.5 of the proposed Big-Learn system with its technical architecture based on the two technologies Lucidworks and SolrJ. Section 3 provides an example of how to use this combination of solutions and how to integrate with our Big-Learn system [4]. The last paragraph presents a general conclusion describing a series of perspectives.

2 The Big-Learn 2.0

2.1 The application architecture of the Big-Learn Model

The Big-Learn system offers a complete solution to the problem of processing massive data used by a learner in an e-Learning environment for its information or documentation needs [3]. It is a complete and effective model to processes massive data generated by the Big Data phenomenon. This data is subsequently indexed and presented to the user [4].

Based on our previous study [4], we have discovered the version 2.0 of the Big-Learn system, with its different components, namely the Spark system for the massive data processing with the MapReduce technique as well as the storage of this data via the HDFS system. We also saw the Lucene component for data indexing, as well as the Solr Framework for data searching and exploring. The Big-Learn 2.0 solution today offers a complete online search solution for learners in the Big Data environment and supports the processing of the heterogeneous (structured and unstructured) data [4].

Figure 1 below shows the application architecture of the Big-Learn system with its different processing layers.

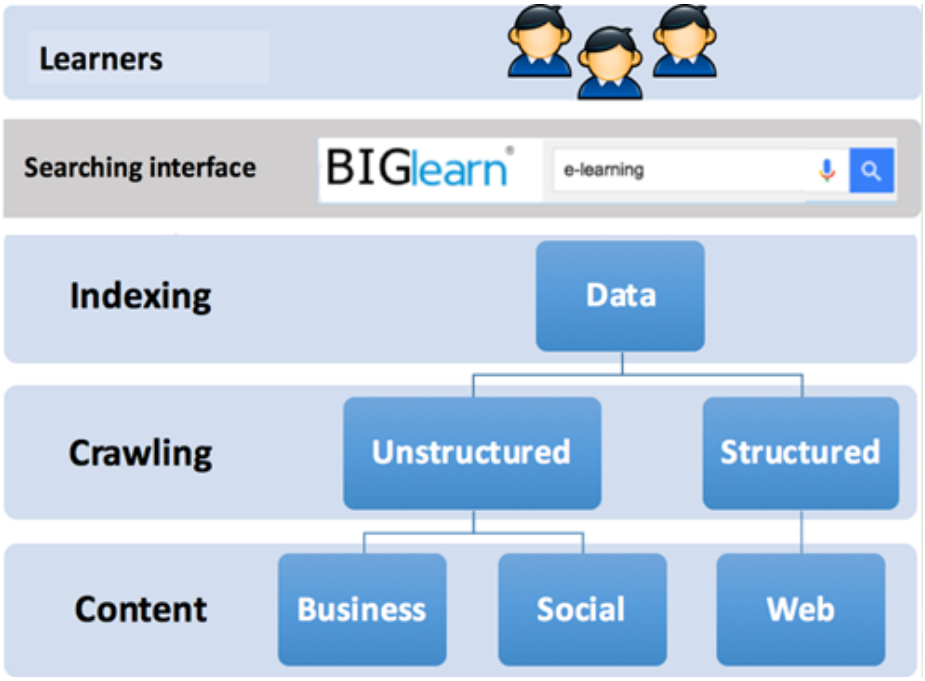


Fig. 1. The application architecture of the Big-Learn system.

As schematized in this figure, the learner connects to the interface of the online search engine to launch an information search request. Then the engine will search the data on the Internet. These data are based on different sources (social networks, web, commercial data, etc.) and in several formats (structured, semi-structured or unstructured). Thus, the indexing engine data to be presented as results at the interface of the plate-form.

These results data are in the form of pages containing links that redirect to the result found. In addition, the results page includes a large amount of representative data in several forms of information [4].

2.2 The technical architecture of Big-Learn 2.0 system

The integral architecture of the Big-Learn 2.0 solution to be adopted in order to meet our needs to improve the process of online research in the Big Data environment is to combine Apache Spark technology for massive and heterogeneous data storage and processing, with the Solr Framework for data mining and search, based in turn on the Lucene indexing engine. Figure 2 defines the overall architecture of this solution integrating the different layers of each technique and its role in the integrated Big-Learn 2.0 system.

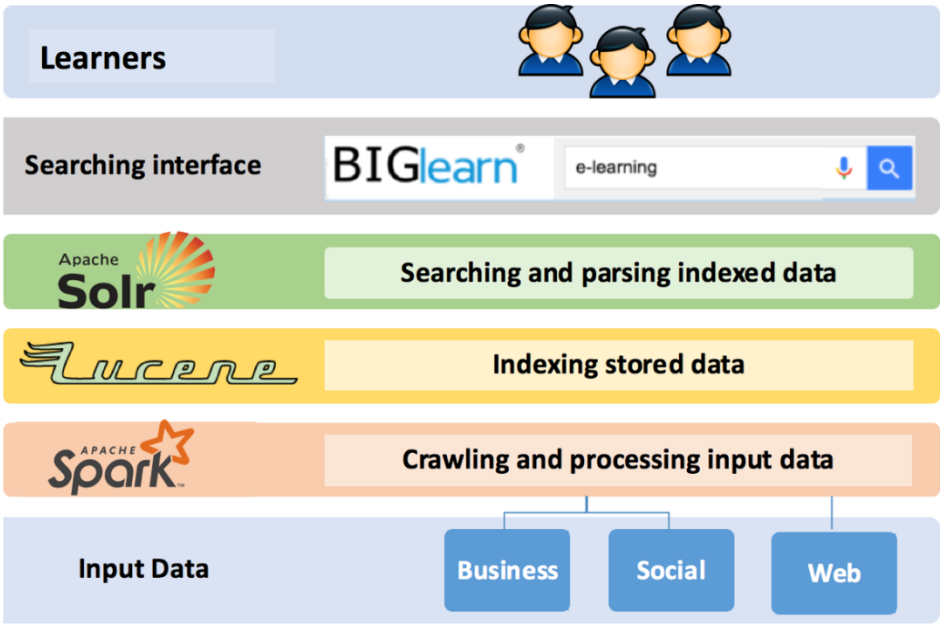


Fig. 2. The technical architecture of the Big-Learn 2.0 system.

As outlined in this figure, the learner accesses the Big-Learn interface to launch his search request for his information or documentation needs. Thus, the system searches the data on the Internet. These data come from different sources (social networks, websites, business data, etc.) and are manifested in different forms (structured, semi-structured or unstructured). These data are intercepted by the Spark system, which will store them in the Hadoop distributed file system (HDFS) and then process them using the MapReduce technique. Thus, the data is loaded and processed during the Map() phase, and then combined and stored during the Reduce() phase to create the index with Lucene. The Lucene indexing engine reads the stored HDFS data and stores them using the Lucene schema, which in turn stores the data as documents in the Lucene index. Once all the files are indexed at the Lucene layer, it is now possible to perform queries and explore this data via the Solr interface which will be responsible for organizing and presenting this data on the page of the user interface as a result [6].

3 Big-Learn 2.5: using Lucidworks and SolrJ to improve online search in an e-Learning environment

3.1 Benchmarking of Big Data technologies

The available search tools on the market do not provide an efficient and complete solution to process information search in a massive and heterogeneous environment.

This is usually due to the absence of Big Data technologies that allow to integrate various types of data and to provide a unified data view to the user of online search. Table 1 presents a benchmarking of Big Data solutions.

Table 1. Big Data Technologies Comparison.

Technology	Description
Hadoop	It is an open source project based on Google for processing large sets of structured and unstructured data. It is a Framework that enables large-scale data analysis and uses the MapReduce algorithm for data processing. Hadoop provides the HDFS file system for distributed storage of data, and a Java API that allows parallel processing of data on the cluster nodes.
Spark	It is a Big Data processing framework built to perform sophisticated analysis and designed for speed and ease of use, it exceeds Hadoop in the fast processing of large data, and also in the real-time analysis. It relies on a programming model similar to Hadoop's MapReduce but extends with an abstraction of data sharing called RDD (Resilient Distributed Datasets) [8]. Using this extension, Spark can capture a wide range of processing loads that previously needed separate engines including SQL, streaming, machine learning and graphics processing.
Lucene	It is an open source Apache project that offers a text search engine library. It includes indexing, ranked searching, powerful query types like phrase queries, wildcard queries, proximity queries, range queries, fielded searching such as title, author, contents.
Solr	It is an open source search platform based on Apache Lucene project. It includes full text search. It uses the Lucene as search library for full-text indexing and search. It provides distributed indexing, replication and load-balanced querying, automated failover and recovery, centralized configuration.

According to this comparative table of solutions, we can say that to meet our need for improvement of online research in the Big Data environment, the best solution is to combine Spark technology for the storage and processing of massive and heterogeneous data, with a search framework as Solr, in addition to an indexing engine data like Lucene. Moreover, a specific development is needed in terms of this system presentation layer to meet the ergonomic needs and formatting of search results returned by the online search system.

3.2 The new Big-Learn 2.5 Model

The Big-Learn system for online search in Big Data environment, as already presented in our previous work [4] intercepts data from the Big Data layer via the Spark system which stores them in its HDFS storage system after their processing through its

MapReduce technique. The Lucene layer reads the data stored in HDFS, and indexes them using Lucene Scheme. Once all files are indexed to the Lucene layer, their queries are possible via the Solr layer. However, based on our study of the new Big-Learn model, the new architecture of the system after integrating Lucidworks (the apache spark-solr project) can be shown in the Figure 3.

Fig. 3. Integrating Lucidworks and SolrJ in the Big-Learn system.

As shown in this figure of the new architecture, data coming from the Big Data will be intercepted and processed by the Apache Spark system and will always use the HDFS system for storing processed data. These data will be indexed from Spark into Solr using SolrJ so that they can be interrogated via the Solr interface.

4 Using Lucidworks and SolrJ

4.1 Installation and configuration

This section includes tools for reading data from Solr as a Spark RDD and indexing objects from Spark into Solr using SolrJ. It can be processed on 3 steps:

a) <u>Import jar File via spark-shell:</u>

i. Version Compatibility

The spark-solr project has several releases, each of which support different versions of Spark and Solr. The compatibility chosen below shows the versions of spark and solr tools supported across the 'Connector' that refers to the 'spark-solr' library:

Table 2. The spark-solr connector version compatibility.

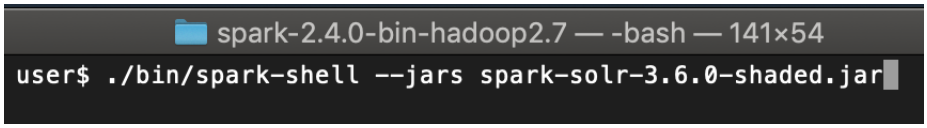
Connector	Spark	Solr
3.6.0	2.4.0.	7.5.0

ii. Importing the shaded jar file

The shaded jar "spark-solr-3.5.5-shaded.jar" file can be downloaded from the Maven Central or built from the respective branch (package: com.lucidworks.spark:spark-solr) at this URL:

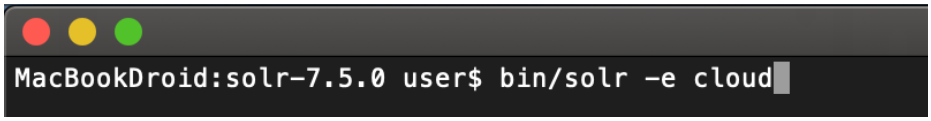
https://search.maven.org/artifact/com.lucidworks.spark/spark-solr/3.6.0/jar

Now, we can import the shaded jar file "spark-solr-3.5.5-shaded.jar" after running this command-line:



b) Connect to the SolrCloud Instance:

The bin/solr script makes it easy to get started with SolrCloud as it walks you through the process of launching Solr nodes in cloud mode and adding a collection. To get started, simply do:



This starts an interactive session to walk you through the steps of setting up a simple SolrCloud cluster with embedded ZooKeeper.

The script starts by asking you how many Solr nodes you want to run in your local cluster, with the default being 2.

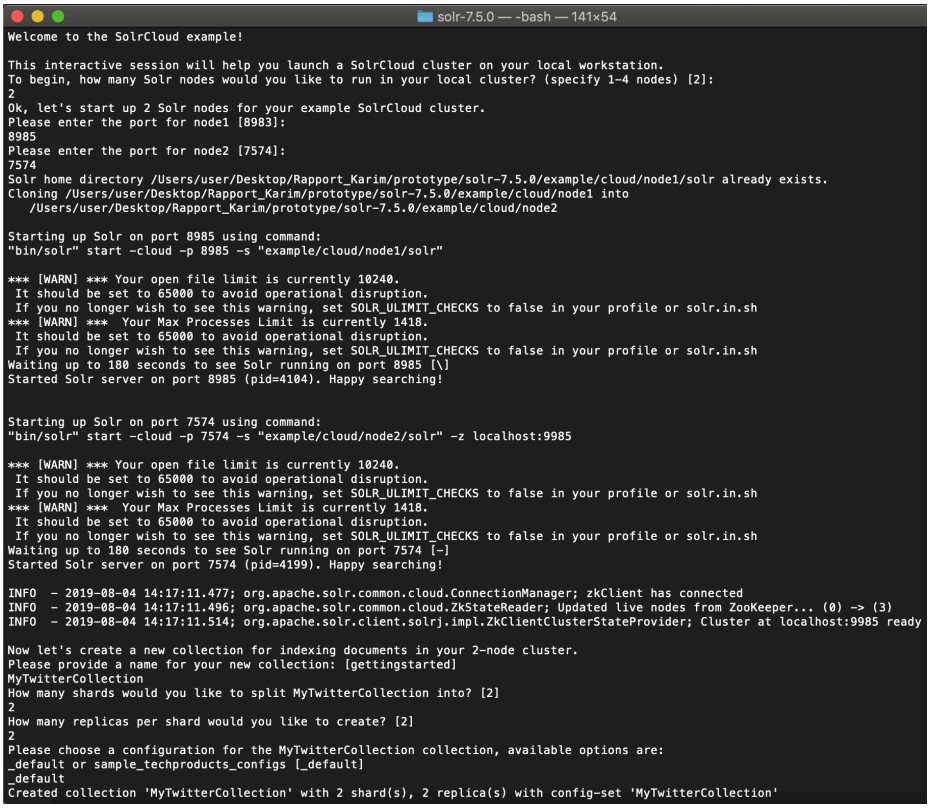


Fig. 4. Creating the "MyTwitterCollection" SolrCloud instance.

We can connect to the SolrCloud Instance using different ways: DataFrame, RDD or RDD (Java). The code scripts below presents the set of these methods:

i. Connecting via DataFrame

```
val options = Map(
  "collection" -> "{solr_collection_name}",
  "zkhost" -> "{zk_connect_string}"
)
val df = spark.read.format("solr")
  .options(options).load
```

ii. Connecting via RDD

```
import com.lucidworks.spark.rdd.SelectSolrRDD
val solrRDD = new SelectSolrRDD(zkHost, collectionName, sc)
```

The "SelectSolrRDD" is an RDD of SolrDocument.

iii. Connecting via RDD (Java)

```
import com.lucidworks.spark.rdd.SolrJavaRDD;
import org.apache.spark.api.java.JavaRDD;
SolrJavaRDD solrRDD = SolrJavaRDD.get(zkHost, collection, jsc.sc());
JavaRDD<SolrDocument> resultsRDD = solrRDD.queryShards(solrQuery);
```

c) Download/Build the jar Files:

i. Maven Central

The released jar files (1.1.2, 2.0.0, etc..) can be downloaded from the Maven Central repository. Maven Central also holds the shaded, sources, and javadoc .jars for each release. We need to add these lines in our maven dependencies file:

```
<dependency>
   <groupId>com.lucidworks.spark</groupId>
   <artifactId>spark-solr</artifactId>
   <version>{latestVersion}</version>
</dependency>
```

ii. Build from Source

We can also build the jars from the source with this command line:

```
mvn clean package -DskipTests
```

This will build 2 jars in the target directory:

- spark-solr-${VERSION}.jar
- spark-solr-${VERSION}-shaded.jar

${VERSION} will be something like 3.6.0-SNAPSHOT, for development builds.

The first .jar is what we want to use if you were using spark-solr in your own project. The second is what we use to submit one of the included example apps to Spark.

4.2 Implementation and manipulation

This section includes tools for indexing and querying Twitter data.

After building the previous jars, we need to populate a SolrCloud index with tweets. It can be processed on these steps:

a) Indexing tweets:

We need to start Solr running in Cloud mode and create a collection named "socialdata" partitioned into two shards with this command line:

```
bin/solr -c && bin/solr create -c socialdata -shards 2
```

Fig. 5 shows the result notes after running this command.

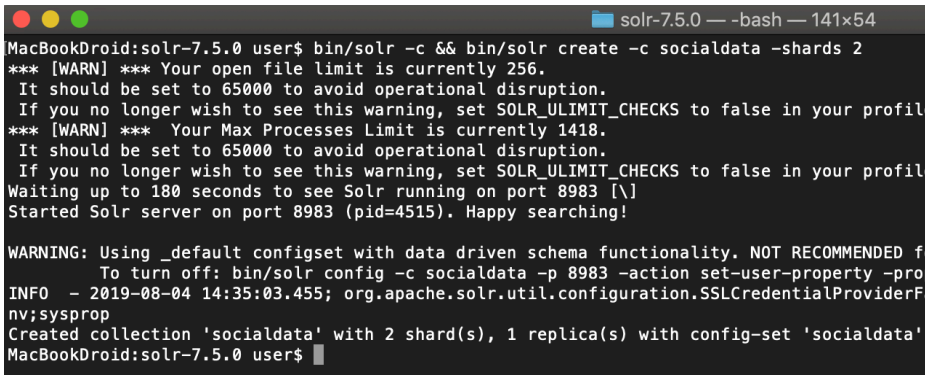


Fig. 5. Creating the "socialdata" collection in Cloud mode.

Now, we need to populate Solr with tweets using Spark streaming. So, we need to run this command line:

```
$SPARK_HOME/bin/spark-submit --master $SPARK_MASTER \
   --conf "spark.executor.extraJavaOptions=-
Dtwitter4j.oauth.consumerKey=? -Dtwitter4j.oauth.consumerSecret=? -
Dtwitter4j.oauth.accessToken=? -Dtwitter4j.oauth.accessTokenSecret=?"
\
   --class com.lucidworks.spark.SparkApp \
   ./target/spark-solr-1.0-SNAPSHOT-shaded.jar \
   twitter-to-solr -zkHost localhost:9983 -collection socialdata
```

We need to replace $SPARK_MASTER with the URL of our Spark master server. If we don't have access to a Spark cluster, we can run the Spark job in local mode by passing:

```
--master local[2]
```

However, when running in local mode, there is no executor, so we need to pass the Twitter credentials in the `spark.driver.extraJavaOptions` parameter instead of `spark.executor.extraJavaOptions`.

b) Querying the tweets:

Now, let's start up the Spark Scala REPL shell to do some interactive data exploration with our indexed tweets:

```
  cd $SPARK_HOME
  bin/spark-shell --jars $PROJECT_HOME/target/spark-solr-${VERSION}-
shaded.jar
```

`$PROJECT_HOME` is the location where you cloned the spark-solr project.

Now, it's time to load the "socialdata" collection into Spark by executing the following Scala code in the shell:

```
val tweets = spark.read.format("solr").options(
 Map("zkHost" -> "localhost:9983", "collection" -> "socialdata")
 ).load
 .filter("provider_s='twitter'")
```

i. We used the sparkSession object loaded into the shell automatically by Spark to load a DataSource named "solr". Behind the scenes, Spark locates the solr.DefaultSource class in the project JAR file we added to the shell using the --jars parameter.
ii. We passed the configuration parameters needed by the Solr DataSource to connect to Solr using a Scala Map. At a minimum, we need to pass the ZooKeeper connection string (zkHost) and collection name (collection). By default, the DataSource matches all documents in the collection, but you can pass a Solr query to the DataSource using an optional query parameter. This allows to you restrict the documents seen by the DataSource using a Solr query.
iii. We're loading the data into DataFrame.
iv. We used a filter function to only select documents that come from Twitter (provider_s='twitter').

At this point, we have a Spark SQL DataFrame object that can read tweets from Solr. In Spark, a DataFrame is a distributed collection of data organized into named columns. Conceptually, DataFrames are similar to tables in a relational database except they are partitioned across multiple nodes in a Spark cluster.

In addition, it should be noted that Spark does not load the collection socialdata in memory at this level. We are only preparing to perform analysis on these data; the actual data is not loaded into Spark until it is needed to perform some calculation later in the job. This allows Spark to perform the necessary column and partition pruning operations to optimize data access into Solr.

5 Conclusion and future work

In this article, we have seen the Big-Learn system in its version 2.5 that is based on Lucidworks, the Apache Spark and Solr project. We have seen also how Indexing data from Spark into Solr using SolrJ and Querying it from the Solr Data source.

As a perspective of this work, we need to implement the whole Big-Learn model of using online search with this new architecture of the Big-Learn 2.5 solution. Then, in a second step, we will study the degree of integration and participation of our solution to improve learning and scientific research for students and which will take as case study students from Abdelmalek Essaadi University.

References

1. Padillo, F., Luna, J. M. and Ventura, S. 2017. Exhaustive search algorithms to mine subgroups on Big Data using Apache Spark. Progress in Artificial Intelligence, 1-14.
2. C. Leeder, C. Shah (2016). Mesuring the Effect of Virtual Librarian on Student Online Search. The Journal of AcademicLibrariabship. [Online]. 42(1), pp. 2-7.
3. Aoulad Abdelouarit, K., Sbihi B. and Aknin N., 2017. Improving Online Search Process in the Big Data Environment using Apache Spark. Proceeding of The Mediterranean Symposium on Smart City Applications (SCAMS 2017), 25-27 October, 2017.
4. Aoulad Abdelouarit, K., Sbihi B. and Aknin N., 2017. Big-Learn 2.0: Towards a new Big Data tool for online search based on open source. Proceeding of The International Conference on Computing and Wireless Communication Systems (ICCWCS 2017), 25-27 October, 2017.
5. Abdelouarit, K. A., Sbihi, B., & Aknin, N. (2017). Towards an Approach Based on Hadoop to Improve and Organize Online Search Results in Big Data Environment. Proceedings of the International Conference on Communication, Management and Information Technology (ICCMIT 2016) (April. 2016).
6. Aoulad Abdelouarit, K., Sbihi B. and Aknin N., 2016. Solr, Lucene and Hadoop: Towards A Complete Solution To Improve Research In Big Data Environment (Case of The UAE) MEDITERRANEAN CONGRESS OF TELECOMMUNICATIONS (CMT'2016), 12-13 May, 2016.

Traffic demand and longer term forecasting from real-time observations

Alexandros Sopasakis

Department of Mathematics,
Solvegatan 18A, 22100 Lund, Sweden

Abstract. We propose an end to end system comprised of real-time image processing in combination with neural networks to describe upcoming traffic demand in order to forecast short as well as longer term traffic evolution and congestion from real-time traffic camera images. The neural networks use both current and historic traffic information collected by analyzing traffic camera images from a number of roads in an actual traffic network.
Specifically we design and train a long short-term memory, a gated recurrent unit and a stacked autoencoder network. We train these networks on data from a single camera location and use each of the three networks to predict traffic density by processing images arriving in real time at all the other camera locations in this traffic network.
The results reveal that such a system could be helpful to provide information about traffic demand and formation of congestion for hours into the future. The traffic data used is collected from the traffic network of Goteborg in Sweden.

Keywords: Forecasting, Time Series, Traffic Demand, Neural Networks, Image Processing, LSTM, GRU, SAEs

1 Introduction

Emerging intelligent transport system technologies could help relieve the ever-increasing congestion with tools like traffic management and forecasting. But these tools require traffic simulations of large-scale urban transportation systems, which is a challenging task. The computational demand of processing massive numbers of events and interactions between commuters requires more computational power than traditional computing solutions can provide. Simple, but accurate, forecasting approaches are needed to provide reliable information about traffic evolution.

Traffic is a chaotic phenomenon and notoriously hard to predict. Traffic scientists agree [3], [13], [14] that there is a gap in theory related to the formation and appearance of traffic congestion and resulting jam.

Traffic behavior and evolution is intrinsically a multi-scale problem. In essence information at very small spatial and temporal scales can profoundly impact intermediate- and large-scale behavior. Fluctuations in the dynamics can play

a dominant role [16] in the system evolution as is evident in long time simulations [8] and asymptotic analysis in a linearized stochastic PDE limit [15]. As a result, resolving the microscopic dynamics is critical. The most widely used methods to produce detailed solutions of traffic models are lattice-based methods involving stochastic Monte Carlo [8] or Cellular Automaton [3], [11] techniques. Errors in such classical modeling methods are known [14], [18] to increase to levels that render predictions useless, when the number of vehicles is 35% (150 veh/lane/km) dense or higher. It is shown [**?**] that the larger the number of vehicles involved, the larger the error will be for currently used state of the art methods. It is computed [**?**] that the state of the art solution is approximately 22% wrong when the traffic is dense; e.g when it matters the most.

Machine learning and neural networks (NN) on the other hand have been successful in uncovering patterns within data. The rise of NN models is attributed to availability of vast data sources and computational resources. It is shown [9] that NN models can easily outperform statistical models such as ARIMA, KARIMA, ARMA and others. Furthermore NN can achieve equivalent and in some cases better results [1], [20] than those produced by advanced mathematical models which have been developed and refined over decades of scientific research. A number of such neural networks have been tried for short-term traffic forecasting [19]. The results indicate that NN can achieve better predictions than classical approaches for short time horizons which can be up to 15 minutes into the future.

For such time-series data recurrent neural networks (RNN) are typically used. LSTM NN, which is a type of RNN have also been used to predict traffic flow and showed performance which is better than advanced non-parameter models [19]. It has been observed however that, for some data, gated recurrent units (GRU) which is another type of RNN could converge faster and perform slightly better than LSTMs [4]. At the same time stacked autoencoders (SAEs) NN have also been tried in describing traffic. SAEs networks make the model deeper and can be considered as a deep-learning method.

In this work we process real-time images and produce values of vehicle densities in order to train our NN models. The data used for this work consists of traffic densities recorded at different locations and during different times. We provide a description of the image processing which we carry out at each of the cameras in order to estimate vehicle densities in Section 2. We then discuss the structure and design of each of the three neural networks (NN) which we use in this work. We provide information about how training on our collected historical data is curried out for each of these network in order for them to learn traffic density patterns for forecasting purposes in Section 3. Then in Section 4 we present a number of resulting forecasts. We end with a discussion of the results in Section 5.

2 Traffic networks and neural networks

We use images from traffic cameras in combination with image processing methods, which we present below, in order to make it possible to automatically produce traffic density at the specific camera locations for the traffic network. We record these densities for a period of time in order to accumulate sufficient historical data from the traffic network. We then use the data to allow each of the three neural networks we designed to learn traffic patterns in a supervised way. Subsequently we use the network to produce traffic predictions from real-time camera images.

2.1 Processing of the traffic camera images

We download images directly from each camera in the Goteborg traffic network a part of whihc is shown in Figure 1. The images are available at a resolution of 960x720 pixels each. The images are only available in real time and are lost immediately after that, unless saved. These images are available at intervals which are one minute apart and provide a snap-shot of the state of traffic at the camera locations 24 hours a day over the whole city throughout the year.

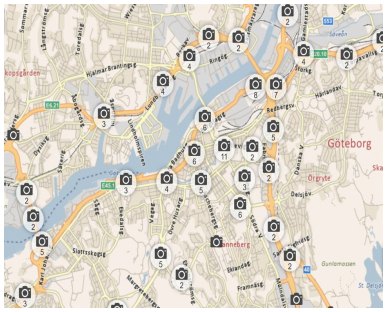


Fig. 1. A section of the Goteborg traffic network with multiple cameras indicated along each road each collecting images every minute. Data provided by the Swedish traffic authoridy: Trafikverket.

Images are not saved on the cameras nor on our server. Instead we analyze each image in real time in advance of the next recording. During that one minute therefore we download the images from all cameras in the city, process them and record their estimated traffic densities for each location at that time instance.

We use a mask, which can be seen in Figure 2 to restrict processing to only those parts of the image for which the vehicle densities are to be counted. Each camera must have its own mask according to the location of the road of interest on that image. Note that the original image is purposely blurred at the source (the traffic camera) in order to adhere with local and European privacy laws so that no individuals or other sensitive and identifying information is possible to be recorded by any of the cameras in the city.

We apply a series of quick low pass and high pass filters which assist with detecting edges while at the same time segmenting the image, as needed, in order to compute vehicle densities within each frame.
Specifically we impose the following series of filtering methods in that order: denoising, blurring, Canny edges, bilateral filter and finally a threshold for counting

in order to more accurately produce current road density (normalized capacity). Some of these tests can be seen on Figures 2-4.

We provide more details for each step below although the details of this classic approach by John Canny can be found in [2]:

Fig. 2. Original image with superimposed mask on the region of interest.

Low pass filter - Denoising. Denoising is performed immediately on the image since our next step, edge detection, is sensitive to noise. Since the cameras used in traffic are perhaps not of the clearest resolution possible such a step is needed to ensure a minimum standard providing guarantees for other image processing procedures which will follow. To remove the noise we process the whole image with a moving 5x5 pixel box using a Gaussian filter. Other filters could have also been used to do this (averaging, median etc).

High pass filter - Canny edge detection. We perform edge detection by computing the intensity gradients of the image. Assuming an image G these are simply the partial derivatives, G_x and G_y, in the x and y directions between neighboring pixels. Edge detection and the corresponding direction are then computed from $E_G = \sqrt{G_x^2 + G_y^2}$ and $\tan(\theta) = G_y/G_x$ respectively. We eliminate edges which are week by imposing a double threshold on the above gradients together with blob detection.

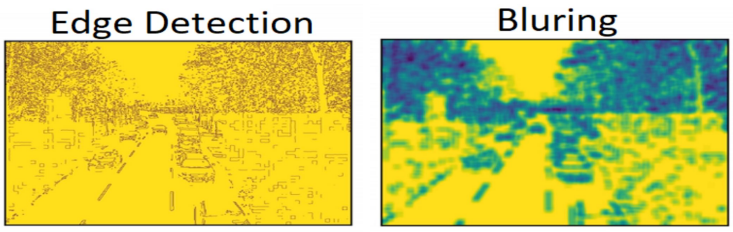


Fig. 3. Left: edge detection finds the clear structures within the mask imposed earlier - see Fig. 2. Right: blur fills pixels in between edges and prepares image for more accurate counting later.

Blur. We apply a bilateral, nearest neighbor, Gaussian filter which is another low pass filter in order to further remove noise from color as well as to produce better segmentation and allow us to more accurately localize vehicles against the background. Bilateral filtering is effective here since it does not destroy our edge detection from the previous step. We present such a result in the left side of Figure 3.

Threshold. We now apply a final filter which changes all pixels above a specified threshold value to white while the remaining pixels will be changed to black (see Figure 4). We use an automatic threshold value which is computed as a local average of a rolling 5x5 pixel window.

Counting. Vehicle densities are now simply computed as the number of white pixels over the total pixels in a given area of the image which was marked in advance as a polygon. The image above was estimated to contain a vehicle density of 30.7% within the green mask (see Fig. 2. We mark a polygon shape mask of appropriate size for each camera, manually. This polygon mask is an input required by our image detection algorithm in order to focus its analysis on that specific region instead of the whole image. Since the traffic cameras are not moving this type of work need to only be performed once for each camera. Note also that due to many local and European laws camera images automatically blurr, at the source, surrounding information which is not related to the traffic on the road.

Fig. 4. Threshold example just before counting density. This is a highway image where we have both oncoming (on the left side) and outgoing (on the right side) traffic. In this instance two different masks, one on each side of the highway, are used.

3 Neural networks and training on time series data

In this section we present the network design used as well as training features for three different types of neural networks. All three networks will be trained to learn traffic patterns from the same available historic data and subsequently will be asked to reproduce vehicle densities from real-time traffic data. In that respect we explore which of the three networks would perform best to predict traffic densities. Recurrent type networks are implemented since they allow events from past time points to be held in memory if they are deemed as important enough based on the historic data provided. As with all networks it is up to the network to learn which patterns or events within the data will be deemed as important or not. This is typically the work of back propagation which is responsible for finding and re-enforcing the importance of each connection between neurons in the layers of the network.

Recurrent networks allow connections to be maintained not only between network layers but also between different points in time as can be seen in Figure 5. There is a variety of such Recurrent Neural Networks (RNNs) networks each with different properties of interest to be exploited according to modeling targets. Examples of RNNs are the Long Short Term Memory networks (LSTMs) and the Gated Recurrent Units (GRUs). We choose LSTMs and GRUs since such networks have shown good recall capabilities and are not suffering from the

exploding or vanishing gradient problems [6] that classic RNNs have. Furthermore LSTMs and GRUs are typically stacked on top of each other, as we explain in Section 3.1, thus emulating a deep network type structure. For our third NN we use a Stacked Autoencoder (SAEs) network since their behavior also resembles deep-learning networks. Such SAEs networks and have been successful in producing promising short-term forecasts in traffic modeling [19].

Since these networks create connections which are based on time then the way by which we feed data into the network is very important. In order to assist the networks in their learning and in particular to built meaningful connections between historical time events we must make sure to provide data which, in contrast with typical neural network training, are not going to be randomly chosen. We therefore make sure to maintain the time ordering of the data in order to allow the network to learn from temporal events.

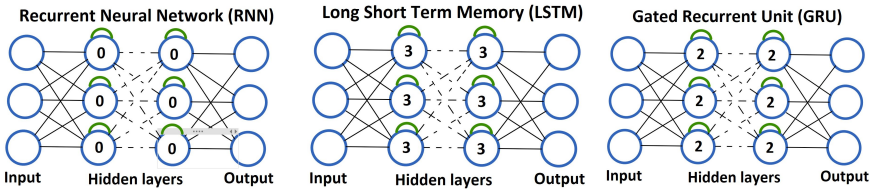


Fig. 5. Diagrams depicting three different types of neural networks typically applied on time series data. The numbers within each cell indicate the number of decision gates. Left: Recurrent NN (RNN) incorporate hidden layers which process both new and past information by allowing cells to connect to themselves. No gates, to control information, are present. Center: Long Short Term Memory (LSTM) network. In total 3 gates control information: input, output and forget gate. Right: Gated Recurrent Unit (GRU). They look a lot like LSTMs but instead of an input, output and a forget gate they only have 2 gates: a reset and an update gate.

3.1 LSTMs, GRUs and SAEs networks

As NN are trained on data the weight values between their vast network of connections are slowly improving. There are however some values which cannot be determined by the provided data. These are called hyperparameters and must instead be optimized by brute force. In all such modeling approaches therefore serious effort is spend on finding which set of hyper-parameters will perform best for the given data.

Weights are responsible for maintaining connections between neurons in networks. At each time step information within the data is fed to the network in order to strengthen or weaken these connections. RNNs are slightly different since they allow connections not only between their neurons but also with past information as can be seen in Figure 5. The length of time by which neurons in

the network remember will depend on the weights between these time connections not becoming 0. To updated these weight we must compute the gradient of the loss function with respect to each weight. While computing the weights for RNNs their gradients tend to either explode or become 0 over longer times. Due to this problem RNNs cannot maintain memory information for long time periods and become ineffective for our purposes at least.

In contrast each neuron in an LSTM network has an input gate, an output gate and most importantly a forget gate. In total therefore 3 gates are used to control information. The forget gate in particular guards the information flow between time steps and either allow it to be remembered or to be forgotten. These gates allow control of the respective gradients by not allowing them to become 0 or extremely large and as a result such networks can remember events within the data for longer times. LSTMs therefore allow us a lot more control in terms of their memory capabilities [12]. Inherently however it is the data which indirectly dictates whether information is sufficiently important to be remembered or not. A typical LSTM network can be seen in Figure 5.

GRUs are very similar to LSTMs but have a slightly different internal decision system. GRUs do not have an output gate. They only have 2 gates in total: a reset and an update gate. The main difference is that GRUs are able to access the full information available. In some cases the GRUs are less expensive to train than LSTMs [7]. However which of the two NN is better for a specific application is dependent on the data in a way that is not clear. As a result the best way to decide is to try both networks on the specific problem to be solved and decide based on the outcome.

We also use a stacked autoencoder (SAEs) network to understand patterns within the traffic data. Autoencoders are typically associated with applications in compression. In an autoencoder network we represent features in the data with a lower amount of cells than those in the input. As a result we create a bottleneck inside the hidden layer of the network as can be seen in Figure 6. That bottleneck in return is fully connected with the output layer which is at the same dimensionality as the input layer. In general autoencoders are used to find the features which are most important in the data. In a SAEs network we simply stack the autoencoders to a successive bottleneck within multiples of hidden layers. Thus the network successively shrinks and learns which are the important features in the data.

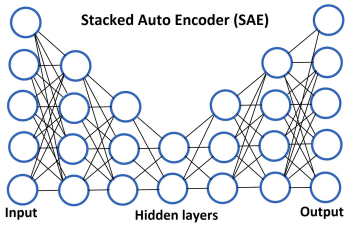


Fig. 6. Schematic of a Stacked Auto-Encoder (SAEs). In this example an encoder/decoder each with two hidden layers.

We present in Tables 1 - 3 the specific architecture as well as respective sizes of the hidden layers for each of the three networks tried. These designs were chosen

Table 1. Structure of LSTM network designed to process each of the camera images. One dropout layer is used between the 2nd hidden layer and the dense layer. We use a Sigmoid activation function in the dense layer. Note that in total the LSTM network has a total of 49985 parameters.

LSTM Network	1st Hidden Layer	2nd Hidden Layer	Dense Layer
number of cells	64	64	1
input/output	288/(64,288)	(64,288)/64	64/1
Parameters	16896	33024	65

after extensive testing where we changed not only the network hyper-parameters but also the number of hidden layers in order to make sure we reached a good network configuration for the problem data given.

Table 2. Structure of GRU network designed to process each of the camera images. One dropout layer is used between the 2nd hidden layer and the dense layer. We use a Sigmoid activation function in the dense layer. Total number of parameters here is 37505.

GRU Network	1st Hidden Layer	2nd Hidden Layer	Dense Layer
number of cells	64	64	1
input/output	288/(64,288)	(64,288)/64	64/1
Parameters	12672	24768	65

Table 3. Structure of SAEs network designed to process each of the camera images. In total we have 2×438603 parameters.

SAEs Network	Dense 1	Dense 2	Dense 3	Dense 4	Dense 5	Dense 6
Size of layer	400	400	400	400	400	400
Parameters	116001	160801	160801	160801	160801	116001

The network learns and self-adjusts its weights in order to improve the next time around based on error estimates produced by the loss function chosen.

We use the Mean Square Error (MSE) as a loss function which is typical of many neural network implementations. Even thought we received images and recorded the traffic density every minute we aggregated these recording to 5 minute intervals. The data was then formatted to have a lag of 288. This allowed for a time-series input structure representing a full day comprised of our 5-minute aggregated measurements. The batch size or number of samples used is 256 with a lag of 288 for all three networks. We apply an Adam optimizer during learning on each batch.

4 Numerical results

In this section we begin by first presenting the image processing carried out in order to obtain quick estimates of the traffic density within each image. We then provide the resulting predictions from real-time data based on the training carried out.

4.1 Training procedure and results

We train each of our three networks for 600 epochs with all our data spanning approximately 2 months. The data is first averaged out to 5 minutes and then batch fed into the networks. We use a batch size of 256. We use a time series of 288 five minute data to train our networks on recognizing daily patterns. The batch size as well as a number of other hyper-parameters is varied in order to find an optimal implementation. The results of loss as well as validation loss is provided in Figure 7.

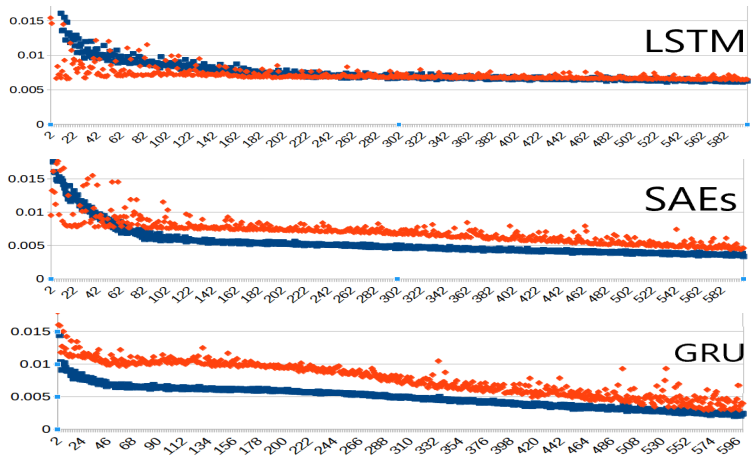


Fig. 7. Training and validation loss results for LSTM, SAEs and GRU networks over 600 epochs. All networks display great learning potential although the GRU is achieving lower loss values for equivalent epochs. Blue is training loss. Red is validation loss.

It is already clear from Figure 7 that the best result is achieved by the GRU network. This however becomes a lot clearer later when forecasts are produced by each of those three networks and compared against real traffic densities. We present these results in the next section.

4.2 Predictions

We present results from both short and longer time predictions and compare them against reality for the respective times. As a result for each of these times

a specific NN must be specifically trained. We present some of these results below while comparing all three NN for predictions spanning from 30 minutes and up to 2 hours.

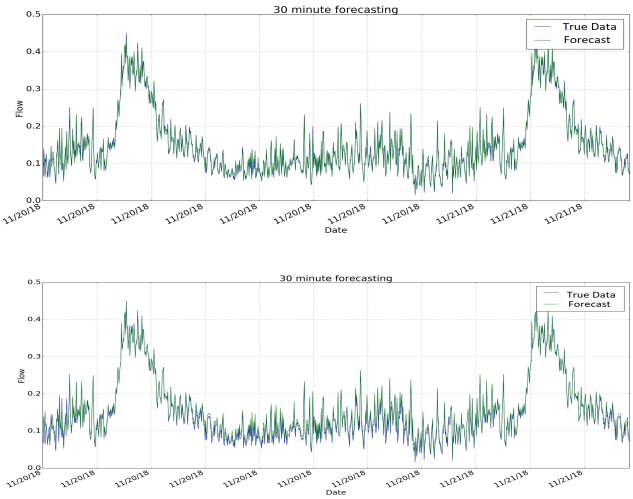


Fig. 8. A day and a half of 30 minute predictions presented. After adjusting the hyperparameters all three networks produced much better results. Top: GRU network. Bottom: SAES network. The differences between the network predictions are now minor however the GRU seems to have a minor advantage over the SAEs network.

We present a forecast produced by a 30 minute GRU NN versus the 30 minute SAEs in Figure 8. After optimizing the hyperparameters in all three networks their predictions are much closer to actual traffic densities observed. However the GRU network seems to still have a slight advantage in accuracy of predictions over both the SAEs and the LSTM network. In Figure 9 we present longer term forecasts between the GRU and the LSTM networks. These 60 minute predictions although not as accurate as the shorter term predictions presented in Figure 8 they still show that major trends in vehicle densities are possible to be captured. Minor fluctuations although important are not so accurately predicted however. This result and many other like it shows that this methodology could perhaps be used for forecasting major upcoming trends in the data instead of localized fluctuations. More discussion about these and similar findings are given below in Section 5.

5 Discussion

In this work we presented a system which uses minute to minute traffic camera images in order to produce short and longer time predictions of traffic densities

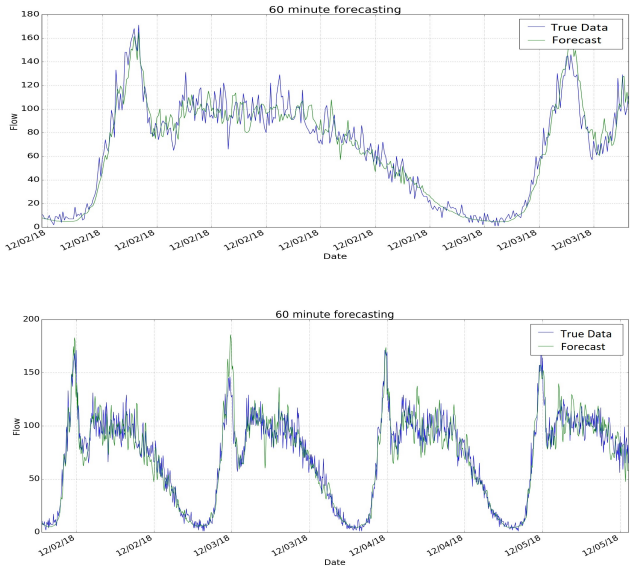


Fig. 9. 60 min predictions presented. Top: predictions from the GRU network for a day and a half. Bottom: LSTM network predictions for 4 days.

at each of the camera locations. To accomplish this we begin by uploading each of those images from every camera in Goteborg. Once each image is collected we apply a sequence of filtering algorithms in order to remove noise and clarify the vehicles against the background. This provides an estimate of the traffic density for that image. Our algorithms only estimate the vehicle densities on previously defined polygonal areas in the image since the traffic cameras are not moving. Note also that once these polygonal areas are established in each image need no further adjustment. We collected such traffic density data for approximately 2 months from large number of cameras in the city center. The collected data was subsequently used to train an LSTM a GRU and a SAE neural network to produce predictions at a number of future time instances. Subsequently the networks were also used on never seen before real-time data in order to establish their capabilities for short and longer time predictions of traffic densities at each of the given camera locations.

A number of interesting results were found from analyzing predictions produced after initial training of each of the NN designed in this study. Specifically,

- It is possible to train a NN on a single camera data and using this produce realistic forecasts on all the other camera stations in the network.
- Small fluctuations could be captured for shorter predictions under 15 minutes but only bigger trends were possible to be predicted for forecasts ranging from 30 minutes to 2 hours (Figure 10).

– In the case of capturing bigger trends it is perhaps possible to capture rare events not only in the short term but also for longer term time horizons. We present such a rare event below although further study is needed to understand which factors influence such longer term predictions.

A particularly surprising result was found for data collected during November 20th and until November 23, 2018 in Figure 10. In that figure we present a 2 hour GRU forecast although the results are equivalent for predictions ranging from 10 minutes to 2 hours. Note that the usual daily traffic peak is missing. The reason that the real traffic density never reached its usual peak is that this was a black Friday and apparently a number of drivers have not followed their daily trip to work in the city center. In Figure 10 we see that this deviation from a usual rush-hour commute can be predicted by such a network.

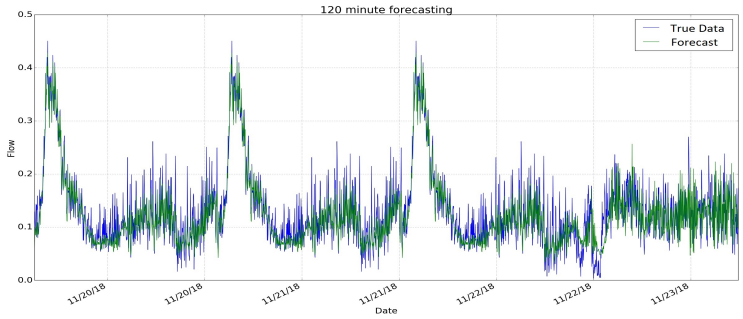


Fig. 10. An unusual result. Predictions using a GRU trained to forecast for 2 hours. The network is predicting an anomaly. Predictions from data during 4 full days including the black Friday on the last day which is missing the daily traffic peak.

Acknowledgment. The author would like to thank Lunarc, the supercomputing facility at Lund university, for dedicated parallel computer time. We also would like to thank Trafikverket (the transportation authority of Sweden) and the city of Gothenburg for access to their traffic camera data.

References

1. Chen C., Wang Y., Li L., Hu J. and Zhang Z.: The retrieval of intra-day trend and its influence on traffic prediction Transportation Research Part C: Emerging Technologies, Vol 22, 103–118 (2012)
2. Canny F. J.: A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6):679–698 (1986)
3. Helbing D., Hennecke A., Shvetsov V., and Treiber M.: Micro and macro simulation of freeway traffic, *Math. Comp. Modelling*, 35:517 (2002)

4. Fu R., Zhang Z. and Li L.: Using LSTM and GRU neural network methods for traffic flow prediction, Chinese Association of Automation, 2017:324-328 (2017)
5. Lv Y., Duan Y., Kang W., Li Z. and Wang F. Y.: Traffic Flow Prediction With Big Data: A Deep Learning Approach, IEEE Transactions on Intelligent Transportation Systems, 16(2):865-873 (2015)
6. Elman J. L.: Finding structure in time, Cognitive science, 14.2, 179-211 (1990)
7. Chung J., Caglar G., Kyung H. C. and Yoshua B.: Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv, 1412.3555 (2014)
8. Katsoulakis M. A., Majda A. J., and Vlachos D. G.: Coarse-grained stochastic processes for microscopic lattice systems In: *Proc. Natl. Acad. Sci. USA*, **100** (3), pp.782–787 (2003)
9. Lingras P. J., Sharma S. C., Osborne P. and I. Kalyar I.: Traffic volume time-series analysis according to the type of raod use, Computer Aided Civil and Infrastructure Engineering, 15, 365–373 (2000)
10. Morimura T., Osogami T. and Ide T.: Solving inverse problem of Markov chain with partial observations, Technical Report RT0952: IBM-Research, Tokio (2013)
11. Nagel K. and Schreckenberg M.: A cellular automaton model for freeway traffic, *J. Phys. I*, 2:2221 (1992)
12. Sepp H. and Schmidhuber J.: Long short-term memory, Neural computation, 9.8, 1735-1780 (1997)
13. Schadschneider A.: Traffic flow: a statistical physics point of view, *Physica A*, 312:153 (2002)
14. Schreckenberg M. and Wolf D. E.: *Traffic and Granular Flow*: Springer, Singapore (1998)
15. Sopasakis A. and Katsoulakis M.: Improving traffic model fidelity and sensitivity through information theory, *Trasnp. Res. Part B*, 86, p.1-18 (2016)
16. Sopasakis A.: Lattice free stochastic dynamics, *Comm. Comput. Phys.*, 12(3), pp.691–702 (2012)
17. Tympakianaki A., Koutsopoulos H. and Jenelius E.: c-SPSA: Cluster-wise simultaneous perturbation stochastic approximation algorithm and its application to dynamic origin-destination matrix estimation, Transportation Research Part C, **55**, pp.231-245 (2015)
18. Tossavainen O. and Work D.: Markov chain Monte Carlo based inverse modeling of traffic flows using GPS data. In: Networks and Heterogeneous Media, **8** (3), pp.803–824 (2013)
19. Vlahogianni E. I., Karlaftis M. G. and Golias J. C.: Short-term traffic forecasting: Where we are and where we're going. Transportation Research Part C: Emerging Technologies 43, 3-19 (2014)
20. Zhong M., Sharma S. and Lingras P.: Genetically Designed Models for Accurate Imputation of Missing Traffic Counts, Transportation Research Record, 1879, 71-79 (2004)

The Impact of Signed Jump Variation on Forecasting Realized Variance

Ioannis Papantonis* Leonidas Rompolis Elias Tzavalis

papantonis@aueb.gr rompolis@aueb.gr e.tzavalis@aueb.gr

Athens University of Economics & Business

May 29, 2019

Abstract

It is a well-known stylized property that statistical distributions of financial time-series exhibit significant variation over time, as well as prevalent deviations from normality. The vast majority of earlier financial econometric studies has employed GARCH and SV model-classes in order to capture the empirically-observed time-variation (clustering) in volatility. This has also offered a flexible parametric framework to address other empirically-observed properties of the data, such as leverage and/or feedback effects -among others- which are partly responsible for inducing asymmetries in the distributions of returns. Another major source of asymmetry and tail-heaviness has been attributed to the existence of latent jump dynamics in both returns and variance. Recent advances to the econometrics literature of high-frequency data have popularized approaches that permit identification of the underlying jump dynamics. These realized jump-driven estimators carry significant information not only for the returns, but also for volatility.

This paper extends Hansen's Realized-GARCH framework to allow for the impact of jump variation in the conditional variance dynamics of returns. First, we build intraday evenly-spaced observations from tick-by-tick data that we use as building blocks for our realized estimators. We decompose realized variation into its continuous (quadratic) and discontinuous (jump) components and we utilize the realized semi-variances to construct realized signed jump variation measures. Next, we employ a collection of asymmetric GARCH-type processes that we augment to incorporate our realized estimators. At the same time, we explore the parametric specification of the model that can better explain the joint dynamics of returns and realized variance, by performing non-nested model-comparisons and appropriate evaluation of volatility forecasts. This methodology allows us to test several interesting hypotheses; among these we examine if conditional variance responds differently to the upside and downside semi-variances, if there's a material jump contribution to variance, and also if there is an asymmetric behavior behind the impact of signed jump variation. Last, our main objective is to assess the performance

*Corresponding Author. The author gratefully acknowledges financial support from the Onassis Foundation.

of this joint approach in forecasting realized variances at multiple horizons/frequencies. Our paper also implicitly tries to bridge the gap between the GARCH models of conditional variance and the more-recent HAR/HEAVY models of realized variance, and it has really important implications for risk-management, option-pricing and macro-economic uncertainty forecasting.

Keywords: Realized Variance; Semi-Variance; Quadratic Variation; Jumps; Signed Jump Variation; Multi-horizon forecasts.

Copper Price Variation Forecasts using Genetic Algorithms

Raúl Carrasco[1,2][0000−0002−5023−9349], Ismael Soto[1][0000−0002−5501−5651], Christian Fernández-Campusano[3][0000−0001−7033−4178] Carolina Lagos[4][0000−0003−4061−8035], Nicolas Krommenacker[5][0000−0003−0599−3614], and Claudia Durán[6][0000−0002−0903−4333]

[1] Departamento de Ingeniería Eléctrica, Universidad de Santiago de Chile, Santiago 9170124, Chile
[2] Facultad de Ingeniería, Ciencia y Tecnología, Universidad Bernardo O'Higgins, Santiago 8370993, Chile
[3] University of the Basque Country UPV/EHU, Faculty of Informatics, Department of Architecture and Computer Technology, Donostia-San Sebastián, Spain
[4] Facultad de Ingeniería, Pontificia Universidad Católica de Valparaíso, Chile
[5] Université de Lorraine, CRAN CNRS UMR 7039, F-54506 Vandoeuvre Les Nancy, France
[6] Faculty of Engineering, Department of Industry, Universidad Tecnológica Metropolitana, Santiago 7800002, Chile
raul.carrasco.a@usach.cl

Abstract. Today, the use of Genetic Algorithms and the Big Data improves decision-making effective, concerning variation in copper prices. This work analysis volatility forecasting for the copper market in a time period, which is of interest in different participants such as producers, consumers, governments, and investors. For this, we propose to apply genetic algorithms to predict the variation in copper prices, in order to improve the degree of certainty by incorporating of the inverse of the percentage of sign prediction PSP.

Keywords: Genetic Algorithms · Forecasting · Directional Accuracy · Copper.

1 Introduction

The price of copper and its variations is an essential financial problem for mining companies. In the case of Chile, this problem affects the Chilean government, due to the strong impact on the results in the country's economy. The series of prices in the general markets and of the commodities, as is the case of the copper, present the high volatility, dynamics and turbulence, due to this it is imperative to estimate its price.

Mathematical modelling and prediction of commodity values is the subject of constant research by private agents, insurers and government institutions to

ensure free competition in the securities market [1,2,3,4]. In this case, to apply Genetic Algorithms to predict the variation in copper prices.

Today, capital markets require real-time information (data) to support and enable short-term and long-term decision-making, allowing them to manoeuvre effectively in the face of the turbulent global economy (which we denoted as *latency reduction*).

Regarding the latency and the value of the data. The value of the data decreases rapidly, that is, the low latency data has more value than the high latency data. Reducing data and analysis latency depends essentially on technical solutions in Big Data Analytic. However, reducing decision latency demands changes in business processes. Thus, providing fresher data does not create business value unless it is used in a timely manner.

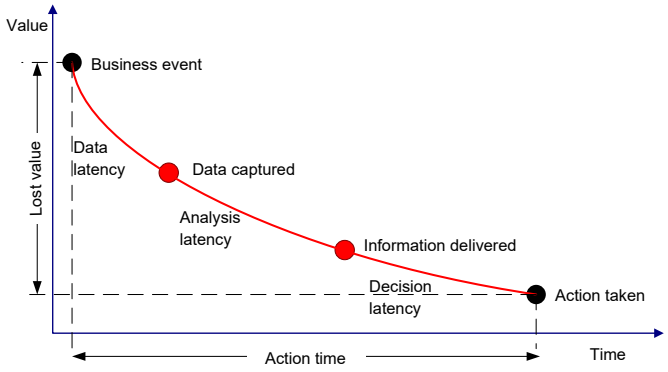


Fig. 1: Latency in Decision-Making.

In Figure 1 proposed in [5], we can see the latency reduction curve. The longer the delay or latency of the response, the lower the value. Figure 1 shows the time of action (or distance of action), that is, the duration between the event and the action, and the net benefit is the value of the decisions (lost or gained) over some time.

2 Genetic Algorithm

The Genetic Algorithms (GA) refers to the evolutionary algorithms class and its development was inspired through the process of natural genetic evolution. Initial work on GA was conducted by Holland [6] in 1975, which explores their use in the study of a wide range of complex, naturally occurring processes, concentrating on systems having multiple factors that interact in non-linear ways. Works that

allowed investigations such as the related to modelling dynamic multivariate forecasting systems [7].

For the model, a partial solution is represented by binary chains of constant length. This local solution is improved using multi-point search methods based on evolutionary theories, achieving better quality and speed solutions in relation to the search algorithms previously investigated.

The algorithm is implemented in an object-oriented platform to obtain forecasts on copper prices. On the other hand, the data analysis is carried out with the statistical program R [8].

2.1 Phases of the Genetic Algorithm

Figure 2 shows four phases of the genetic algorithm, which are generation, population, actions and results. Which are detailed below:

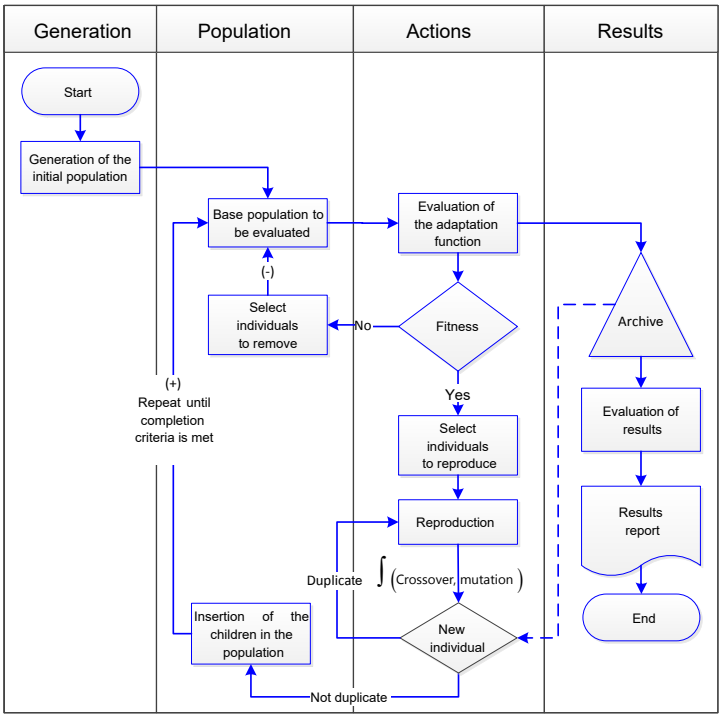


Fig. 2: Four Phases of the Genetic Algorithm.

P1. Generation: It generates the initial population in a stochastic process of predefined size.

P2. Population: Determine a population of thirty individuals, which are renewed through the departure of unfit individuals with the incorporation of the new evolved genetic material. The steps in this stage are:

- To prepare the base population for evaluation.
- To eliminate individuals who do not qualify according to the aptitude function defined in the problem.
- To incorporate into the base population, the children reproduced by the best individuals in the population, adding, in this way, new genetic material.

P3. Actions: Evaluate and select individuals according to aptitude function, then reproduce and mutate. The steps of this stage are:

- To evaluate individuals according to the function of aptitude.
- To select the individuals:
  - Capable, those will remain in the base population and transfer their genetic material to the next generation;
  - Incapable, those will be eliminated from the base population.
- To reproduce capable individuals according to crossing operators.
- To mutate the reproduced individuals according to the mutation operators and the restrictions of not duplicating the individuals, which must pass to the base population.

P4 Results: To archive the results of each generation and, once the algorithm is finished, evaluation and analysis of the results are made to produce a report of these.

3 Application: A Copper Price Forecast

The applicability of the genetic algorithm is to obtain a dynamic multivariate forecast model, which maximizes the predictive percentage of sign related to the daily variations of copper prices (Cu) presented by the London Metal Exchange discovering a mathematical formula that generates approximately the historical patterns of copper time series.

Our time series corresponds to a sequence of values that measure the price of markets at equal intervals of time, maintaining the consistency in the activity and the method of measurement. In this case, the extra sample data correspond to daily closing values for the period from February 24 to August 29, 2016. For the intro-sample variability period, the days of lags corresponding to *Rolling* of 60 days was considered.

The dynamic multivariate forecast models used help to project the future prices of the time series [2,9], understanding the behaviour and what is happening with the data of the Cu and Dow variables as a stock market, according to their values *Stragglers*.

3.1 Metrics and Data

Determination of Variations. To determine the price variations of the Dow and Cu, in general terms it is expressed with the difference operator ∇. This operator is used to express relations of type $\nabla Y_t = X_t - X_{t-1}$, where X_t is a balance variable and ∇Y_t will be the corresponding flow variable [10].

∇ is defined as:

$$\nabla Y_t = X_t - X_{t-1} \ \ \forall t, \ \ t \in Z \tag{1}$$

Where: ∇Y_t is the price variation, X_t is the price of the period, X_{t-1} is the price of the previous period.

We can represent the variations of Dow ***and*** Cu***, according to equation (1).***

$$\nabla Dow_t = Dow_t - Dow_{t-1} \ \ \forall t, \ \ t \in Z \tag{2}$$

Where: ∇Dow_t is the price variation of the Dow, Dow_t is the price of the Dow period, Dow_{t-1} is the price of the previous Dow period.

$$\nabla Cu_t = Cu_t - Cu_{t-1} \ \ \forall t, \ \ t \in Z \tag{3}$$

Where: ∇Cu_t is the price variation of Cu, Cu_t is the price of the Cu period, Cu_{t-1} is the price of the previous Cu period.

Determination of Projections. The projection of the variation of Cu will be denoted as a function of the lags variations of the Dow and Cu of equations 2 and 3 and of forecast errors.

It is defined in this study as:

$$\nabla \overline{Cu}_t = \sum_{i=1}^{4} \theta(\nabla Dow_{t-i} \bullet \beta_{di} + \nabla Cu_{t-i} \bullet \beta_{ci} + \epsilon_{t-i} \bullet \beta_{ei}) \tag{4}$$

Where:

$\nabla \overline{Cu}_t$, is the projected variation,
$\theta()$ is the Heaviside function, which multiplies the betas calculated with the input of the variable,
∇Dow_{t-i} are the lags of the Dow variations,
∇Cu_{t-i} are the lags of the variations of Cu,
ϵ_{t-i} is the prediction error.

The projection is done by minimizing the squared error of the 60-day *Rolling* estimate with Newton Raphson, defined by:

$$\min\left(e^2(n)\right) = \min\left(\sum_{i=1}^{n}\left(\nabla Cu_t - \nabla\overline{Cu_t}\right)^2\right) \tag{5}$$

Where:

$n = 60$,
$e^2(n)$ is the squared error of the estimate in the n periods,
∇Cu_t is the value of the variation of the period,
$\nabla\overline{Cu_t}$ is the value of the projected variation.

Note 1: The objective function is not subject to a series of constraints.

Note 2: The vector of decision variables corresponds to the calculated betas, which minimize the sum of the errors to the prediction table for *Rolling* of 60 days.

Determination of the Percentage of Sign Prediction (PSP). Moreover, we must determine the PSP [11]. To calculate the PSP, the sign of the projected variation is compared with the sign of the observed variation, in each period of $t+n$, where $t = 1, 2 \ldots, n$ starting from $t+1$. If the signs of the projected variation and observed variation coincide, the value "1", is obtained, which represents a success. In the opposite case, "0" indicates a model prediction error [12]. If the signs coincide, the effectiveness of the prediction increases, and in case of no coincidence, the prediction error of the model increases. The PSP of the model is defined by:

$$PSP = \frac{\sum_{j=1}^{n}\theta\left(\nabla Cu_{j,t+1} \bullet \nabla\overline{Cu}_{j,t+1}\right)}{n} \qquad \forall, 1 \leq j \leq n \tag{6}$$

Where:
PSP is the percentage of sign prediction (PSP) presented in the equation,
∇Cu_t is the value of the period variation,
$\nabla\overline{Cu_t}$ is the value of the projected variation,
$\theta()$ is the dichotomous function of Heaviside; $\theta() = 1$ iff $\nabla Cu_{j,t+1} \bullet \nabla\overline{Cu}_{j,t+1} > 0$, or $\theta() = 0$ iff $\nabla Cu_{j,t+1} \bullet \nabla\overline{Cu}_{j,t+1} \leq 0$,
n is the total number of predictions performed.

The $PSP_{\max}$ variable used will be the maximum between PSP and $(1 - PSP)$, as shown in equation 7.

$$PSP_{\max} = \max(PSP, (1 - PSP)) \tag{7}$$

Test of Pesaran and Timmermann. The directional accuracy Test of Pesaran and Timmermann [13] was applied in order to measure the statistical significance of the predictive capacity of the dynamic multivariate prognostic model with genetic algorithms [14,15].

The directional correctness test is used to measure the statistical significance of the predictive ability of the models analyzed [11,16]. The directional correctness test tests the null hypothesis that the observed variations are distributed independently of the projected variations. Therefore, if the null hypothesis is rejected, it is said that there is statistical evidence that the model has the ability to predict the future evolution of the observed variable.

This test compares the sign of the projection with that of the observed value for each *j-esime* observation of the sample set ($j = 1, 2, ..., n$). Where the sign indicates the direction in which the stock market will move: up if it is positive, or down if it is negative. If the signs coincide, the prediction effectiveness increases, and in case of no coincidence, the prediction error of the model increases (same as the methodology used to calculate PSP).

In order to obtain the percentage of real positive changes observed, the following equation is represented by:

$$P = \frac{\sum_{j=1}^{n} \theta\left(\nabla P_{j,t+1}\right)}{n} \tag{8}$$

Where:

P is the percentage the real positive changes observed, and

$\theta()$ is the dichotomous function Heaviside; $\theta() = 1$ iff $\nabla P_{j,t+1} > 0$ or $\theta() = 0$ iff $\nabla P_{j,t+1} \leq 0$.

The percentage of positive projection variations is represented in the following equation:

$$\overline{P} = \frac{\sum_{j=1}^{n} \theta\left(\nabla \overline{P}_{j,t+1}\right)}{n} \tag{9}$$

Where:

$\overline{P}$ is the percentage of projected real positive changes, and

$\theta()$ is the dichotomous function Heaviside; $\theta() = 1$ iff $\nabla \overline{P}_{j,t+1} > 0$, or $\theta() = 0$ iff $\nabla \overline{P}_{j,t+1} \leq 0$.

In addition, the success ratio when the actual variations and projected variations are independently distributed for $\nabla P_{j,t+1}$ and $\nabla \overline{P}_{j,t+1}$, SRI, is given by:

$$SRI = P\ \overline{P} + (1 - P)(1 - \overline{P}) \tag{10}$$

To determine the variance of the SRI ratio, it is defined as:

$$\operatorname{var}(SRI) = \frac{\begin{pmatrix} n\left(2\overline{P}-1\right)^2 P\left(1-P\right)+ \\ n\left(2P-1\right)^2 \overline{P}\left(1-\overline{P}\right)+ \\ 4P\overline{P}\left(1-P\right)\left(1-\overline{P}\right) \end{pmatrix}}{n^2} \tag{11}$$

On the other hand, the variance of the SR ratio, it is defined as:

$$\operatorname{var}(SR) = \frac{SRI\left(1-SRI\right)}{n^2} \tag{12}$$

Finally, the Directional Accuracy (DA) by [15] is given by:

$$DA = \frac{\left(SR-SRI\right)}{\sqrt{\operatorname{var}\left(SR\right)-\operatorname{var}\left(SRI\right)}} \quad \xrightarrow{d} \mathrm{N}\left(0,1\right) \tag{13}$$

This test follows a standard normal distribution. The result of this equation is compared to a critical one, which will depend on the level of trust required to be tested. That is to say; If the DA value is between the rejection values, we do not reject the null hypothesis that the observed variations are distributed independently of the projected variations [17,18]. From the latter, it is understood that we try to reject the null hypothesis. That is to say, that the value DA is not between the critical values mentioned and that, therefore, there is a predictive capacity.

3.2 Codification of Variables

Each chromosome has several genes, which correspond to the parameters of the problem. To work computationally with the genes is necessary to encode them in a string (i.e., in a sequence of symbols composed, in this case, of zeros and ones).

For correct codification and good resolution of the problem. It was constructed by matrix blocks of the Dow_t, Cu_t and ε_t lags. We used the heuristic rule called the building blocks rule, that is, related parameters that must be close to each other on the chromosome.

Chromosome [1010][1110][1101], would be represented by a three-block string, where the first four-gene block is represented by the Dow_t at $t-1$, $t-2$, $t-3$ and $t-4$ lags. The second block of four genes ranging from gene five to gene eight is represented by Cu_t lags in its lags from $t-1$ to $t-4$. For the third block of four genes ranging from gene 9 to gene 12 is represented by ε_t in the prediction for $t-1$ to $t-4$.

3.3 Generation of Initial Population and First Generation

A population of fixed-sized chromosomes of only thirty individuals has been chosen arbitrarily, considering the calculation times of our computational resources and privileging the search and optimization based on the Darwinian theory.

Our initial population, or first generation of chromosomes, is generated from a random generation. The genetic algorithms being tools to obtain approximate solutions to problems in which to evaluate the exact resolution would be very costly in time. So, It is necessary to program in a way that does not allow the twins (siblings alike) not to evaluate more than once the same model. Also, it is advisable not to generate or reproduce a null chromosome, composed of zeros for the three blocks, represented as [0000][0000][0000].

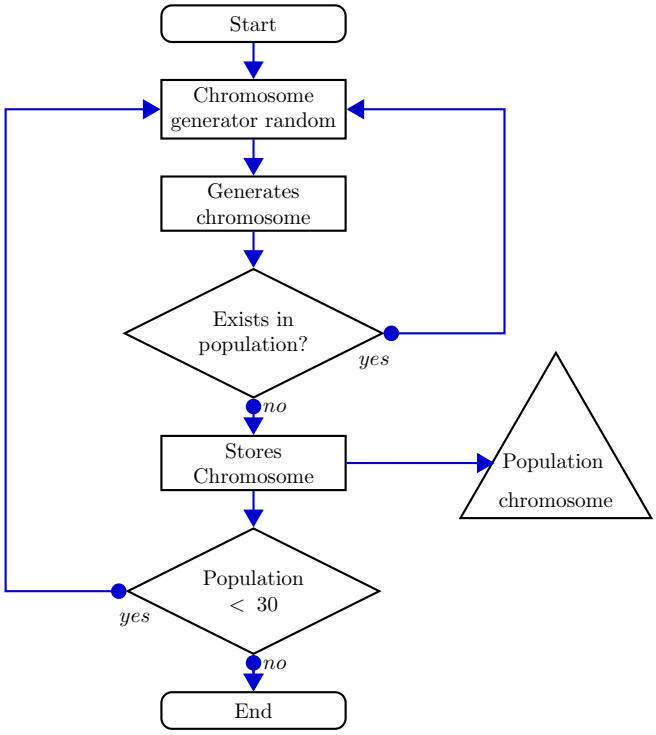


Fig. 3: Initial population generation.

From an initial population of Cu chromosomes generated randomly from thirty models, as shown in Figure 3. The ten best chromosomes are chosen by the aptitude assessment function (in an elitist manner), which are assigned to the second generation to be reproduced until it completes again thirty, as shown in the Figure 2. In this way, elitism can rapidly increase the performance of a genetic algorithm since this avoids losing the best solution found [19]. Then, it is possible that this method quickly leads to a local optimum.

3.4 Evaluation and Selection

The evaluation function fitness, according to equation 14, plays an essential role in the potential classification of solutions in terms of their characteristics. It is the criterion evaluation of the quality of the individuals.

$$Fitness = \frac{PSP_{\max} + Profitability}{2} \tag{14}$$

Where:

$Fitness$ is the multi-objective evaluation function,

$PSP_{\max}$ is the maximum PSP between PSP and $(1 - PSP)$, as shown in Equation 7,

$Profitability$ is the individual's or chromosome's profitability.

It provides higher power and robustness to the search technique. And this operator fulfils the function of making a selection of the best individuals so that they are considered in the process of generation of the new population [19]. The genetic algorithm uses the elitist selection technique to select the individuals to be copied to the next generation. The technique ensures the selection of the most suitable members of each generation and preserves them to deliver their attributes to their descendants. That is, those selected as reproducers of the next generation. During the evaluation, the gene is decoded, turning it into a series of parameters (presented at the coding point of the variable). Finally, a solution is obtained with the best score based on the best performance.

3.5 Mutation, Reproduction and Stop

5% was assigned for the mutation function, with the restriction of not allowing duplicate individuals, which increases the effective mutation rate after each generation. In gene transfer, the father provides the first two genes of each block, and the remaining two are the mother, which are selected for random reproduction among the best individuals. The stopping criterion was applied upon completion of evaluating the seventeenth generation.

4 Results

4.1 The Best Models

According to Table 1, the best model [0100][1100][0111] are born in the twenty-first generation both with a predictive capacity (PSP) of 67.12% and a profitability of the period of 9.02% .

In the results of positions 2 to 5 can be observed in Table 1.

Figure 4 shows the evolution of the relative price of Cu and the best individual [0100][1100][0111], showing the best result of 9.02% of the active management of chromosome [0100][1100][0111] versus the Buy and Hold strategy of 0.45% represented by Cu. Active management with the support of genetic

Table 1: Top five models.

Chromosome	$PSP_{\max}$	Profitability	Generation
[0100][1100][0111]	67.12%	9.02%	21^o
[0100][1000][0011]	65.75%	8.03%	17^o
[0011][1000][0000]	61.64%	10.23%	5^o
[0100][1100][0011]	64.38%	7.32%	19^o
[0111][0001][0000]	61.64%	9.89%	5^o

algorithms improves the profitability result. We can see that with active portfolio management, buying and selling according to the signal variation of the price of the positive or negative Cu, a yield of 9.02% is obtained at the end of the evaluation period. Represented by the best individual [0100][1100][0111].

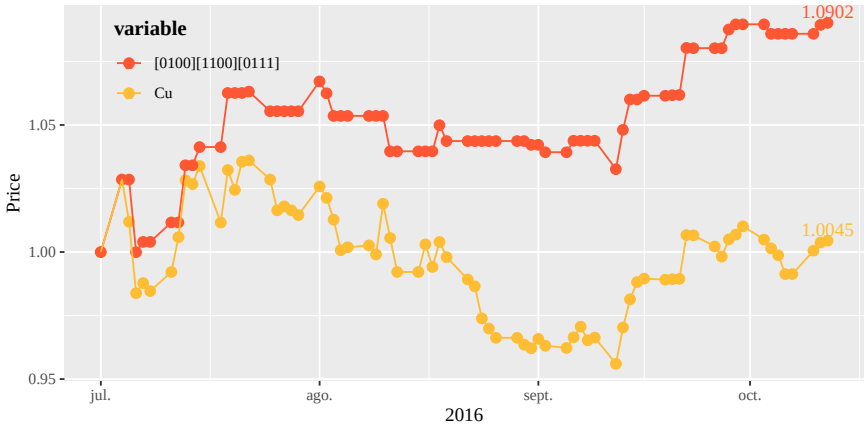
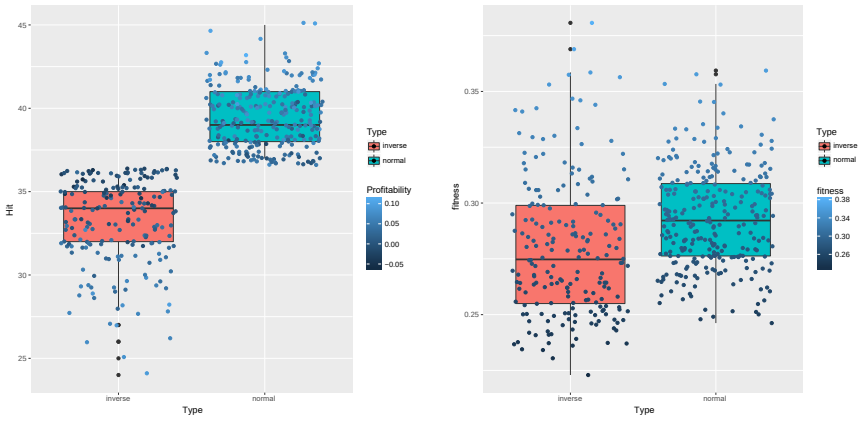


Fig. 4: Results Comparison.

4.2 Findings

The $PSP_{\max}$ of the best results of the individuals selected for their best fitness in each generation, turned out to be maximum $(1 - PSP)$ on PSP as shown in equation 7 which is part of equation 14, and can be observed in Figures 5a and 5b [8].

The importance of finding and using an individual with a low level of success (<50%), it allows being classified as a liar or inverse, which is shown in Figure 5a. Identifying and measuring the liar as $(1 - PSP)$, has allowed us to obtain better results with some individuals. It can be seen in Figure 5b, for the distribution of fitness for the inverse and normal groups.



(a) Hit. (b) Fitness.

Fig. 5: Normal and Inverse.

5 Conclusions

The application of genetic algorithms to the Cu price prediction models effectively allowed an improved model. Our results showed that the chromosome [0100][1100][0111] obtained a maximum PSP of 67.12% and a yield of 9.02%, versus a "Buy and Hold" return of 0.45%, in 73 days for investment.

It is worth mentioning that the first generations do not show good results, however, reproduction and mutation achieve an evolution of generations with desired results. The above described makes it possible to obtain forecast models of the copper price variation with greater precision.

In addition, the models showed the highest accumulated profitability during the evaluation period. Thus, the models constructed from genetic algorithms presented a statistically significant predictive capacity, as demonstrated by the results of the directional accuracy test of Pesaran & Timmermann.

Acknowledgement

The authors are grateful for the financial support of the projects Fondef/Conicyt IT17M10012 and STIC-AmSud 19-STIC-08.

References

1. Carrasco, R., Vargas, M., Soto, I., Fuentealba, D., Banguera, L., Fuertes, G.: Chaotic time series for copper's price forecast: neural networks and the discovery of knowledge for big data. In: Liu, K., Nakata, K., Li, W., Baranauskas, C. (eds.) Digitalisation, Innovation, and Transformation, IFIP Advances in Information and Communication Technology, vol. 527, pp. 278–288. Springer, Cham, London, UK (jul 2018). https://doi.org/10.1007/978-3-319-94541-5_28

2. Carrasco, R., Vargas, M., Alfaro, M., Soto, I., Fuertes, G.: Copper Metal Price Using Chaotic Time Series Forecating. IEEE Latin America Transactions **13**(6), 1961–1965 (2015). https://doi.org/10.1109/TLA.2015.7164223
3. Carrasco, R., Astudillo, G., Soto, I., Chacon, M., Fuentealba, D.: Forecast of copper price series using vector support machines. In: 2018 7th International Conference on Industrial Technology and Management (ICITM). pp. 380–384. IEEE, Oxford, UK (2018). https://doi.org/10.1109/ICITM.2018.8333979
4. Seguel, F., Carrasco, R., Adasme, P., Alfaro, M., Soto, I., Adasme, P., Soto, I., Alfaro, M., Soto, I.: A Meta-heuristic Approach for Copper Price Forecasting. In: Liu, K., Nakata, K., Li, W., Galarreta, D. (eds.) Information and Knowledge Management in Complex Systems, IFIP Advances in Information and Communication Technology, vol. 449, pp. 156–165. Springer International Publishing, Toulouse, France (2015). https://doi.org/10.1007/978-3-319-16274-4_16
5. Hackathorn, R.: The BI Watch Real-Time to Real-Value. DM REVIEW (2004)
6. Holland, J.H.: Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press (1975)
7. Hsu, C.M.: A hybrid procedure for stock price prediction by integrating self-organizing map and genetic programming. Expert Systems with Applications **38**(11), 14026–14036 (may 2011). https://doi.org/10.1016/j.eswa.2011.04.210
8. R Core Team: R: A Language and Environment for Statistical Computing (2016)
9. Guerino, M.: Técnicas avanzadas para la predicción de la variación de IFN. Tesis para optar al grado de magíster en finanzas, Universidad de Chile (2006)
10. Guerrero, V.: Análisis estadístico de series de tiempo económicas. International Thomson, México, D.F., 2a. edn. (2003)
11. Parisi, A., Parisi, F., Díaz, D.: Forecasting gold price changes: Rolling and recursive neural network models. Journal of Multinational Financial Management **18**(5), 477–487 (dec 2008). https://doi.org/10.1016/j.mulfin.2007.12.002
12. Parisi, A., Parisi, F., Cornejo, E.: Algoritmos genéticos y modelos multivariados recursivos en la predicción de índices bursátiles de américa del norte: IPC, TSE, NASDAQ y DJI. Fondo de Cultura Económica **71**(284(4)), 789–809 (2004)
13. Ahn, Y.B., Tsuchiya, Y.: Directional analysis of consumers' forecasts of inflation in a small open economy: evidence from South Korea. Applied Economics **48**(10), 854–864 (2016). https://doi.org/10.1080/00036846.2015.1088144
14. Anatolyen, S., Gerko, A.: A Trading Approach to Testing for Predictability. Journal of Business & Economic Statistics **23**(4), 455–461 (2005)
15. Pesaran, M.H., Timmermann, A.: A Simple Nonparametric Test of Predictive Performance. Journal of Business & Economic Statistics **10**(4), 461–465 (1992). https://doi.org/10.2307/1391822
16. Tsuchiya, Y.: Do production managers predict turning points? A directional analysis. Economic Modelling **58**, 1–8 (2016). https://doi.org/10.1016/j.econmod.2016.05.019
17. García, I., Trigo, L., Costanzo, S., ter Horst, E.: Procesos gaussianos en la predicción de las fluctuaciones de la economía mexicana. El Trimestre Económico **77**(3), 585–603 (2010)
18. Trigo, L., Costanzo, S.: Redes neuronales en la predicción de las fluctuaciones de la economía a partir del movimiento de los mercados de capitales. El Trimestre Económico **74**(74), 415–440 (2007)
19. Marczyk, A.: Algoritmos genéticos y computación evolutiva. Departamento de Informática, Universidad de Colorado (2004)

On the stress of testing credit default.

Viani Djeundje Biatat and Jonathan Crook
The University of Edinburgh (UK)

Abstract

Analyzing and predicting credit default continues to attract a lot of interest, both in academia and among practitioners.

In this work, we look at uncertainty and stress using account-levels auto-regressive models within a survival analysis framework.

Within this framework, we (i) identify a number of components contributing to the distribution of credit loss, and then (ii) quantify the impact of each component on standard stress metrics.

An Automated Lane- Change Strategy for Autonomous Vehicles Based on QoS Forecasting

Jamal Raiyn
Computer Science Department, Al Qasemi Academic College. Baqa Al Gharbiah, Israel
raiyn@qsm.ac.il

Abstract: A large number of accidents are caused by human lane change behaviours. Delays in drivers' reaction times and errors in judgement are the main causes of the vast majority of accidents. Autonomous vehicles (AVs) can decrease the occurrence of the accidents through platooning, as well as increase road capacity. *Platooning* is defined as the gathering together of AVs, which interact with each other to coordinate the safe joining, exiting, and changing lanes. It manages the traffic on urban roads by reducing distance between AVs as much as possible. This paper discusses the impact of the lane changing of AVs in platooning congestion. A strategy is introduced for successful lane changing using V2I communications and intelligent speed assistance (ISA). ISA forecasts the travel speed in real-time and detects when traffic congestion is due to an accident or repeated failure to change lanes. The performance of the V2I communication is measured by QoS parameters, such as delay and interference. Latency and inter-carrier interference can have an impact on cooperative communication based V2I; therefore, an automated lane change strategy is proposed to overcome the limitations of cooperative Communication.

Keywords: autonomous vehicle, platooning, V2X, lane change behavior.

1. Introduction

The aim of this paper is to increase the capacity of highways by automatically coordinating and controlling vehicles to form platoons (Zhao & Sun, 2013) in which vehicles are kept at a small distance from each other. There are real-time communications both within and between vehicles (Sun et al. 2016). A new challenge faced by the designers of AVs is delays, which are caused by many factors. Furthermore, in AV networks, information is transferred through vehicle-to-vehicle (V2V) (Bergenhem, 2012) and vehicle-to-infrastructure (V2I) communication channels. We have added a new communication type V2IoT, which is used to communicate with internet of things devices (V2X) (Weiß,2011; Schünemann,2011; Muhammad & Safdar, 2018) over a fifth-generation (5G) wireless system (Mitra & Agrawal, 2015; Dey, 2016).

For the most part delays and human lane changing behaviours caused most traffic accidents. Delays in drivers' reaction time and drivers' judgement error are key causal factors in the vast majority of accidents. However, autonomous vehicles (AVs) can decrease accidents, as well as, increase road capacity, through the use of platooning, which reduces the distance between AVs as much as possible and gathers together groups of AVs, which interact with one another to coordinate the safe joining, exiting, and changing of lanes.

This paper discusses the lane changing of AVs in platoon and introduces a strategy for successful lane changing. The process is based on the use of V2X communications, the detection of AV positioning and forecasting of AV speed in real-time. The performance of the V2X communication is measured by QoS parameters, such as delay and interference. Latency and inter-carrier interference can have an impact on cooperative communication based V2X. An automated lane changing strategy is proposed to overcome the limitations of cooperative communication. Interference should be kept to a minimum and below the acceptable threshold. Increased delays may affect the performance metrics of a positioning terminal, which are characterized in terms of availability, accuracy, and integrity. Many untargeted transmitted signals can even interfere with transmitted signals (Wang et al., 2013; Ancans et, al. 2017). Delays may be caused by the weak performance of network equipment and the heterogeneity of equipment attributes. The proposed solution is based on maintaining a minimal distance between AVs (Lam et al., 2016). The novel aspect of this study is the proposed system model for automated lane changing in platoon (Atagoziyev et al., 2016). Given different lane changing environments, it is necessary to accurately estimate the positioning AVs and to forecast their real-time travel speed.

The rest of this paper is organized as follows: Section 2 gives an overview of related research; section 3 describes the lane changing strategy in platooning; section 4 and 5 discuss the performance evaluation and conclude the paper.

adfa, p. 2, 2011.

2. Related Research

In the academic literature, there is no a consistent definition for the term *platooning*. However, most definitions are similar. Hall and Chin (2005) defined *platoon* as a number of vehicles that travel on a highway in a closely spaced group. Bergenheim et al. (2012) defined it as a collection of vehicles that travel together, actively coordinated in formation. In the literature, however, some relevant aspects of platooning are not covered. For example , V2V, V2X, V2I communications are needed to manage platooning activities.

Dey et al. (2016) used Het-Net to support connected vehicle applications based on V2V and V2I communications. Sepulcre and Gozalvez (2018) present an architecture for context- aware heterogeneous V2I communication in vehicular networks to improve the quality of service and satisfy vehicular application requirements. Brandl (2016) introduced an initial proof of concept for future connected vehicular landscapes, focusing on the basics of "vehicle-to-everything" (V2X) communication from vehicle to vehicle and from vehicle to infrastructure. Weiß (2011) introduced a field operational test simTD, which is the first of its kind to evaluate the effectiveness and benefits of applications based on vehicular communication.

Jin et al. (2013) proposed an improved multi-agent intersection management system, in which vehicle agents may form platoons using connected vehicles technologies. Compared to the conventional traffic signal control system, the proposed platoon-based multi-agent intersection management system can shorten average travel time and reduce fuel consumption.

Gora and Rüb (2016) proposed self-driving and connected vehicles, communicating with one another (V2V technology) and with the road infrastructure (V2I technology), and they designed a microscopic traffic simulation model for such vehicles, including a robust protocol for exchanging information. Li (2016) proposed a cooperative traffic control algorithm based on vehicle-to-Infrastructure (V2I) connections to reduce traffic delays and decrease fuel consumption. Bergenhem (2012) described a vehicle-to-vehicle (V2V) communication system that enables vehicles to drive in platoons. Jiang et al. (2010) presented a practical model for characterizing V2V communication channel and the impact of inter-carrier interference (ICI) generated in orthogonal frequency division multiplexing (OFDM).

Sun et al. (2016) investigated the radio resource management problem for D2D-based V2V communication and proposed direct device-to-device (D2D) links a possible enabler for vehicle-to-vehicle (V2V) communications, where incurred intracell interference and the stringent latency and reliability requirements are challenging issues. Du and Dao (2015) proposed an analytical formulations to estimate information propagation time delay via a V2V communication network serving a one-way or two-way road segment with multiple lanes. This technical view of platooning describes inter

platooning interactions based on V2X communication and examines the maintaining of a platoon, for instance, while AVs are joining, leaving, or changing lanes. Furthermore informative speed assistance and AV positioning estimation are essential for platoon control.

2. Lane Changing Strategy

An AV that would like to communicate with other vehicles must be held to a minimal distance from them in order to avoid interference and latency. This cooperation should take place within the coverage area. In other words, the vehicles should maintain a minimum distance and should remain within transmission range in the same zone. In V2X communication, the quality of service (QoS) parameters should be controlled, especially for delay. Delay is the second most prevalent cause of interference and lead to increased traffic congestion.

The lane change process in platooning can be divided into three phases: informative speed assistance (ISA), AV positioning estimation, and cooperative communication based message exchange, as illustrated in Figure 1.

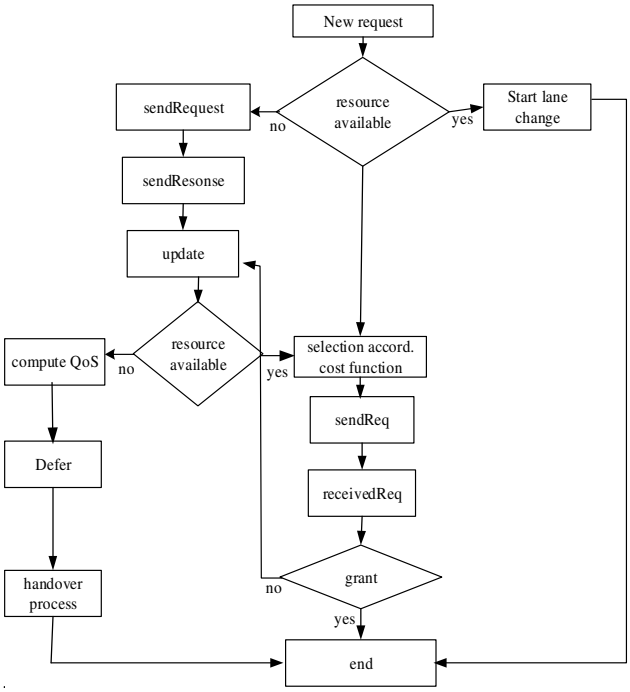


Figure 1: Lane changing process

2.1 Communication within Platoons

The maturing of wireless communication technologies (vehicular communication ITS-G5 - 802.11p, cellular 4G or 5G, etc.) (Ancans et al., 2017) at affordable costs and with the right levels of security will enable the wide deployment of so-called connected vehicles. All of the ITSs based on cooperation and communication between vehicles or between vehicles and infrastructures are called cooperative systems (Sun et al., 2016; Muhammad & Safdar, 2018).

It is usual to separate the various communication technologies (Schünemann, 2100; Wang et al., 2018; Choudhury et al. 2016) supporting cooperative systems into two groups:

- Vehicle to vehicle (V2V) communication systems, which enable safer transport, for example, they mediate safe distance keeping, collision avoidance, and early warnings of unsafe conditions.
- vehicle to infrastructure (V2I) communication systems (and vice versa), which enable better use of existing infrastructure and provide valuable and consolidated information to intelligent vehicles for example improved information regarding travel times, ongoing roadwork, and weather and traffic conditions, and up-to-date information about parking availability or other means of transport.

Altogether, information, communication and positioning technologies will play a key role in future transport systems and services. For instance, floating car data are mainly composed of position and/or speed information, and the cooperative awareness messages (CAM) exchanged by the ITS stations in V2V or V2V communications all contain the reference position of the station emitting the message (Maimaris & Papageorgiou, 2016). The success of positioning will depend on the system capacity for better performance (including improvement and control), its tight integration with other ITS technologies in smart multi-service and multi-standards platforms, and the affordability of relevant services. Furthermore, to improve drivers' safety, V2V systems share information relating to traffic information and accident warning with nearby vehicles and road infrastructures. However, the incorporation of these new systems into vehicles increases of security risks. For example, some of the latest models can be hacked within 360 seconds; the actuators of modern vehicles can be remotely controlled; terrorists can potentially hack into V2V and V2I systems to cause chaotic traffic accidents (e.g., by hacking into an autonomous intersection system (Jiang et al., 2010); and privacy information can be stolen from any driver. AVs use vehicle–to-vehicle communication for cooperative merging on the highway: the lane changes of platoons based on cooperative communication is illustrated in Figure 2.

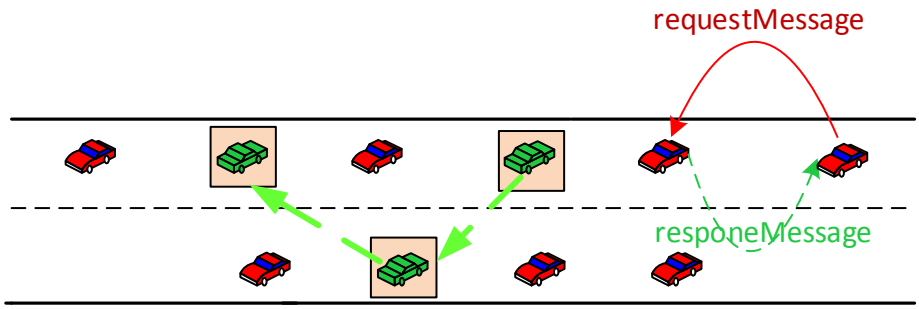


Figure 2: Platooning lane change

2.2. ISA in Platooning

Intelligent speed assistance (ISA) makes a distinction between traffic congestion that occurs due to an accident and one that results from a failed lane changing process. The following section characterizes traffic congestion that is caused by an accident.

2.2.1 Forecasting in Real Time

The occurrence of abnormal conditions in traffic flow travel information decreases the reliability of forecasts based on historical information and may increase the complexity of the forecasting of unusual incidents. A forecasting model that is based on real-time information, gives a little weight to historical information but great weight to real-time observations.

$$tt(t+1,k) = tt^{H}(t+1,k) + \gamma * (tt^{M}(t,k) - tt^{H}(t,k)) \quad (1)$$

where $0 < \gamma < 1$.

2.2.2 The Mutual Influence of Sections

In real-time forecasting, the effect of the upstream (UP) and downstream (DS) are considered.

$$tt(t+1,k) = tt^{H}(t+1,k) + \gamma_1 * desired + \gamma_2 * UP + \gamma_3 * DS \quad (2)$$

where

$$desired = \left| tt^{M}(t,k) - tt^{H}(t,k) \right|$$

$$upstearm = \left| tt^{M}(t,k-1) - tt^{H}(t,k-1) \right|$$

$$downstream = \left| tt^{M}(t,k+1) - tt^{H}(t,k+1) \right|$$

k is the desired section, *(k-1)* is the upstream section, and *(k+1)* is the downstream section.

An incident occurring on section *i* within time interval *t* is considered to have a significant impact on traffic when traffic measurements from the upstream and downstream stations satisfy the following conditions:

i. the difference between the upstream speed *si,t* and the downstream speed *si+1,t* is greater than the threshold value;

ii. the ratio of the difference between the upstream and downstream speeds to the upstream speed (*si ,t - si+1,t*)/*si,t* , is greater than the threshold value; and

iii. the ratio of the difference between the upstream and downstream speeds to the downstream speed (*si,t - si+1,t*)/ *si+1,t* is greater than the threshold value.

An abnormal record is one that reports a traffic speed that is at least 30 km/h lower traffic speed slower than the average speed of all records at the same time on the same day of the week. The threshold of 30km/h is a symbolic value of the smallest speed change that people would consider "abnormal". The vehicle speed starts to decrease upstream, while the downstream speed starts to increase.

$$\frac{tt(k,t)-tt(k+1,t)}{tt(k,t)} > threshold \quad (3)$$

$$\frac{tt(k,t)-tt(k+1,t)}{tt(k+1,t)} > threshold \quad (4)$$

2.2.3 Forecasting in Accident Situations

The travel time forecasting model considers incident and non-incident conditions. We distinguish among

- accidents during peak travel times (morning/afternoon)
- accident during regular hours
- heavy accidents and
- light accidents.

The accident is cleared at current time *t* in section s, the duration is known and the speed is considered to be reduced by 30 km/h below the average speed.

$$tt(t+1,k)=tt^{H}(t+1,k)+\gamma*(P_t)*(tt_t^{M}-tt_t^{H}) \quad (5)$$

$$P_t = P(accident)_t = \frac{1}{1+e^{-\upsilon_t}},\ \upsilon_t = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

$$x_1 = \frac{\left(\sigma_t - \sigma_t^H\right)}{\sigma_t^H}, x_2 = \frac{\left(tt_t - tt_t^H\right)}{\sigma_t^H}, x_3 = \frac{\left(tt_t - tt_t^H\right)}{\sigma_t^H} - \frac{\left(tt_{t-1} - tt_{t-1}^H\right)}{\sigma_{t-1}^H},$$

$$x_4 = \frac{\left(\sigma_t - \sigma_t^H\right)}{\sigma_t^H} - \frac{\left(\sigma_{t-1} - \sigma_{t-1}^H\right)}{\sigma_{t-1}^H}$$

where X denotes the vector of the predictor variables. β is the vector of the coefficient associated with the predictor variables and can be computed according to the binary logit model. vt is the logit link function (which is a linear combination of the predictor variables).

The ISA algorithm detects road accidents based on travel time variations. We consider accidents during peak periods (i.e. morning or late afternoon) and during non-peak periods. Equation 4 is used to forecast the accident scheme.

$$tt_{acc}^F(t+1,k) = EMA_{acc}^H(t+1,k) + \delta(tt_{acc}^M(t,k) - tt^H(t,k)) \quad (6)$$

where, $0 < \alpha \leq 1$, $0 < \delta < 1$, $tt^M(t, k)$ is the actual travel time in section k at time t, and $tt^H(t, k)$ is the historical travel time in section k at time t.

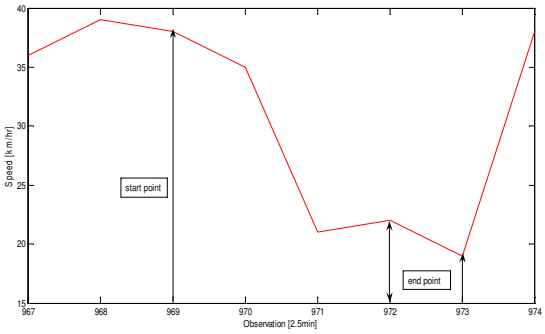


Figure 3: Accident detection

Table 1: Comparing of EMA and Real Observations

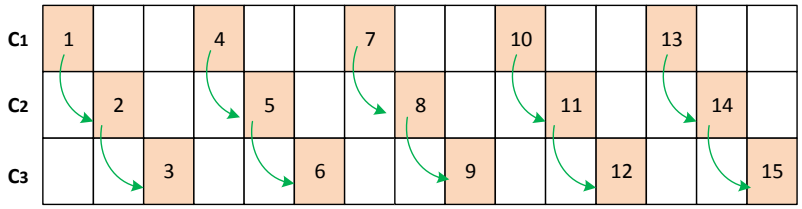
accident condition	EMA	Real-Time

mean data	46.402	46.402
mean prediction	44.779	52.536
std data	11.642	11.642
std prediction	14.012	9.3218
Observations with error over 5 km/hr	39.565	74.009
Observations with error over 10 km/hr	24.974	32.41
max abs. error	53.022	36.362
max. relative error	267.91	243.58
mean error	1.6225	-6.1346
mean abs. error	8.0107	8.831
mean relative error	20.728	24.458
root mean squared error	13.749	10.663
root mean squared percent error (1)	37.311	37.867
root mean squared percent error (2)	29.63	22.981

2.3 Lane Changing Strategy for Platooning

Platooning defined as a collection of traveling together. The platooning of AVs is mediated by intelligent cooperative communication. Platooning uses V2X communication to exchange information in order to improve traffic management and increase human safety on urban roads. Platooning improves traffic management by introducing a new strategy based on dividing the urban road into virtual sections called *resources*. The urban road (UR) is partitioned into k sections: $C_1, C_2, ..., C_k$, such as $UR = \bigcup_{i=1}^{k} C_i$ and $C_i \cap C_j = \phi$, $1 \le i, j \le k$, where C_i is considered the road section class i. Furthermore, we assume that $C_i = \{AV \mid AV \bmod k = i\}$. The platooning system detects which resources are free expressed as $Free_i$ and otherwise as $busy_i$.

The free resources are not always available for other AVs. Sometimes, the AVs are hindered from using a free road section due to high latency and interference in V2X communication.



2.3.1 Algorithm description

Algorithm I presents the message exchange strategy, which is based on *SendMessage* and *ResponseMessage*. *SendMessage* indicates that the sending AV is requesting information about a place for a lane change. *ResponseMessage* indicates that the sending AV is receiving an answer from the receiving AV. The response message grants the request or informs AV that the resource is being used. Algorithm II describes intelligent speed assistance (ISA). ISA estimates the AV's speed and positioning. Algorithm III describes the process of lane changing for the platoon. When AV encounters a mandatory lane change, the AV will change lanes if the target lane has sufficient space; otherwise, the vehicle will only be able to stop and wait for the next lane change opportunity.

Algorithm I: Communication V2I

For all AVs **do**
SendRequestMessage
AV$_i$ sendResponseMessage
If Resource is Free
AV enter lane change process
else
wait new updates
end if
end for

Algorithm II: ISA

for all AV$_i$ **do**
At $ts_{1,}$ leader computes its forecasting speed
At $ts_{1,}$ leader sends its next acceleration and speed forecasting
At $ts_{j,}$ leader gets from IVA$_j$ computed next acceleration to use at the next updating cycle
End for

Algorithm III: Lane Changing Process

```
While The AVs in platooning do
Send request message to head AV_k
  If AV_k Receive replay then
     Change lane
   else
      if No response then
         No lane change
      end if
  end if
end while
```

3 Performance Evaluation

The platooning concept describes the interaction of AVs with other platoons, e.g., joining, leaving, or changing lane. To manage the platoon, it is necessary to use intelligent speed assistance and AV positioning estimation. In the lane changing process, collisions may be caused by faulty human decision making and bad estimations of the road traffic environment including uncooperative drivers. Traffic education and behaviour vary from community to community.

To improve traffic efficiency and ensure safety, a lane changing strategy is proposed. The process is mediated by ISA, which alerts the AVs with updated speed limits. Based on speed assistance, the lane changing strategy controls the free resources. An AV starts the lane changing process when ISA alerts all the AVs that it has detected a free resource in the target lane with enough space. The automated lane change strategy uses an *xor* logical operator to detect a free resource, defines as follows

$$J = X \oplus Y$$

$$J(x) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{otherwise} \end{cases}$$

A new request for a lane change is rejected when ISA is informed that there is traffic congestion or that the QoS requested cannot be provided as the delay is greater than the threshold $Delay^{tgt}$. We consider four AV decision scenarios, described as follows:

i. Probability (Approve, Delay < $Delay^{tgt}$ && $AV_{speed} > ISA_{speed}$)

ii. Probability(Reject, Delay > $Delay^{tgt}$ || $AV_{speed} < ISA_{speed}$)

From the viewpoint of an AV, there are two kinds of decisions: a good and a bad. A good decision is made when the AV can use a viable section without congestion; a bad decision is made when the use of a viable section may cause congestion. The two scenarios (AV beliefs) can be expressed as follows: The vehicle assignment can be expressed and denoted by X, where $X \in \{1,0\}$, which means, X = 1 or X = 0, denoting, respectively a lane change granted or rejected. The AV decision can be expressed as follows:

$$Y(decision) = \begin{cases} 1 & \text{aquire the section } k \\ 0 & \text{otherwise} \end{cases}$$

QoS parameters like signal-to-noise ratio (SNR) and delay influence the lane changing process. SNR is a measure used to compare the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to noise power as illustrate in Figures 7 and 8. When the lane changing process ends, traffic congestion increased as illustrates Figure 9.

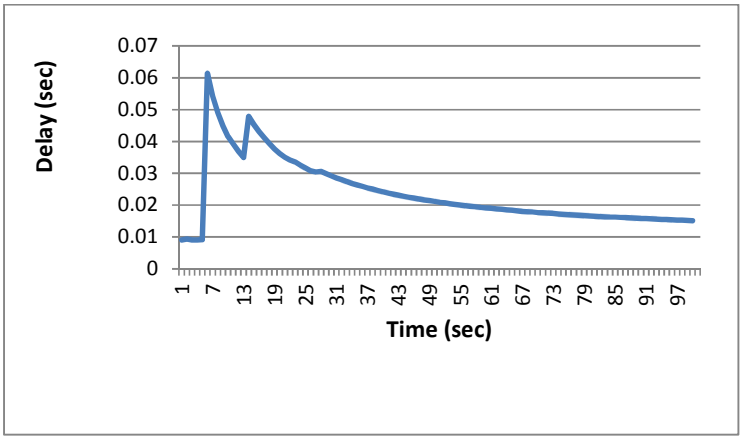


Figure 7: Delay

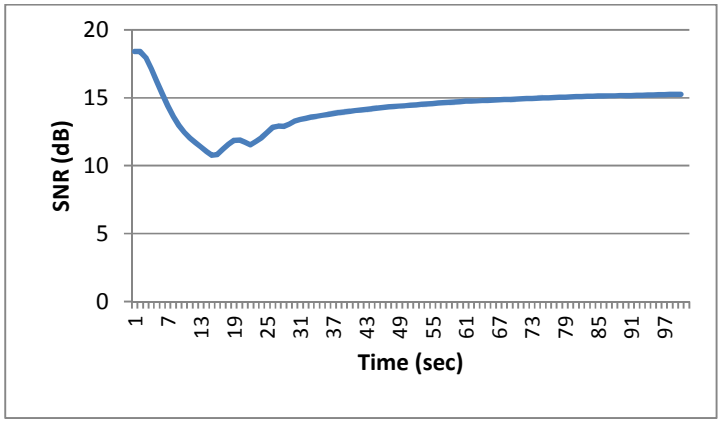


Figure 8: Signal-to-Noise Ratio

Figure 9 shows that the speed is increased in sections that used a cooperative communication strategy. In sections where the lane change failed, the speed is decreased and congestion results.

Figure 10 shows that ISA has detected that the traffic congestion was caused by an accident.

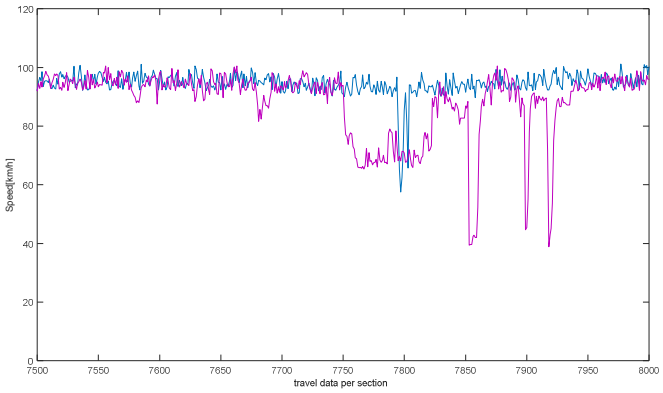


Fig. 9: Detection of failed lane change

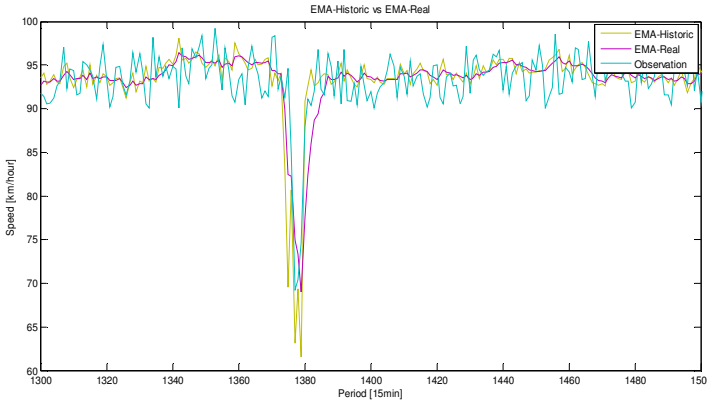


Figure 10: Comparison of EMA-H and EMA-H, EMA-R

4 Conclusion

This paper has discussed a platooning scenario for AVs. Furthermore, state of the art intelligent cooperative communication in platooning was reviewed.

The study investigated in detail how ISA characterizes traffic congestion that is caused by accidents or failed lane changes in platooning. The lane change process can be aborted due to delay and interference that can influence communication in AV networks. In cooperative communication, delay and interference are the main reasons for stopping the lane changing process. The rejection of a high number of demands for lane changes causes congestion in platooning.

To overcome the limitations of the communication, this work demonstrates an idea based on automated lane changing by AVs. The proposed lane changing process is based on ISA.

The ISA introduced here possesses new detection capabilities that can decrease traffic congestion in platooning.

References

Ancans, G., Bobrovsa, V., Ancansb, A., Kalibatiene, D. 2017. Spectrum Considerations for 5G Mobile Communication Systems, *Procedia Computer Science*, 104. pp. 509 – 516.

Atagoziyev, M., Schmidt, K. W., Schmidt, E.G. 2016. Lane Change Scheduling for Autonomous Vehicles, *IFAC* 49-3. pp. 061–066.

Bergenhem, C., Hedin, E., Skarin, D. 2012. Vehicle-to-Vehicle Communication for a Platooning System, *Procedia - Social and Behavioral Sciences*, 48, pp. 1222 – 1233.

Brandl, O. 2016. V2X traffic management, *Elektrotechnik & Informationstechnik*. 133/7: pp. 353–355.

Choudhury, A., Maszczyk, T., Math, C. B., Li, H., Dauwels, J. 2016. An integrated simulation environment for testing V2X protocols and applications, *Procedia Computer Science*, Vol. 80. pp. 2042–2052.

Dey, K. C. et al. 2016. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network – Performance evaluation, *Transportation Research Part C* 68. pp. 168–184.

Du, L., Dao, H. 2015. Information Dissemination Delay in Vehicle-to-Vehicle Communication Networks in a Traffic Stream, *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, VOL. 16, NO. 1. pp. 66-80.

Gora, P., Rüb, I. 2016. Traffic models for self-driving connected cars, *Transportation Research Procedia* 14. pp. 2207 – 2216

Hall, R., Chin. C. 2005. Vehicle sorting for platoon formation: Impacts on highway entry and throughput, *Transportation Research Part C: Emerging Technologies,* 13, 5–6. pp. 405–420.

Jin, Q., Wu, *G.,* Boriboonsomsin, K., Barth, M. 2013. Platoon-Based Multi-Agent Intersection Management for Connected Vehicles, 16th International IEEE Conference on Intelligent Transportation Systems.

Jiang, T., Chen, H.-H., Wu, H.-C., Yi, Y. 2010. Channel Modeling and Inter-Carrier Interference Analysis for V2V Communication Systems in Frequency-Dispersive Channels, *Mobile Network Application*, 14, pp. 4–12.

Lam, S., Taghia, J., Katupitiya, J. 2016. Evaluation of a transportation system employing autonomous Vehicles, *JOURNAL OF ADVANCED TRANSPORTATION*, 50: pp. 2266–2287

Li, J., Dridi, M., El-Moudni, A. 2016. A Cooperative Traffic Control of Vehicle–Intersection, *(CTCVI) for the Reduction of Traffic Delays and Fuel Consumption*. Sensors. 16. pp. 1-20.

Maimaris, A., Papageorgiou, G. 2016. A Review of Intelligent Transportation Systems from a Communications Technology Perspective, *IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, November 1-4, 2016.

Mitra, R. N., Agrawal, D.P. 2015. 5G mobile technology: A survey, *ICT Express*, 1. pp. 132–137.

Muhammad, M., Safdar, G.A. 2018. Survey on existing authentication issues for cellular-assisted V2X communication, *Vehicular Communications*, 12. pp. 50–65.

Schünemann, B. 2011. V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems, *Computer Networks*, 55. pp. 3189–3198.

Sepulcre, M., Gozalvez, J. 2018. Context-aware heterogeneous V2X communications for connected vehicles, *Computer Networks*, 136. pp. 13–21.

Sun, W., Ström, E.G., Brännström, F., Sou, K.C., Sui, Y. 2016. Radio Resource Management for D2D-Based V2V Communication, *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, VOL. 65, NO. 8. pp. 6636- 6650.

Raiyn, J. 2017(a). Road traffic congestion management based on search allocation approach, *Transport and Telecommunication*. 18. (1). pp. 25-33.

Raiyn, J. 2017(b). Developing Vehicle Locations Strategy on Urban Road, *Transport and Telecommunication.* 18. (4). pp. 253–262.

Wang, Y., Duan, X., Tian, D., Lu, G., Yu, H. 2013. Throughput and Delay Limits of 802.11p and Its Influence on Highway Capacity, *Procedia - Social and Behavioural Sciences*, 96. pp. 2096 – 2104.

Wang, T., Zhao, J., Li, P. 2018. An extended car-following model at unsignalized intersections under V2V communication environment, *PLoS ONE,* 13(2): e0192787. pp. 1-13.

Weiß, C., 2011. V2X communication in Europe – From research projects towards standardization and field testing of vehicle communication technology, *Computer Networks*, 55. pp. 3103–3119.

ZHAO, L., SUN, J. 2013. Simulation Framework for Vehicle Platooning and Car-following Behaviors under Connected-Vehicle Environment, *Procedia - Social and Behavioural Sciences* 96. pp. 914 – 924

Stochastic Analysis and Modeling of Local Temperature Fluctuations

Faeze Minakhani[1] and M. D. Niry[2,3]

[1] Department of Physics, Faculty of Science, University of Zanjan, Zanjan 313, Iran
minakhani@alumni.znu.ac.ir
[2] Department of Physics, Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan 45137-66731, Iran
[3] Center for Research in Climate Change and Global Warming (CRCC), Institute for Advanced Studies in Basic Sciences (IASBS), Zanjan 45137-66731, Iran.
m.d.niry@iasbs.ac.ir
http://www.iasbs.ac.ir/∼m.d.niry/

Abstract. We examine the Markov property of temperature fluctuation in four meteorological stations. Using stochastic analysis, we characterize the complexity of the temperature fluctuation by means of the single and multi-variable Fokker-Planck equation and find the Kramers-Moyal coefficients. The Langevin equation enables us to reconstruct temperature time series with similar statistical properties compared with the observed fluctuations in real temperature sequence.

Keywords: Markov . Langevin equation . Fokker-Planck equation . Kramers-Moyal coefficients

1 Introduction

Weather forecasting has a great importance in various fields such as climate assessment, pollution dispersal, detection of drought and similar effects on agriculture, aviation industry, communications, planning in the energy industry, and etc. It is a data gathering process including temperature, humidity, pressure, the speed of the wind, and its direction in different atmospheric conditions. These data are used to interpret weather conditions. The temperature and its fluctuation play an essential role in the forecasting and the classification of the climate. The prediction of the temperature is made by collecting quantitative data of the current atmosphere conditions. Meanwhile, some of these parameters may passively follow other parameters.
Modeling this system may help us to understand which part of the system is more important. It seems that the temperature has an active effect on other parameters; We are trying to show that its stochastic nature can be reproduced even when other elements are ignored. Stochastic processes are widely used to model systems and phenomena whose behaviors appear to be random. Random processes based on their properties can be divided into different categories including Markov processes, Levi processes and Gaussian processes.

Extensive studies have been conducted to get an effective way to reconstruct and describe such events. Obtaining of such a method has the following advantages: Firstly, it is possible to find out the statistical properties and nature of the process. Secondly, if the process is time-dependent, then the future behaviors of the system can be predicted, and if there is a spatial dependence with any type of fractality, then the behavior of the system can be anticipated in both larger and smaller scale.
Nowadays, climate change and its effects on our life are subject matters of many research and studies. Regarding the determination of the climate of a region, studing and analyzing of the most significent parameters such as temperature and precipitation relating to other climatic factors are substantial. The climate of the system has been influenced by its own internal dynamics and external factors. Natural phenomena such as solar and lunar variations, volcanic eruptions, and human-induced changes in atmospheric composition are external factors. In order to realize the various plans of human societies and the importance of the role of climate change in this regard, numerous studies have been conducted in the field of time series changes by different procedures all over the world.
Using the random method of Markov processes, any system with the periodic random behavior can be modeled. The climate is a non-linear multi-variable process that its dynamics is complex; Hence in this research, the reproduction of the temperature fluctuations as a random parameter is discussed. One of the advantages of this approach is that if we have weather information for the real area, we can examine the behavior of the system by manually changing some of its parameters to see which one gets longer among hot and cold period in terms of time factor.

2 Data Preprocessing and Methods

The database used in this paper was provided by the Russian Meteorological Station; It includes daily mean, minimum and maximum temperature and daily rainfall [4]. We selected four stations which have provided time series of daily temperature continually in a long period (Table 1).

Table 1. The geographical coordination of selected stations.

Station	Latitude	Longitude	Date
Svjatoj Nos	N 68° 09′	E 39° 46′	1938/04/04
GMO im.E.K.Fedorova	N 77° 43′	E 104° 18′	1936/01/01
Tiksi	N 71° 35′	E 128° 55′	1936/01/01
Russkij	N 77° 10′	E 96° 26′	1978/11/15

We select 8192 days of the temperature time series of each stations, and then the average of time series is shifted to zero and their variance is normalized to

one. We calculate the Fourier transform of temperature time series; it is seprated into the low and high frequency at the specific magnitude which is the four times of the annual frequency. The yearly periodicity is eliminated from the Fourier transform of the time series separately. Hence, the time series which is filtered in the annual frequency (band-pass filter) can be reconstructed by the inverse Fourier transform. Now, we can regenerate the temperature and its filtered time series in two ways:
The obtained time series by the band-pass filter method which does not have the annual characteristic frequency can be reproduced by a single variable Fokker-Planck equation (FPE). The unfiltered time series that has the oscillating characteristic must be regenerated by the bi-variable FPE. Markov process can be used on a temporal or spatial series, depending on the nature and characteristics of the system under investigation, for instance, epileptic brain dynamics, heart interbeat fluctuations, the fluctuations in the daily price of oil, fluctuations in the currency exchange rates and reconstruction of rough surfaces [7–10].

2.1 Method Using the One Dimensional Fokker-Planck Equation

To analyze and extract information on meteorological data, the Markov process can be applied. A prerequisite for the Markov property is to examine the validity of the Chapman-Kolmogorov equation for the description of probabilities [5].

$$p(\Delta T_k, t_k|\Delta T_i, t_i) = \sum_{T_j} \Delta T_j p(\Delta T_k, t_k|\Delta T_j, t_j) p(\Delta T_j, t_j|\Delta T_i, t_i). \tag{1}$$

Eq. (1) shows the Chapman-Kolmogorov equation at different temperatures. It is hold for any value of ΔT_j, in the interval $t_i < t_j < t_k$. T_i , $p(\Delta T_k, t_k|\Delta T_i, t_i)$ are temperature sequence and the conditional probability distribution. Markov length is determined by the length between the points where the corresponding random variables of these points follow the Markov process. By studying a Markov process, it can be concluded that the distribution function of these data satisfies the Kramers-Moyal (KM) equation. Therefore, we estimate the KM coefficients. According to Paula's theorem, if the fourth-order coefficient is small compared to the second-order coefficient, it is possible to ignore the coefficients above the second order, which simplifies FPE to

$$\frac{d}{dt}p(\Delta T, t) = \left[-\frac{\partial D^{(1)}(\Delta T, t)}{\partial \Delta T} + \frac{\partial^2 D^{(2)}(\Delta T, t)}{\partial \Delta T^2}\right] p(\Delta T, t). \tag{2}$$

$D^{(1)}(\Delta T)$, $D^{(2)}(\Delta T)$, and $p(\Delta T)$ are the drift and diffusion coefficients, and the distribution function of data, respectively. $D^{(1)}(T)$ and $D^{(2)}(T)$ are derived using the first and second-order KM coefficients as follows. $D^{(k)}(T)$ is defined as,

$$\begin{aligned} D^{(k)}(T) &= \frac{1}{k!} \lim_{\tau \to 0} M^{(k)}(T, \tau), \\ M^{(k)}(T, \tau) &= \frac{1}{\tau} \left\langle (T(t+\tau) - T(t))^k \right\rangle \Big|_{T(t)=T}. \end{aligned} \tag{3}$$

As mentioned, the characteristic mode, namely the annual frequency was omited from the power spectrum of the temperature sequence. Because of the predominance of noise, the effect of temperature daily fluctuations is only observed during calculating the Markov time scale, and the effect of season changes cannot be seen. The derived Markov time scale is the smallest time scale and it is displayed as τ_{m}. The KM coefficients can be estimated by this time scale in both methods: the single and bi-variable FPEs. The Langevin equation is derived using the Itô interpretation to regenerate the time series as well as the governing equation:

$$T_{i+1} = T_i + D^{(1)}(T_i)\delta t + \sqrt{\delta t D^{(2)}(T_i)}\Gamma(t). \tag{4}$$

$\Gamma(t)$ is a random force with the zero mean, the unit variance, and the Gaussian statistics. T_i is time series of reconstructed temperature. The distribution function of the Langevin equation also is applied to FPE. Therefore, it can be concluded that the derived Langevin equation describes the fluctuation of the time series. It should be noted that the oscillatory motion of the oscillator is not always ideally harmonious and it does not last constantly; but after a while, the oscillatory track get decreased and the oscillator stops. Due to the damping in oscillating systems, notably in the harmonic oscillator which reduces the oscillation amplitude, a driven force is used to continue the oscillating motion. Concequently to consider the effect of the harmonic fluctuations of system as a periodic force, we apply a driven force to the Langevin equation in the single-variable FPE. We adjust the oscillation amplitude that has fluctuations corresponding to the behavior of the data under investigatin. Because, the period of this force has caused the same frequency of annual fluctuations, this magnitude is just modified. As a result, we can rewrite Eq. (4) as following:

$$T_{i+1} = T_i + D^{(1)}(T_i)\delta t + \sqrt{\delta t D^{(2)}(T_i)}\Gamma(t) + A\cos(\omega_0 t). \tag{5}$$

2.2 Method Using the Two Dimensional Fokker-Planck Equation

We define the following state vector $\boldsymbol{T}(t) = \{T(t), \Delta T(t)\}$ that follows the theory of random process:

$$q_j = T_j, (j = 1 : 2^n), \tag{6}$$
$$Q_j = (T_{j+\tau} - T_j). \tag{7}$$

The length of the time series is almost 2^{13} for all staions. An example for further explanation is given below. For temperature time series as

$$T = \{-0.06, -0.1, 0.61, 0.67, 0.1, 0.35, 0.35, 0.27, \cdots\},$$

one can use Eqs. (6 and 7) and finds q and its increment as

$$q = \{-0.06, -0.1, 0.61, 0.67, 0.1, 0.35, 0.35, 0.27, \cdots\},$$
$$Q = \{-0.04, 0.7, 0.05, -0.56, 0.25, 0, -0.08, \cdots\}.$$

where $\tau = 1$ in this example. Due to the periodicity of the temperature time series, the effect of the season variation is observed in estimating of the Markov time scale. The annual fluctuations and season variations are also seen with larger temperature variations. In fact this is called the effect of harmonic oscillations. This time scale depends on the length of the oscillating period of time series and it is displayed with $\tau = \tau_{\mathrm{M}}$. The effect of the season variations is applied to the two coupled stochastic differential equations. The parameter of temperature variation, namely Q is measured by using this time scale as a time step(i.e., $\tau = \tau_{\mathrm{M}}$). Assuming that the process is Markov, we obtain the KM coefficients. Drift and Diffusion coefficients are $D^{(1)}(q;Q)$ and $D^{(2)}(q;Q)$ that are derived using the following equation:

$$\begin{aligned}
D_q^{(1)}(q,Q) &= \lim_{\tau\to 0}\frac{1}{\tau}\big\langle (q(t+\tau)-q(t))\big\rangle\big|_{q,Q},\\
D_Q^{(1)}(q,Q) &= \lim_{\tau\to 0}\frac{1}{\tau}\big\langle (Q(t+\tau)-Q(t))\big\rangle\big|_{q,Q},\\
D_{qq}^{(2)}(q,Q) &= \lim_{\tau\to 0}\frac{1}{2\tau}\big\langle (q(t+\tau)-q(t))^2\big\rangle\big|_{q,Q},\\
D_{qQ}^{(2)}(q,Q) &= \lim_{\tau\to 0}\frac{1}{2\tau}\big\langle (q(t+\tau)-q(t))(Q(t+\tau)-Q(t))\big\rangle\big|_{q,Q},\\
D_{QQ}^{(2)}(q,Q) &= \lim_{\tau\to 0}\frac{1}{2\tau}\big\langle (Q(t+\tau)-Q(t))^2\big\rangle\big|_{q,Q}.
\end{aligned} \tag{8}$$

$D_q^{(1)}(q,Q)$ and $D_Q^{(1)}(q,Q)$ are the first-order KM coefficients of temperature and its increments. $D_{qq}^{(2)}(q,Q)$, $D_{qQ}^{(2)}(q,Q)$ and $D_{QQ}^{(2)}(q,Q)$ are the second-order coefficients of temperature-temperature, temperature-increment, and increment-increment. In general, we can display these coefficients as polynomials:

$$D^{(1)}(q,Q) = \sum_{i+j=1,3} a_{ij}(q^i,Q^j), \tag{9}$$

$$D^{(2)}(q,Q) = \sum_{i+j=0,2} b_{ij}(q^i,Q^j), \tag{10}$$

where a_{ij} and b_{ij} coefficients should be estimated using Eq. (8). $D^{(1)}$ and $D^{(2)}$ are odd and even function, respectively. $D^{(1)}$ must be theoretically an odd function to cause damping. There is not a physical reason for this asymmetric, but $D^{(2)}$ should not be asymmetric. The q and Q time series are reconstructed by the Langevin equation which preserve the same statistical properties to their real time series:

$$\begin{aligned}
q_{t+1} &= q_t + D_q^{(1)}(q,Q)\delta t + \sqrt{\delta t D_{qQ}^{(2)}(q,Q)}\Gamma_2(t) + \sqrt{\delta t D_{qq}^{(2)}(q,Q)}\Gamma_1(t),\\
Q_{t+1} &= Q_t + D_Q^{(1)}(q,Q)\delta t + \sqrt{\delta t D_{Qq}^{(2)}(q,Q)}\Gamma_1(t) + \sqrt{\delta t D_{QQ}^{(2)}(q,Q)}\Gamma_2(t).
\end{aligned} \tag{11}$$

Eq. (8) shows that $D_{qQ}^{(2)}$ and $D_{Qq}^{(2)}$ are symmetric. $\Gamma_1(t)$ and $\Gamma_2(t)$ are a ran-

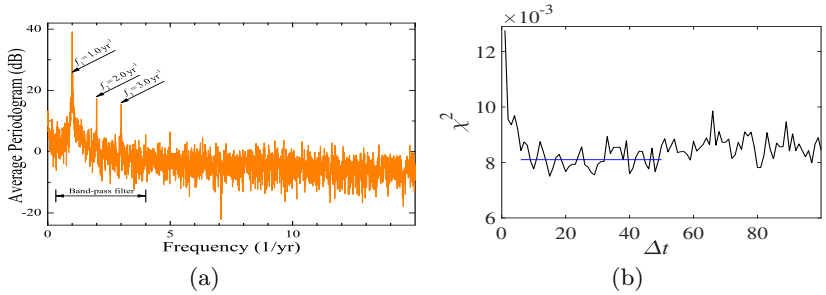
dom force, zero mean with Gaussian statistics. The Langevin equation is the first-order stochastic differential equation. Thus, we transform the differential equation of two-order into a system of two first-order differential equations that are coupled together. We try to derive the bi-variable Langevin equation. The choice of a numerical scheme is substantial for solving differential equations. Since definite ways are usually not suitable for such equations and the full implicit methods also cause random instability. In Euler's scheme numerical error is the order of h. We are looking for a method that has less error (the order of h^2). Therefore, we concentrate on the second-order Runge-Kutta random methods which are explicit in terms of the coefficients of the drift and diffusion [6].

3 Results and Discussion

The 1D Fokker-Planck method, the Fourier transform of each station is computed and the annual frequency is eliminated; The inverse Fourier transform is obtained and then this data is used in computations. The Markov time scale is calculated for each station separately, as shown in Table 2. Fig. 1(a) shows the

Table 2. τ_m and τ_c represent the Markov time scale and characteristic time-scale, respectively.

Station	τ_m (day)	τ_c (day)
Svjatoj Nos	10	24
GMO im.E.K.Fedorova	10	28
Tiksi	10	34
Russkij	10	32



(a) (b)

Fig. 1. (a) The spectral density of the temperature time series. (b) $dt = 10\,\mathrm{d}$ is the obtioned Markov time scale by the Chapman-Kolmogorov equation.

spectral density of temperature fluctuations for the station of GMO im. E. K. Fedorova. According to Fig. 1(b), one horizontal line is drown during 10 to 50 days time interval. The vertical axis error is reached to 0.081 by averaging the fluctuation values of the time ranging. If this line is intersects the diagram at only one point, this meeting point is the Markov time scale τ_m. If it passes the diagram, then going up again, the middle of the interval is τ_m. It turns out that the Markov time scale is about 10 d.

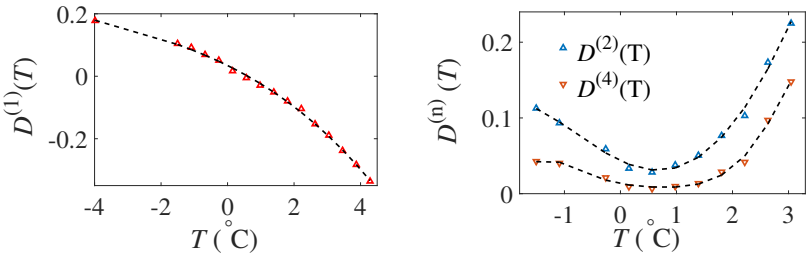


Fig. 2. (a) Drift Kramers-Moyal coefficient for the station of GMO im. E. K. Fedorova. The upward triangles represent the measured coefficient $D^{(1)}$ by Eq. (2) and the dashed curve is fitted equation corresponding to this coefficient. (b) Diffusion and fourth-order Kramers-Moyal coefficients for the station of GMO im. E. K. Fedorova. The upward and downward triangles are represented the calculated coefficients $D^{(2)}$ and $D^{(4)}$ and dashed curves represent fitted equations corresponding to these coefficients.

The KM coefficients are estimated for the first to fourth-orders. The ratio of the fourth order coefficient to the second order coefficient $D^{(4)}/D^{(2)}$ is 0.5. Regression coefficients are above 99 percent for the station of GMO im. E. K. Fedorova. Fig. 2(a) and (b) show the KM coefficients. We obtained the following expression for the $D^{(1)}(T)$ and $D^{(2)}(T)$:

$$D^{(1)}(T) = -0.0004T^3 - 0.006T^2 - 0.05T + 0.03,$$
$$D^{(2)}(T) = 2 \times 10^{-5}T^4 + 0.005T^3 + 0.02T^2 - 0.03T + 0.05.$$

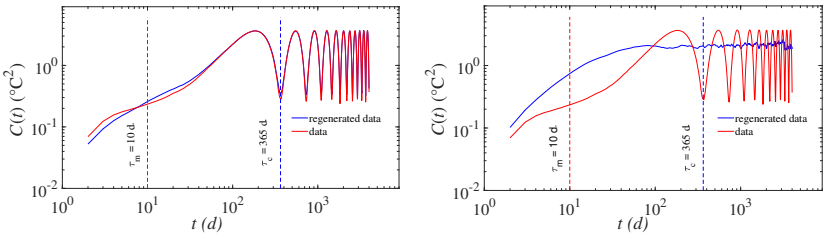


Fig. 3. Log-log plot of the second moment of temperature difference *vs.* t along the Markov time scale $dt = 10$ d, for the real and reconstruced time series by (a) using the driven periodic force and (b) without it.

The regenerated time series by Eq. (4) does not conform to the original one. The large difference at the end of the second moment of the structure functions between unfiltered and reproduced data in Fig. 3(a) (i.e. periodic oscillation *vs.* flat curve) represents the annual characteristic frequency. Due to the annual frequency and the lack of satisfaction of the Paula theorem, the actual auto-

correlation curve has not been reconstructed.
This inconsistency is modified by the addition of the oscillation to the Langevin equation in Fig. 3(b). The value of this periodic force is estimated using the maximum ratio of the real signal to the reproduced signal. According to Fig. 3, driven force is added to the Langevin equation which is the order of the annual frequency [Eq. (5)]. The annual frequency and the amplitude of the force are $\omega_0 = 2\pi\,\mathrm{rad/yr}$ and $A = 0.92$, respectively.

In the bi-variable FPE method, lower frequencies are eliminated from the Fourier transform and the inverse Fourier transform is calculated. The preprocessed data is used to measure the first-order KM coefficients. The first-order KM coefficients for all stations are derived for temperature at 99 percent and temperature variations with a precision of 98 percent by using the presented Markov ltime scale in Table 1. Fig. 4 shows the first-order KM coefficients of temperature and its increments which are estimated for the station of Svjatoj Nos on the following approximation:

$$\begin{aligned} D_q^{(1)}(q,Q) &= -0.05 + 0.07q - 0.20Q + 0.03qQ - 0.02q^2 - 0.12q^3 + 0.15q^2Q \\ &\quad +5\times10^{-4}Q^2 - 0.1qQ^2 + 0.02Q^3, \\ D_Q^{(1)}(q,Q) &= 0.10 + 0.34q - 0.27Q + 0.09qQ - 0.09q^2 - 0.10q^3 + 0.08q^2Q \\ &\quad +4.0\times10^{-4}qQ^2 - 0.04Q^2 - 0.03Q^3. \end{aligned}$$

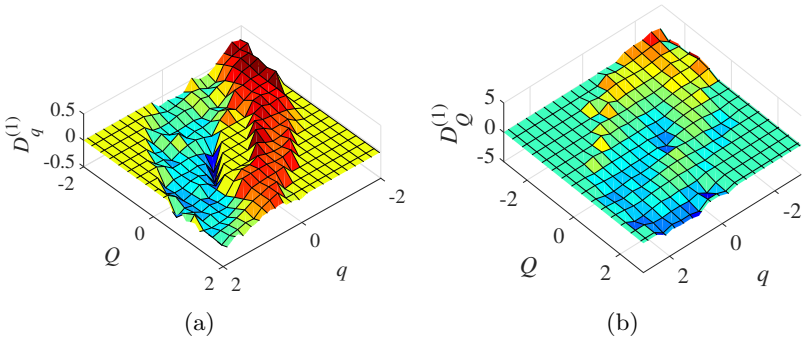


Fig. 4. 2D first-order KM coefficients for the Svjatoj Nos station. (a) Drift coefficients in terms of $D_q^{(1)}$ and (b) $D_Q^{(1)}$.

The second-order coefficients are calculated by the unfiltered data. $D_{qq}^2(q,Q)$, $D_{QQ}^2(q,Q)$, and $D_{qQ}^2(q,Q)$ are obtained for the station of Svjatoj Nos (see Fig. 5). For all stations, regression coefficients for all fits are above 96 percent. The obtained coefficients are embedded in the coupled Langevin equations; the reconstructed signal has not the same statistical characteristics to its original pattern.

$$D^{(2)}_{qq}(q,Q) = 0.01 - 0.003q + 0.008q^2,$$
$$D^{(2)}_{QQ}(q,Q) = 0.03 + 0.002Q + 0.01Q^2,$$
$$D^{(2)}_{qQ}(q,Q) = 0.01 - 0.002Q + 0.006Q^2.$$

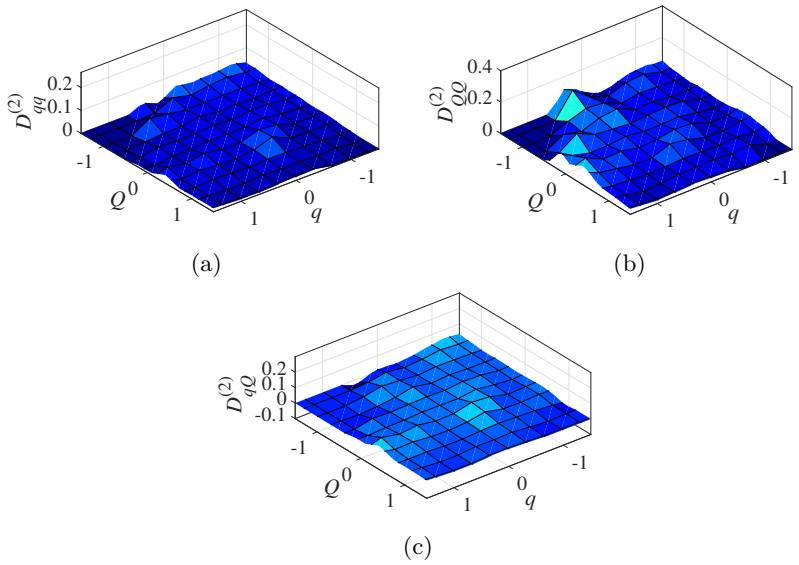


Fig. 5. 2D second-order Kramers-Moyal coefficients for the station of Svjatoj Nos. (a) Diffusion coefficients in terms of $D^{(2)}_{qq}$, (b) $D^{(2)}_{QQ}$, and (c) $D^{(2)}_{qQ}$. $D^{(2)}_{qq}$ is flat along Q axis and the two other coefficients are flat along q axis.

According to Fig. 6(b), driven force is added to the first-order coefficient of temperature which is the order of the annual frequency. The annual frequency and the frequency magnitude are $\omega_0 = 2\pi\,\mathrm{rad/yr}$ and $A = 0.17$, respectively.

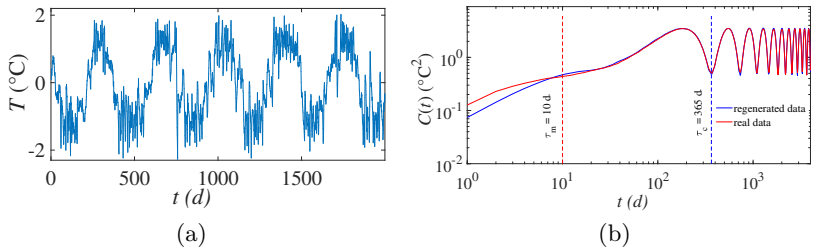
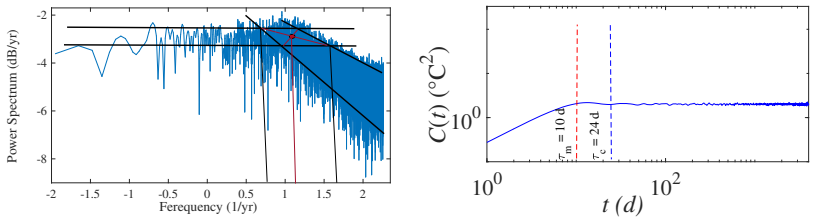


Fig. 6. (a) The regenerated time series by using driven force. (b) Log-log plot of the second moment of temperature difference *vs.* t, for the real and regenerated signal.



(a) (b)

Fig. 7. (a) The power spectrum of the regenerated signal in the absence of deriven force that shows the mean value at $1.1 \pm 0.2\,1/\mathrm{yr}$. (b) Log-log plot of the second moment of temperature difference *vs.* t, for the reconstructed signal without the driven force that shows the characteristic frequency in time scale of $24 \pm 3\,\mathrm{d}$.

The second moment of structure function C is measured by using $C(t) = \langle |T(t_1) - T(t_2)|^2 \rangle$. It shows the match between the real and regenerated signals. The characteristic frequency for the analyzed stations by the 2D method is presented in the Table 2. It is derived $24 \pm 3\,\mathrm{d}$ for this station [Fig. 7(b)]. Because there is a cross point in correlation curve, we can not estimate accurately this frequency by the correlation diagram. So, we plot the power spetrum of the regenerated signal in Fig. 7(a) to estimate the accurate characteristic frequency. This diagram displays the characteristic frequency at $1.1 \pm 0.2\,1/\mathrm{yr}$ which is equal to $27 \pm 2\,\mathrm{d}$ [Fig. 7(a)].
The second moment of structure function C_2 did not have good result for real and reproduced time series of Tiksi station by using single variable FPE [Fig. 8(a)]. The bi-variable FPE method for this station by using driven force had a good match to the real temperature time series [Fig. 8(b)]. The annual frequency and the amplitude of the driven force are $\omega_0 = 2\pi\,\mathrm{rad/yr}$ and $A = 0.13$, respectively. The first and second-order KM coefficients are esitimated as following:

$$
\begin{aligned}
D_q^{(1)}(q,Q) &= -0.19 + 0.35q - 0.2Q - 0.09qQ + 0.14q^2 - 0.29q^3 + 0.30q^2Q \\
&\quad +0.06Q^2 - 0.08qQ^2 - 0.01Q^3, \\
D_Q^{(1)}(q,Q) &= -0.08 + 0.37q - 0.008Q + 0.05qQ + 0.07q^2 - 0.13q^3 - 0.04q^2Q \\
&\quad -0.03Q^2 + 0.19qQ^2 - 0.14Q^3, \\
D_{qq}^{(2)}(q,Q) &= 0.01 - 0.004q - 0.002q^2 + 0.002qQ + 0.001Q^2, \\
D_{QQ}^{(2)}(q,Q) &= 0.01 - 0.003Q + 0.003Q^2, \\
D_{qQ}^{(2)}(q,Q) &= 0.02 - 0.0003Q + 0.004Q^2.
\end{aligned}
$$

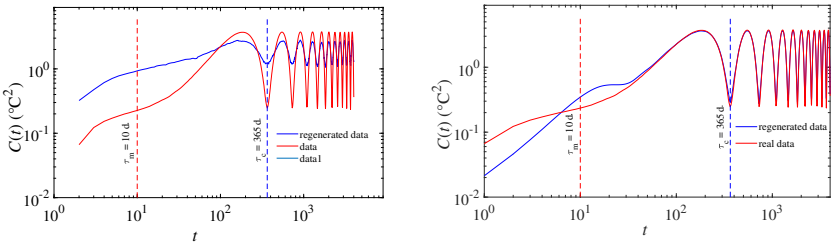


Fig. 8. Log-log plot of the second moment of temperature difference *vs.* t, for the real and regenerated signal of Tiksi station by using (a) the one dimensional FPE and (b) the two dimensional FPE .

4 Conclusions

The best results are obtained for stations of GMO im.E.K.Fedorova and Svjatoj Nos by means of the single and bi-variable Markov methods, respectively. Tiksi station has an appropriate result for the bi-variable FPE while it does not have a good result for the single variable FPE. Regenerated time series are very similar in statistical sense to their original one. The auto-correlation curves of reconstructed samples are inconsistent to their real one, therefor driven forces are added to the Langevin equation and first-order KM coefficient in both one and tow-dimensional Fokker-Planck equations, repectively. Charasteristic modes in the absence of driven forces are estimated for all patterns, but we can not accurately attribute these frequencies to effects of the solar or lunar variations.

Acknowledgments

Database for this work was collected by All Russian Research Institute of Hydrometeorological Information World Data Centre Obninsk, Russia.

References

1. Devi, C. J., Reddy, B. S. P., Kumar, K. V., Reddy, B. M., and Nayak, N. R.: ANN approach for weather prediction using back propagation, International Journal of Engineering Trends and Technology **3**(1), 19–23 (2012)
2. Shereef, I.K, Santhosh Baboo, Dr. S.: A New Weather Forecasting Technique using Back Propagation Neural Network with Modified Levenberg-Marquardt Algorithm for Learning, IJCSI International Journal of Computer Science Issues **8**(6) (2011)
3. Treut, H. Le, Somerville, R., Cubasch, U., Ding,Y., Mauritzen, C., Mokssit, A., Peterson, T., and Prather, M.: Historical Overview of Climate Change. In: 4th Assessment Report of the Intergovernmental Panel on Climate Change, pp. 95–127.

The Physical Science Basis, Cambridge University Press, Cambridge, New York (2007)
4. CDIAC Homepage, http://cdiac.ess-dive.lbl.gov/ndps/russia_daily518. Last accessed March 2018
5. Risken, H., Haken, H.: The Fokker-Planck Equation: Methods of Solution and Applications. 2nd edn. Springer, Washington (1989)
6. Rößler, A.: Second Order Runge–Kutta Methods for Itô Stochastic Differential Equations. SIAM Journal on Numerical Analysis **47**(3): 1713—1738 (2009).
7. Friedrich, R., Peinke, J., Sahimi, M., Rahimi Tabar, M.R.: Approaching complexity by stochastic methods: From biological systems to turbulence. Physics Reports **506**(5), 87–162 (2011)
8. Jafari, GR., Fazeli, SM., Ghasemi, F., Allaei, SM., Rahimi Tabar, M.R.: Stochastic analysis and regeneration of rough surfaces. Physical Review Letters **91**(22), 226101–226101 (2003)
9. GhasemiMuhammad, F., Sahimi, M., Peinke, J., Rahimi Tabar, M.R.: Analysis of Non-stationary Data for Heart-rate Fluctuations in Terms of Drift and Diffusion Coefficients. New Journal of Physics **11**(10), 117-–128 (2006)
10. Bahraminasab, A.R, Ghasemi, F., Stefanovska, A., McClintock, P., Friedrich, R.: Physics of brain dynamics: Fokker–Planck analysis reveals changes in EEG δ–θ interactions in anæsthesia. New Journal of Physics **11**(10), 103051–103051 (2009)

Selective Attention in Exchange Rate Forecasting*

Svatopluk Kapounek [I], Zuzana Kučerová [II]

April 2019

Abstract

The paper investigates exchange rate forecasting performance of mainstream macroeconomic fundamentals, uncertainty, attention and news. Although currency markets react to many different information, we hypothesize that smaller sizes models offer better fitting and forecasting results because economic agents accept only a limited amount of information. Our results confirm selective attention hypothesis of currency market participants and explain behavioural biases in investment decision-making process.

We employ Dynamic Model Averaging approach to reduce model-selection uncertainty and to identify time-varying probability to include regressors into our models. We analyse 100.5 thousand news articles about the 6 most traded Forex currency pairs in the period from 1979 to 2016 and confirm growing impact of the news about foreign trade and monetary policy issues on the Euro/U.S. Dollar exchange rate after the financial crisis. The other analysed currencies react mostly to output differentials and stock market returns induced by portfolio rebalancing.

Keywords: Exchange Rate, Selective Attention, News, Dynamic Model Averaging
JEL Classification: F33, G41, C11

* This research was funded by the Czech Science Foundation via grant No. 16-26353S "Sentiment and its Impact on Stock Markets". We benefited from comments and suggestions made by Makram El-Shagi, Jesus Crespo Cuaresma, Jarko Fidrmuc, Roman Horváth, Tomáš Holub and other participants of the 5th HenU / INFER Workshop on Applied Macroeconomics in Kaifeng, March 2019, 10th Biennial Conference of the Czech Economic Society in Prague, December 2018, WU Vienna Workshop Macroeconomic policy in the Eurozone, September 2018, Western Economic Association International Conference in Vancouver, June 2018, Baltic Economic Conference in Vilnius, June 2018, and International Atlantic Economic Society Conference in Montreal, October 2017.
[I] Mendel University in Brno, Faculty of Business and Economics, Czech Republic.
Full corresponding address: Mendel University in Brno, Faculty of Business and Economics, Zemědělská 1, 613 00 Brno, Czech Republic, Phone: +420 545 132 444, Fax: +420 545 132 450, e-mail: kapounek@mendelu.cz
[II] Mendel University in Brno, Faculty of Business and Economics, Czech Republic.

1. Introduction

Traditional forecasting models using macroeconomic variables (such as output, inflation, interest rates, trade, etc.) still seem to fail in some cases, generally because of overfitting the model or the selection of bad or too many fundamentals (see Sarno and Valente, 2009 or Rapach et al., 2010). Moreover, good exchange rate forecasts are strongly related to good macro fundamentals forecasts and this relationship is time-varying and as such it could be the fact why forecasting models may fail (Dick et al., 2015). Moreover, the *theory of rational inattention* raised a critique about the predictive value of standard financial and macroeconomic models (Simon, 1971; Sims, 2003, 2006, 2010; Maćkowiak and Wiederholt, 2015), it is important to study other than macroeconomic factors that might explain the exchange rate dynamics.

We follow this stream of literature and provide contribution in several ways. First, we assume that large exchange rate swings could be better explained by market sentiment and uncertainty resulting from news announcements. We construct several indices from 100,5 thousand published news articles about economic activity, monetary issues, price development, and foreign trade related to the countries representing selected currency pairs. The indices reflect selective attention of newspapers to relevant information about the macroeconomic fundamentals which theoretically influence the analysed exchange rates. Additionally, we use Google queries as a measure of behavioural attention and the CBOE volatility indexes as a measure of currency demand and attractiveness.

Second, we deal with a problem of information overload. We hypothesise that economic agents are overwhelmed by information using various devices to gain information from various sources but, at the same time, they are equipped with only a limited attention and ability to process these data. There is a discussion about *selective attention* psychology but the empirical research incorporating limited amount of information accepted by market participants is very limited.[1] Our comparison of several models (including traditional macroeconomic fundamentals and behavioural factors) and estimating techniques (time-varying parameter VAR, Dynamic Model Averaging and Dynamic Model Selection) show that behavioural factors improve forecasting performance but only after we select the information to which market participants pay attention.

[1] From the psychological point of view, there is a discussion about selective attention when economic agents decide to accept only a limited amount of information even though this decision does not lead to optimum instead of behaving inattentively. For a detailed review of theoretical and empirical papers concerning the economics of attention, see Festré and Garrouste (2015).

Third, we explore different exchange rate forecasting performance of macroeconomic fundamentals and behavioural factors for the 6 most traded Forex currency pairs (the U.S. dollar to British pound, Euro, Australian dollar, Canadian dollar, Japanese yen and New Zealand dollar). Our results show decreasing impact of interest rate differential on the most of currency pairs while portfolio rebalancing after the financial crisis (represented by stock returns) influenced only U.S. dollar, Euro and Australian dollar. The Euro/U.S. dollar exchange rate reacts sensitively to the article news about foreign trade and monetary policy issues.

The structure of the paper is as follows. Section 2 reviews the literature concerning macroeconomic determinants of exchange rate movements, behavioural factors, and the attention. Section 3 introduces data and the methods used in the paper. Section 4 compares differences and estimations errors between our basic and behavioural models, and time-varying probability to include regressors into the models. Section 5 contains robustness analysis employing different estimation techniques (time-varying parameter VAR, Dynamic Model Averaging, and Dynamic Model Selection) and section 6 concludes.

2. Literature Review

Exchange rates and its movements could be explained by established economic theories and by empirical exchange rate models that try to explain and forecast the movements of exchange rates. However, these traditional empirical models did not serve as a good way for proper testing and forecasting of exchange rate movements because, in many cases, large exchange rate swings were also explained by news, sentiment, uncertainty, inattention (or by other behavioural determinants) than by macroeconomic fundamentals as proved by empirical research. Using new econometric techniques, new observable variables and richer datasets during the last two decades and because of increasing level of financial integration and globalisation, many authors tried to verify and reassess the relationship between nominal exchange rates and macroeconomic fundamentals and to improve forecasting models. Sims (1998, p. 344) argue that "*…actual behavior of macroeconomic aggregates shows a combination of real and nominal sluggishness…*" and as such, "*…macroeconomists should rethink their commitment to modeling behavior as continuous dynamic optimization, with delays and inertia represented as emerging from adjustment costs.*"

There are three subsections in this part of the paper. First summarises mainstream macroeconomic fundamentals which influence market parity conditions in the long-run. Subsection 2 explains impact of the news announcements and other behavioural factors of the exchange

rate movements. Subsection 3 contributes with the *selective attention hypothesis* at the currency markets.

2.1 Macroeconomic exchange rate determinants and forecasting

There are several mainstream macroeconomic models explaining the determinants of the exchange rate movements: (1) the purchasing power parity theory; (2) the uncovered interest parity theory; (3) the monetary model of exchange rate; (4) the real interest differential model; and (5) the portfolio balance model. Dornbusch (1976) contributes with the hypothesis of overshooting exchange rate movements considering the role of asset markets, capital mobility and expectations. He emphasizes impact of monetary policy responses (especially interest rate changes) to inflation rate and real output targets. Frenkel (1976) or Bilson (1978) follows with the early discussion concerning *the monetary model* of exchange rate determination where the nominal exchange rate is determined mainly by the relative (difference between the domestic and foreign) real income and relative money supply assuming that the demand for money is a stable function and that both the purchasing-power parity theory and the uncovered-interest parity theory hold. There is wide range of literature confirming the existence of the monetary model, i.e. the existence of the significant long-term relationship between nominal exchange rate and monetary fundamentals (see e.g. MacDonald and Taylor, 1993; Mark, 1995; Mark and Sul, 2001; Rapach and Wohar, 2002; Cerra and Saxena, 2010; Loría et al., 2010; Dabrowski et al., 2014; Chang and Su, 2014; Burns and Moosa, 2015; Chen and Chou, 2015).

Frankel (1979) follows the approach of Dornbusch (1976) too and introduces a modified model of exchange rate determination (*the real interest differential model*, RID) where he employs additional macroeconomic variables, i.e. short- and long-term interest rates when short-term rates represent the role of monetary policy (or liquidity) and long-term interest rates include inflation expectations. He aims to capture the difference between the short and long run which is caused by sticky prices in goods markets. Frankel (1979) confirms the validity of the Dornbusch (1976) model and refuses the simple monetary model. Issac and Mell (2001) follow Driskill and Sheffrin (1981) who redefine the RID model into the form of RIDRE model under rational expectations, however, with better results.

In another paper, Frankel (1984) also tests the validity of the monetary model and *the portfolio balance model*. According to Hooper and Morton (1982), the exchange rate is determined not only by money supply, real output and short-term interest rates but also by expected inflation and cumulated trade balances. The trade balance is an important variable in this model as current account imbalances cause the exchange rate changes when asset holders rebalance their

portfolios in reaction to the external imbalances. In the Hooper and Morton (1982) model specification, the exchange rate movements are influenced particularly by the expectations of asset holders (the static expectations approach). Taylor (1995) summarises mainstream macroeconomic fundamentals and describes (1) monetary model with flexible and (2) sticky prices; (3) equilibrium exchange rate model; (4) liquidity exchange rate models and portfolio balanced model and points to the role of money supply, nominal interest rates and real income. Yuan (2011) examines the impact of basic macroeconomic determinants (money supply, real GDP, CPI, short-term and long-term interest rates, and current account balance) on the nominal exchange rate.

Meese and Rogoff (1983) started a new type of discussion concerning the reliability and forecasting capacity of exchange rate models. They compare the out-of-sample forecasting accuracy of structural and time series exchange rate models from 1973 to 1981 and conclude that a random walk model performed no worse than any estimated time series model (the "*Meese-Rogoff Puzzle*"). As Frankel and Rose (1995) claim this finding negatively influenced the modelling of exchange rates after this research had been published.

Following this seminal work, many authors try to verify the existence of the "*exchange rate disconnecting puzzle*", i.e. that exchange rate models are disconnected from macroeconomic fundamentals, with both positive and negative results, e.g. Engel and Hamilton (1990), Meese (1990), Leitch and Tanner (1991), Christoffersen and Diebold (1998), Tashman (2000), Faust et al. (2003), Cheung et al. (2005), Engel and West (2004, 2005), Bacchetta and Wincoop (2006), Engel et al. (2007), Gourinchas and Rey (2007), Balke et al. (2013), Rossi (2013) or Moosa and Burns (2014).

However, the discussion about the role of macroeconomic fundamentals in the exchange rate forecasting process is still active as standard forecasting models using macroeconomic variables (such as output, inflation, interest rates, trade, etc.) fail in some cases. Bacchetta and van Wincoop (2004, 2013) verify the existence of a stable relationship between macroeconomic fundamentals and exchange rate and find that the existence of a set of unobservable fundamentals not captured in empirical models can generate a considerable confusion of economic agents particularly in the short to medium term as they do not include these unobservable fundamentals in their decisions and attribute the changes of exchange rates only to macroeconomic fundamentals instead (authors use the term "*scapegoat*" in this context). In other words, economic agents blame macroeconomic fundamentals for large and unexpected exchange rate movements. Moreover, these expectations may significantly vary over time and create a high level of uncertainty in the exchange rate fluctuations. These findings are confirmed by Fratzscher et

al. (2015) or Beckmann and Czudaj (2017) who support this scapegoat theory. Therefore, the role information and uncertainty in the economy stays in the centre of contemporary exchange rate research.

2.2 News Puzzle

There is also wide range of literature on the predictive power of exchange rate *expectations* despite the limited availability of aggregated expectation data. Very often, authors use high-frequency survey data and consensus forecasts (i.e. expected rather than realised data) and measure the reactions of economic agents to *macro- and micro news and announcements* concerning economic policy or economic situation in the context of "*the news puzzle*". The so-called announcement phenomenon could be responsible for model estimations different from postulations of macroeconomic theory (Engel and Frankel, 1984). While first results were a bit disappointing in the sense that they can predict future exchange rate movements particularly in the short run, later studies offer a possible way how to successfully forecast the movement of exchange rates and find that macro news is responsible for currency price variations (see Hardouvelis, 1988; Ito, 1990; Chinn and Frankel, 1994; Engel et al., 2007; Faust et al., 2007; Clarida and Waldman, 2008; Rosa, 2011; Cavusoglu and Neveu, 2015; Dick et al., 2015; Kočenda and Moravcová, 2016; Omrane and Savaşer, 2016; Beckmann and Czudaj, 2017 and others).

Similar interesting stream of empirical literature has also risen in recent years; authors formulate *the information-based interpretation of exchange rate movements*. Using a simple two-country open economy model, Klein et al. (1991) estimate the response of the USD exchange rates to trade news (more precisely, movements in the external balance) and conclude that this reaction exists but only after the Plaza agreement in 1985. Evans and Lyons (2002) test the role of trade innovations and the information effect in the portfolio-shift model (the portfolio-balance effect) using the signed order flow data. While Evans and Lyons (2002) use a simple econometric method with ambiguous results, Payne (2003) employs VAR model in order to detect the long-run effects of trade information on exchange rates. Breedon and Vitale (2010) develop an alternative way to differentiate between the information effect and portfolio-balance effect including the role of the inventories of forex investors. These studies are followed by Lyons and Moore (2009), Evans (2010), Rime et al. (2010), Cerrato et al. (2011), Zhang (2014) or Chen and Zhang (2015).

2.3 Uncertainty and Selective Attention

We deal with both uncertainty and attention issues in our models which results in selective attention of market participants. It is generally agreed that people make judgements under uncertainty and use several heuristic principles which produce systematic errors (Tversky and Kahneman, 1974). The phenomenon of *uncertainty* became a popular research topic again in the period after the financial crisis of 2007 and 2008 and the subsequent economic recession. According to Bloom (2009, 2014), uncertainty can have an impact on output, employment or foreign exchange rate expectations and its volatility, particularly in the periods of recessions or worse economic performance.

Moreover, volatility both at macroeconomic and microeconomic level is more common in periods of lower economic growth or recessions. Bachman and Bayer (2011) focus only on German firms and conclude that the causality between uncertainty and economy could be reverse, i.e. that the uncertainty could be a result of economic downturns rather than its cause.

There are several approaches how to measure the uncertainty, however, there is no single and objective measure and researchers use only proxies such as the *volatility* or *dispersion* of macroeconomic, microeconomic or financial variables, e.g. the VIX index (the CBOE Volatility Index) measuring the market's expectation of future volatility in U.S. equity markets, or *the appearance of specific words* in newspapers and other publications (for a survey of studies concerning this topic, see Bloom, 2014; Égert and Kočenda, 2014; Jurado et al., 2015 or Caporale et al., 2017). Beckmann and Czudaj (2017) study the impact of economic policy uncertainty on the exchange rate expectations in the US and state that announcements and uncertainty concerning policy decisions are important determinants of exchange rate expectations. Therefore, uncertainty together with economic policy may serve as a proxy for unobservable components not included in former theoretical models of expectations (see the scapegoat theory defined by Bacchetta and van Wincoop, 2013). Another interesting proxy for uncertainty could be the *frequency* of newspaper articles containing specific words such as "uncertain/uncertainty", "economy/economics" etd. (Baker et al., 2016).

However, Jurado et al. (2015) emphasise that these proxies may not be well connected to economic uncertainty and provide a new measure of uncertainty derived from macroeconomic activity, i.e. they do not study the volatility or dispersion of selected individual variables as such, but they try to find whether the predictability of the economy (common variation in uncertainty across all time series) is less or more uncertain. Authors identify three main episodes

of macroeconomic uncertainty in the post-war period (1973-74, 1981-82 and 2007-09) and conclude that this general uncertainty is lower than the individual uncertainty (of individual variables).

There is also a possibility to use Google Trends data as a way how to express the uncertainty in the era of limited information and data availability. There are several studies analysing Google queries in the process of forecasting macroeconomic variables. Suhoy (2009) uses the Google search data to test its forecast ability in case of Israel and finds that these indices can help identify inferences about the economic growth before official data are released. Koop and Onorante (2013) nowcast several macroeconomic variables using US data to test whether additional information contained in the Google data can increase the forecasting performance of conventional models using conventional set of predictors. The authors confirm that the inclusion of Google data improves the performance of these forecasts of general macroeconomic aggregates and that using Google data in the form of model probabilities instead of regressors can help identify structural changes in the trend behaviour of macroeconomic variables and deal with forecasts after crisis.

Smith (2012) tests whether Google data can predict the volatility of exchange rates and argue that these data has some predictive power beyond standard models. Kristoufek (2015) who studies the dynamic relationship between the price of BitCoin and search queries on Google Trends and Wikipedia finds a strong bidirectional correlation between these variables, i.e. the search queries have an impact on the prices if BitCoin and the Prices of BitCoin have an impact on the search queries and that this fact can produce frequent bubbles connected with the movement of the price of BitCoin. Bulut (2015) uses internet search data from Google Trends to capture the information set of decision makers and concludes that the utilisation of the Google Search Data concerning current macroeconomic variables and nowcasting of these variables should be an alternative for proper testing of exchange rate determination models because of the existence of the lag in the availability of the official data to the market participants. Therefore, he suggests using the Google Trends Data to nowcast the future exchange rate movement. Goddard et al. (2015) study the relationship between investor attention and the dynamics of currency prices using a Google search volume index for main currency pairs and find that changes in investor attention are associated with changes in the holdings of the largest traders in foreign exchange markets when the causality runs mainly form investor attention to market volatility. Seabold and Coppola (2015) focus on foreign exchange markets and construct a new index for consumer search behaviour and find that the use of the Google Trends data improves the quality of forecasting in about 20 percent.

The theory of *behavioural attention* is closely connected with the uncertainty as many economic agents make their decisions under a certain degree of uncertainty and risk. Formulating the Information Theory, Shannon (1948) states there is a limited capacity of people to work with information and news, even though they are freely available, and emphasise the value of information in the transmission of messages. Consumers are less satisfied, less confident and more confused thanks to overload of on-line information (Lee and Lee, 2004) and attention is becoming a scarce source. In this sense, macroeconomic environment (and particularly consumption, investments and prices, employment or asset returns) is influenced by rational inattention of economic agents when they are deliberately inattentive to some news in a decision-making process as they simply are not able to absorb them all. As Carr (2004) states, agents manage the excessive volume of information in a way that they prioritise some information to be able process them. The theory of *rational inattention* is largely discussed by Sims (1998, 2003, 2006, 2010) who mentions the problem of limited attention of economic agents who are not able to absorb all news and make sense of it in times of information overload, i.e. much more information than they are able to work with. As such, rational inattention produces imprecise responses of agents done in discrete jumps or simply randomly which causes subsequent slow adjustment of macroeconomic variables and misleading results of macroeconomic models and attention could be considered as a scarce cognitive source with specific subjective rules of its allocation and agent try to decide to find an optimum (i.e. rationally).

There is also a stream of strictly theoretical studies proposing various models of rational inattention. Matějka and McKay (2015) focus on the discrete choice behaviour of an economic agent who faces a problem that he must optimally allocate a limited attention to the all available information about the specific choice situation when there are some costs to acquire information. Ellis (2018) works with an agent with a limited information attention and develops a model of an agent reacting optimally to limited attention to provide both the description of the implications of observable choice behaviour and its justification. For models considering consumers under the optimal inattention model, see also Gabaix (2014), De Oliveira et al. (2017), Gul et al. (2017) or Saint-Paul (2017).

Psychological stream of literature focuses on the problem of *selective attention* or *selection exposure hypothesis* when economic agents attend to a limited amount of information or they simply ignore some of them, i.e. they do not behave rationally but select to which information they respond and to which not. The reason may be that agents do not choose optimal decision because of procrastination and obedience and then produce selective and wrong decision (Akerlof, 1991) or information may be assessed as threatening (Caplin, 2003) or negative (Karlsson

et al., 2009) and, therefore, agents refuse collecting additional information. This phenomenon is sometimes called as the *ostrich effect* which is defined by Galai and Sade (2006, p. 2741) as behaviour when investors try to avoid "*…apparently risky situations by pretending they do not exist*". As such, financial investors look for information differently in periods of financial booms and downturns, i.e. there may be some delay in this information-seeking process, and that investors pay more attention to their portfolios and have a tendency to look for information particularly when financial markets are rising while they ignore the information when markets are down and they may potentially face losses (Karlsson et al., 2009).

3. Data and Methods

We analyse forecasting performance of the 6 most traded Forex currency pairs (CAD/USD, JPY/USD, USD/AUD, USD/EUR, USD/GBP, USD/NZD) in the period from 1979Q1 to 2016Q4. We use four groups of exchange rate predictors[2]. First, we follow Taylor (1995) and define mainstream macroeconomic imbalances based on inflation differential (CPI indexes), interest rate differential (3-month interbank interest rate), monetary and portfolio balance model (M1 monetary aggregates, real GDP, and trade balances). Second, we consider that foreign currency demand is significantly affected by expectations about the future volatility and portfolio rebalancing, especially after the financial crisis in 2007. Therefore, we include VIX indexes (EUVIX, JYVIX, and BPVIX) and stock market indexes (DAX, S&P 500, Nikkei 225, FTSE 250, SMI PR, S&P/TSX, S&P/ASX 200, S&P/NZX 50) into our models.

Third, we focus on the impact of economic agents' attention which reflects demand and attractiveness of the topics related to the selected currency pairs. We use Google Trends statistics which provide information about search intensity of selected phrases (volume index of internet search queries in range from 0 to 100).[3]

Fourth, we focus on selective attention to news about the macroeconomic fundamentals related to the selected currency pairs. We follow Baker et al (2016) and develop indices calculated as counts of news articles related to four different categories: economic activity, money, price and trade. We use data from Proquest Database containing more than 315 million news articles related to analysed currency pairs in 3500 English-language newspapers. For the each

[2] We use publicly available datasources: XE.COM, OECD, Eurostat, FRED, CBOE, Yahoo Finance, and Bloomberg Database. Detailed description of the all regressors are provided in the Appendix, Table A1. All the analysed time series are transformed by log differences.

[3] The normalised search query index at a given point in time is a ratio of the total search volume for each query to the total number of all search queries. We use keywords "Australian Dollar", "Canadian Dollar", "British Pound", "Euro", "Japanese Yen", "New Zealand Dollar", "United States Dollar" with emphasize on the searches in the category "Currency".

currency pair we create 5 indices: (1) *output* ("gdp", "output", "recession", "production") giving 21.4 thousand articles; (2) *money* ("money", "interest rate", "monetary", "central bank") giving 55.7 thousand articles; (3) *price* ("price", "inflation", "deflation", "cpi") giving 33 thousand articles; (4) *trade* ("trade", "export", "import") giving 25.2 thousand articles, (5) and *total* (all keywords) giving 100.5 thousand articles. Detailed search conditions are specified in the Appendix, Table A2.

We assume time-varying reactions of exchange rates to the market information and macroeconomic fundamentals with possible endogeneity biases. Moreover, we hypothesize that market participants are overwhelmed by information and they pay time-varying selective attention to predictors. Following these assumption we employ Dynamic Model Averaging (DMA) and Dynamic Model Selection (DMS) approaches (Koop and Korobilis, 2012) and estimate time-varying probability to include selected regressors into the model. We employ Kalman filter to estimate time-varying parameter model

$$\begin{aligned} y_t &= z_t\theta_t + \varepsilon_t \\ \theta_t &= \theta_{t-1} + \eta_t \end{aligned} \tag{1}$$

where y_t represents log returns of selected currency pair and z_t contains all predictors, lagged returns and intercept

$$z_t = \phi + \gamma y_{t-1} + \beta X_{t-1} \tag{2}$$

where *X* represents vector of macroeconomic fundamentals, volatility indices, stock return differentials, search volume indices and indices calculated from news articles. We follow Koop an Korobilis (2012) and define *K* models as predictors $z_t^{(k)}$ for $k = 1, \dots, K$. Thus, $z_t^{(k)}$ is subset of z_t and the set of models (1) is rewritten as

$$\begin{aligned} y_t &= z_t^{(k)}\theta_t^{(k)} + \epsilon_t^{(k)} \\ \theta_t^{(k)} &= \theta_{t-1}^{(k)} + \eta_t^{(k)} \end{aligned} \tag{3}$$

for each currency pair *y*. Thus, we have $K = 2^{m\tau}$ models for *m* explanatory variables in each model and rolling forecasts which use estimation of $\hat{\theta}$ using data from $\tau - \tau_0$. Let $L_t \in \{1,2, \dots, K\}$ denote which model applies at time *t*, and average weighted DMA point forecasts given available data in $t-1$ as

$$E(y_t|y^{t-1}) = \sum_{k=1}^{K} \pi_{t|t-1,k} z_t^k \hat{\theta}_{t-1}^{(k)} \tag{4}$$

where $\pi_{t|s,l} = Pr(L_t = l|y^s)$. We calculate time-varying probability to include the predictors into the model as

$$p(\Theta_{t-1}|y^{t-1}) = \sum_{k=1}^{K} p\left(\theta_{t-1}^{(k)}\middle|L_{t-1}=k, y^{t-1}\right) Pr(L_{t-1}=k|y^{t-1}), \quad (5)$$

where $p\left(\theta_{t-1}^{(k)}\middle|L_{t-1}=k, y^{t-1}\right)$ is given by $\Theta_{t-1}|\mathrm{L}_{t-1}=k, y^{t-1}$. Finally, we employ Dynamic Model Selection (DMS) based on the averaging over predictive results for every model selecting the highest value for $\pi_{t|t-1,k}$ at each point of time. Moreover, we follow Raftery et al. (2010) to involve a forgetting factor which implies that observations in specific period in the past have weight $0 < \lambda_j < 1$[4]. Finally, as a robustness check, we compare DMA and DMS results with time-varying parameter VAR (TVP-VAR) and report mean squared forecast error (MSFE), mean absolut forecast error (MAFE), and the sum of predictive likelihoods (log(PL)) which represents predictive density for y_t given data in time $t-1$ (Geweke and Amisano, 2011).

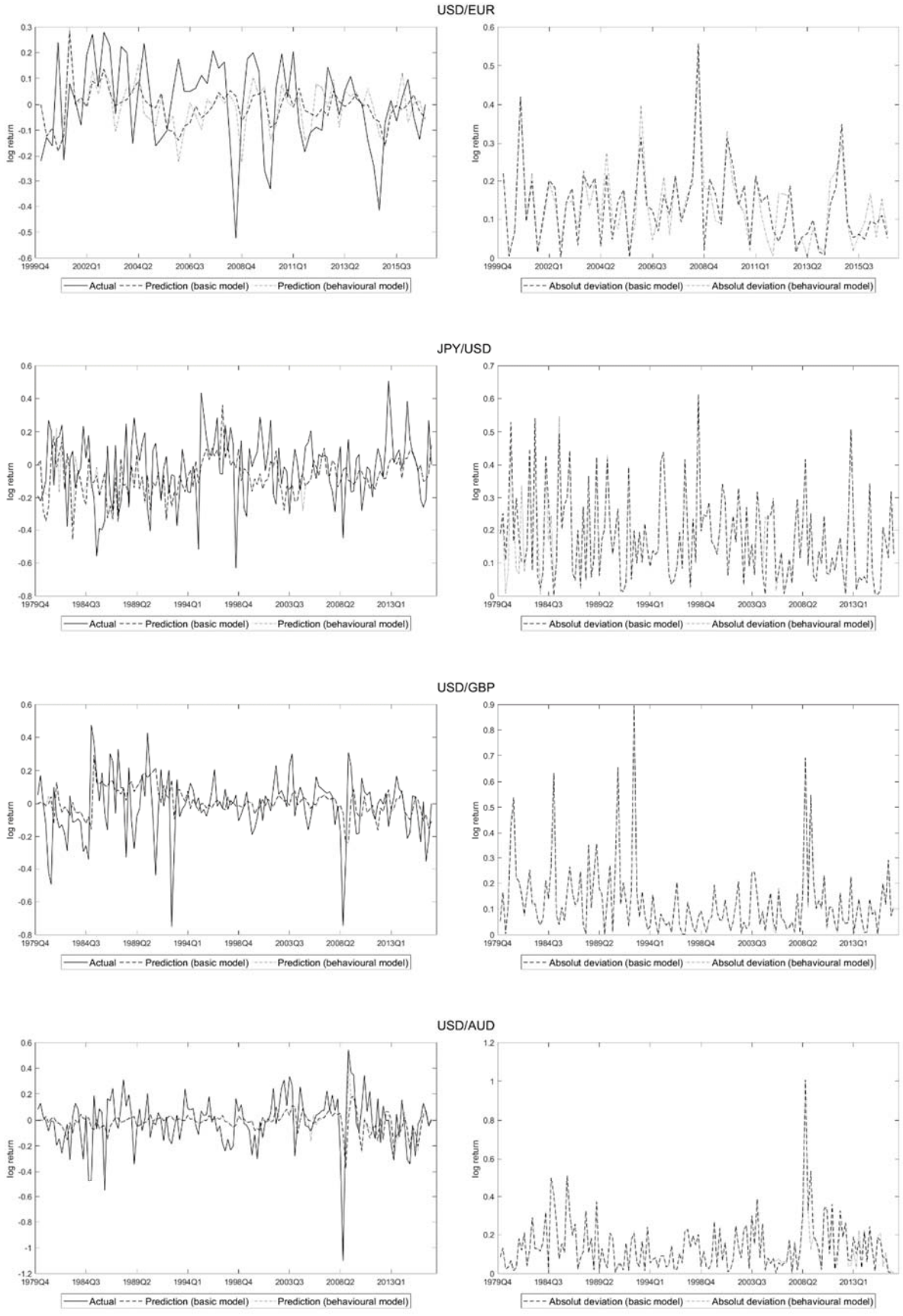
4. Results

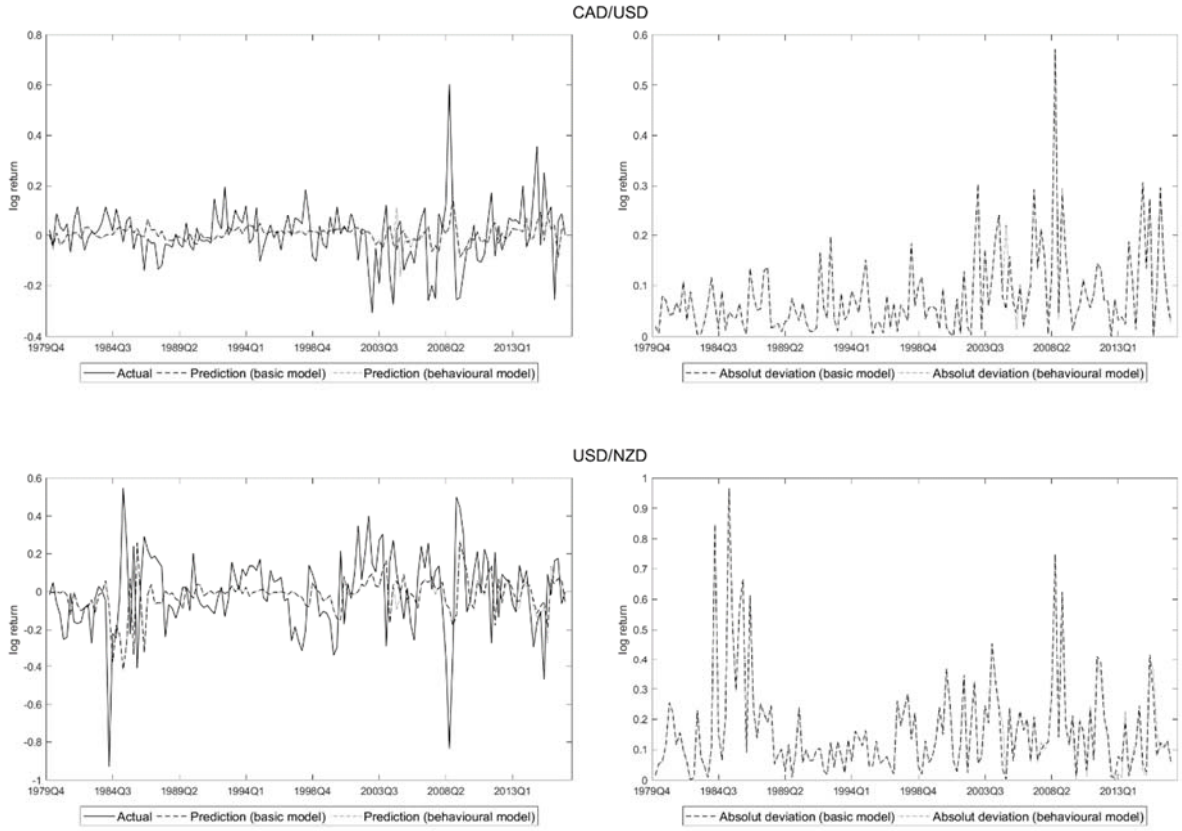
Figure 1 shows actual and predicted data from basic (with macro fundamentals) and behavioural (with both macro fundamentals and behavioural variables) model together with a deviation for these two models for all currency pairs. While left figure plots the data, the right figure plots the deviations between the actual and predicted value of the individual exchange rate. It is apparent that actual data are much more volatile compared to predicted data which could be explained by the fact mentioned by many authors that any prediction model is not able to encompass all the variables influencing the exchange rate movement.

Interesting results are illustrated in right figures where we compare the absolute deviation of the prediction compared with actual data for both the basic and behavioural model. At first sight, both deviations are almost the same for all currency pairs, however, there are significant differences for some currency pairs. In case of USD/EUR pair, the deviations are the most apparent; while the predictions produced by the basic model are less deviated in the first half of the analysed period, the predictions of the behavioural model are more precise in the second half of the period, i.e. in 2006 and particularly after the financial crisis of 2007 and 2008 (with some occasional exceptions, e.g. 2012Q2, 2015Q4 or 2016Q2); these findings confirm the fact that the role of news and behavioural factors has increased in recent decades and supports the idea that these factors should be incorporated into forecasting models to improve their predictive capacity. The same holds for the USD/AUD pair and partly for the USD/NZD pair; we can see that the deviation of the prediction is smaller in the second half of the analysed period and

[4] We follow Koop and Korobilis (2012) and set parametr $\lambda = 0.99$ which ensure that observations five years ago $\approx 80\%$ as much weight as the last period's observation.

Figure 1: Actual and predicted data from basic and behavioural model





particularly after the crisis (with some slight exceptions again). In case of some currency pairs (JPY/USD, USD/GBP or CAD/USD), the situation is not so convincing and both forecasts are almost identical, i.e. behavioural factors do not change the quality of the forecasting model so much. In case of Japanese yen, these results may be cause by the fact that we use news only from archives of journals and magazines in English and not Japanese.

In the next step, we estimate time-varying probability to include different regressors in the time-varying VAR model. As such, we could show that the importance of individual variables influencing the exchange rate fluctuate over time which supports the use of the method of DMA and DMS. Figure 2 plots these estimates of these probabilities for individual variables which could be potentially included in the forecasting model; left figures present estimates for macro fundamentals, volatility indices, and stock return differences, right figures for behavioural variables denoted as "News" (indices based on article news) and "Searches" (data from Google Trends database).

Generally, the probability of inclusion in case of past values of the exchange rate (variable *ER*(*t*-1)) increased over time and approached almost 1 (except for USD/GBP and USD/NZD in which case the probability increased only before the financial crisis in 2008 and in Japan again in 2013 when it reached 0,9), i.e. we can state that past values of exchange rate could influence the current or predicted value of the exchange rate relatively substantially. In case of USD/EUR,

the increase of the probability from the euro creation could be explained by a strong appreciation of euro relative to US dollar from 2000 to 2008 and then by the financial crisis. The sudden drop in probability in case of USD/NZD in 1985 could be caused by the change of the exchange rate regime from fixed to floating regime in New Zealand in this year. However, the same step, i.e. the change from fixed to floating exchange rate regime in 1983, led to a one-year rise in probability of *ER*(*t*-1) (and a simultaneous one-year drop in case of *Inflation diff*) in 1984 in case of USD/AUD, but then it decreased and stayed at the same level in subsequent years (till 1997) as in case of USD/NZD. Thus, we could state that the implementation of floating regime decreased the probability of inclusion of this variable in the forecasting models of these two currency pairs. It is interesting, that we face a higher increase of probabilities in case of USD/AUD and USD/NZD around 1998, probably as a reaction to the creation of euro currency in 1999 and Asian crisis of 1997 and 1998 but it remained at a high level only in case of USD/AUD till the end of the period (while in case of USD/NDZ, it became very volatile). A strong depreciation of AUD against USD in 2003 also increased the probability of the CAD/USD currency pair in this year; this situation continued in subsequent years thanks to the US budget and current account deficits.

The role of interest rate differential (*IR diff*) is almost negligible in case of USD/EUR and was below 0,5 in case of USD/AUD, CAD/USD and partly in case of USD/NZD (except for the period 1979-1988). The only currency pair, where this variable played some role from the beginning of analysed period till approximately 2003, was JPY/USD; however, the probability of inclusion of *IR diff* step by step decreased after 2003. For a limited time, the probability was higher in case of USD/GBP in the second half of 80s till approximately 1992 when interest rates in the UK started decreasing and in case of USD/EUR in 2004 and 2005 (probably as a result of higher Federal Funds Rate which started in June 2004 and continued till June 2006 as a reaction to rising house prices and first signals in house price bubble), however, the level of the probability was still very low. There were also two separate jumps of the probability in 1980 and 1985 in case of USD/NZD (which may be explained by the above-mentioned switch from fixed to floating exchange rate regime in 1985) with a long-term decreasing tendency in 90s (and simultaneously an increasing tendency of the *ER*(*t*-1) variable).

The importance capital markets (*Stock ret diff*) in case of USD/EUR is at the highest level compared to other currency pairs (the value of probability is almost 1 for the analysed period which is from 1999 in this case) which signals that capital markets played a significant role in the exchange rate movement, i.e. this variable was added to the prediction model for this time

and explained the variability of the USD/EUR exchange rate. It is interesting that the probabilities became important in case of USD/GBP, USD/AUD and also USD/NZD in the period after the financial crisis. This fact could signal the effect of portfolio rebalancing in case of these three currency pairs in the crisis period when traditional macroeconomic variables became less important and capital market variables more important as investors moved their portfolios to other capital markets (then in Europe or the US) to the UK or even to smaller markets in Australia or New Zealand (see also results of robustness analysis). In case of JPY/USD, the probability was continuously decreasing in this period (the probability was between 0,2 and 0,5 during 80s partly as a reflection of financial market bubble illustrated by a strongly rising stock price index Nikkei in Japan between 1983 and 1989 when it was eliminated by monetary policy tightening).

GDP differential (*GDP diff*) probabilities give ambiguous results: (1) stable probability in case of USD/EUR around 0,5; (2) rising probability in case of USD/AUD through the period; (3) jumping probabilities in case of USD/GBP in 1984, 1989-1990 and particularly after the financial crisis; (4) high probability in case of CAD/USD in the period from 1985 to 1994 and then higher probability also after the financial crisis; and finally (5) quickly jumping and dropping probabilities in case of USD/NZD during 80s (caused probably by economic reforms forced by rising unemployment and economic stagnation) with an increasing tendency since then. In case of JPY/USD, this variable was strongly insignificant which is not surprising when we consider the long-term economic stagnation in Japan particularly in 1993-2003.

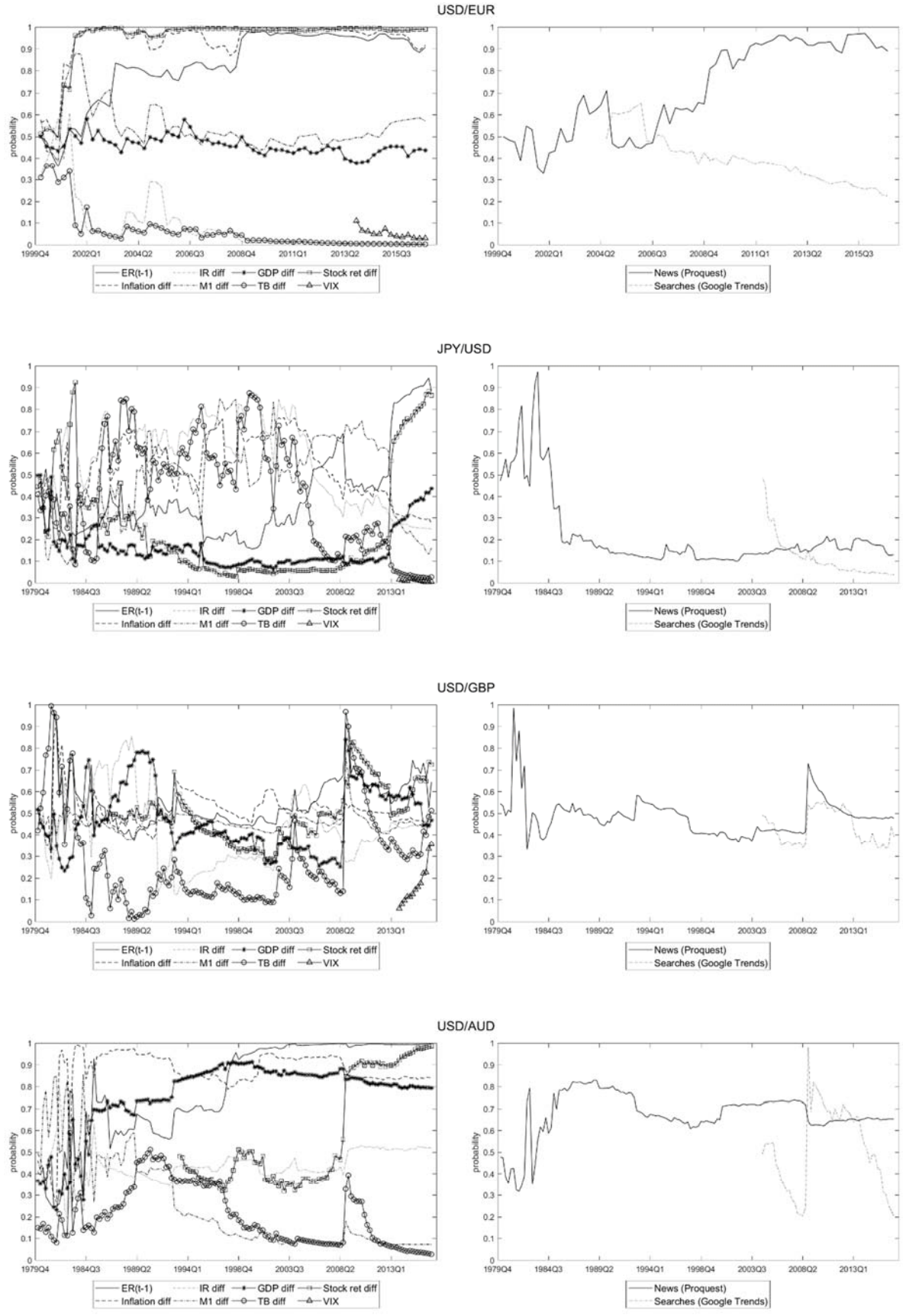
The highest level of probability of inflation differential (*Inflation diff*) was estimated in case of USD/EUR particularly after 2001 which could reflect the focus of monetary policy in Euro Area and then in case of USD/AUD (where economic agents perceived poor results concerning combating inflation particularly in 80s and 90s) with a slightly decreasing probability after the financial crisis. Probabilities around 0,5 were estimated also in case of JPY/USD (which could reflect the fact that inflation/deflation policy in Japan stayed in the centre of attention of both economic agents and policy makers) but with a decreasing tendency after the policy of quantitative easing was implemented in 2001 and also in case of USD/GBP. The probability was rising in years preceding 1983 in case of USD/NZD with the highest values between 8.0 and 0.95 at the end of this period then it fell to almost 0 in 1984 and then grew continuously to approximately 0,5 when liberalisation tendencies concerning monetary policy and the preparation of the inflation targeting regime implementation (from 1990) started in New Zealand. In case of CAD/USD, the probability was higher during 80s but then it fell to a level about 0.4.

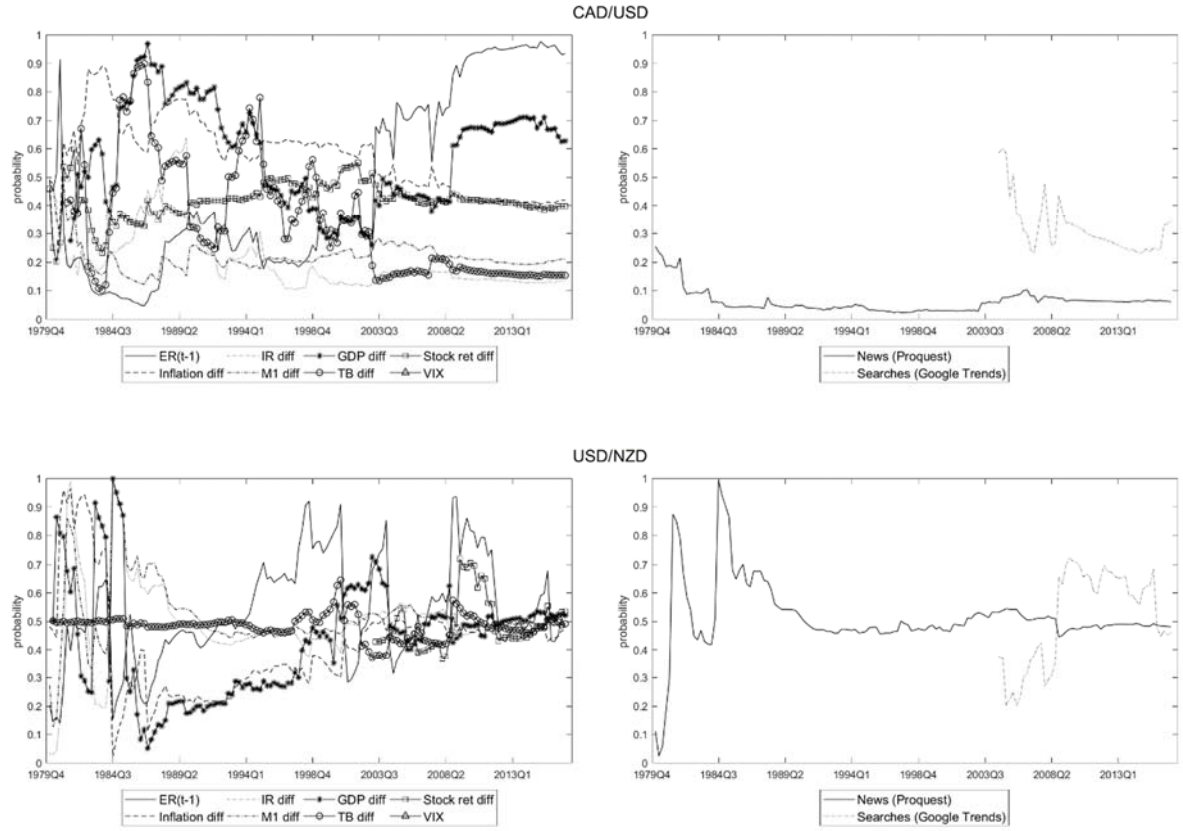
Money supply differential (*M1 diff*) had a long-term impact on the JPY/USD exchange rate where the probability increased in 1989 and again in 1995 probably as a result of monetary policy tightening (after the bubble times in 80s characterised by rising land and stock prices) and then again in 2000 before the implementation of the unconventional monetary policy in Japan. The first policy of quantitative easing introduced in 2001 was replaced by comprehensive monetary easing in 2010. Then, the new policy of quantitative and qualitative easing with yield curve control was applied in April 2013 which may explain the drop of the probability of inclusion of the money supply variable in the forecasting model in that period. The probability of *M1 diff* was higher in case of USD/AUD in the first half of 80s, but it continuously decreased after 1989 (Australia abandoned the money supply targeting regime in 1985) and particularly after 1993 when the first inflation target was set. Generally, the role of money supply was diminishing throughout the time in case of all country pairs thus reflecting the deflection from the monetary transmission mechanism using a monetary base as an instrument influencing money supply and the implementation of inflation targeting regimes instead during late 80s or early 90s. In case of USD/EUR, USD/GBP and from approximately 1990 also in case of USD/NZD, the probability is fluctuating around 0,5. We estimate a really low probability in case of CAD/USD.

The probability of trade balance differential (*TB diff*) was relatively high but very volatile in case of JPY/USD till approximately 2005 when it dropped to almost 0. In case of CAD/USD, a higher probability in the second half of 80s may have been a reflection of the report of the McDonald Commission in 1985 followed by negotiations of the Canada-US Free Trade Agreement which was then prepared in 1987 and signed in January 1988. In case of USD/EUR, the probability was not sufficiently high and reached almost 0. The level of the probability was stable only in case of USD/NZD and was estimated around 0,5. However, the probability of this variable was apparently volatile with occasional jumps and drops in case of the other country pairs.

The VIX index (*VIX*) which represents the market's expectation of future volatility had a low probability (besides the limited data availability). The role of behavioural variables (right figures) could also be assessed as ambiguous; we can see relatively high probabilities of news in case of USD/EUR, USD/AUD and also in case of USD/GBP and USD/NZD throughout the period and in case of JPY/USD at the beginning of the period. Moreover, we can see a rising influence of Google searches in case of USD/NZD or a stable probability in case of USD/GBP and CAD/USD or a high jump in case of USD/AUD during the financial crises and a short episode of high probability in case of USD/EUR in 2005 and 2006 with a subsequent decrease.

Figure 2: Time-varying probability of inclusion of predictors (behavioural model)





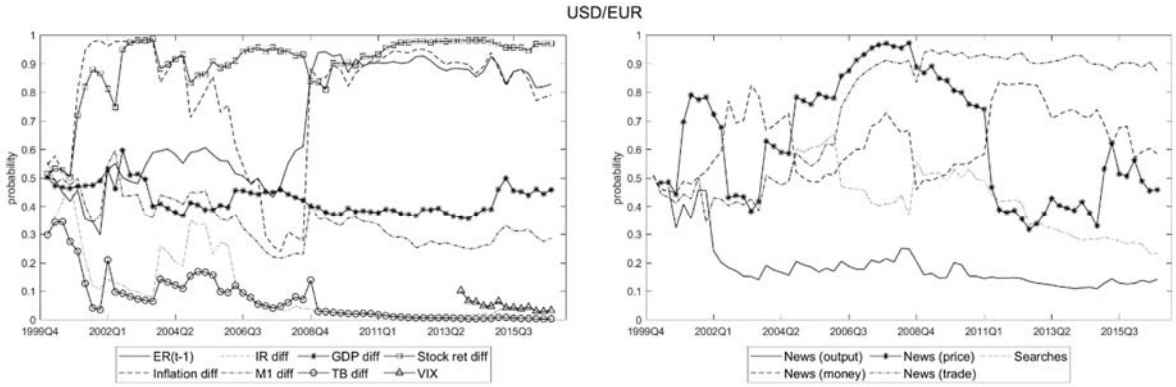
To sum these results up, it is apparent that macro fundamentals played a significant role in the exchange rate determination, however, behavioural factors were also estimated as significant and were added into our models explaining the exchange rate determinants. The only two exceptions are JPY/USD and CAD/USD with a relatively low level of probabilities of inclusion throughout the analysed period. It could be interpreted by the fact that Japan is often considered being a safe heaven for financial investors. In case of Canada, it is probably a result of a relatively small importance of this financial market in the world.

In Figure 3, we present estimations of probabilities of the extended behavioural model where right figures contain probabilities for individual categories of news (i.e. output, price, money and trade) and for Google searches. The probabilities of all categories were relatively stable and very often at same level for almost all country pairs in the analysed period except for USD/EUR where the probabilities differ significantly and partly in case of USD/AUD. High volatility of all categories was apparent at the beginning of the analysed period in case of USD/NZD before economic reforms in New Zealand were adopted. The probability of the news in category "price" was relatively volatile and high in case of USD/EUR (particularly in 2001 and then in 2005-2010 the probability was higher than 0.8). News in category "money" was also volatile in case of USD/EUR, however, the trend is opposite except for 2006-2008 when the probabilities rose and fell together. In 2008, the probability of Google searches jumped to

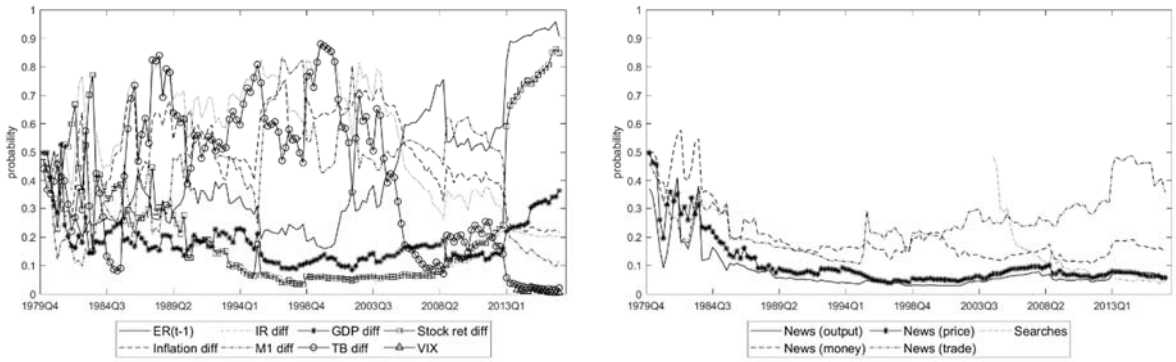
almost 1 in case of USD/AUD, to 0.7 in case of USD/NZD, to 0.6 in case of USD/GBP or fluctuated between 0.2 and 0.45 in case of CAD/USD. On the other hand, the probability dropped in years preceding 2008 in Japan which confirms the fact that Japan is often regarded as an investment safe heaven. We cannot say that there is one category (compared to other categories) with the highest or lowest probability in this period as the probabilities varied in time. For example, the lowest probability of category "output" among other categories was estimated in case of USD/EUR and JPY/USD while the highest probability of the same category was estimated in case of CAD/USD. On the other hand, we can see the highest probability of news in the category "trade" in case of USD/EUR and JPY/USD in the second half of the period while it was lowest in case of CAD/USD from 1988. In case of USD/AUD, the highest probability was estimated for news in the category "price" (the probability increased before the regime of inflation target was adopted).

When we summarise the results of this step of our analysis, it is evident that the decomposition of one general index of news into individual categories does not bring any extra findings refining our previous results except USD/EUR currency pair. Thus, we conclude that USD/EUR exchange rate was influenced significantly by news about prices during the years 2006–2008 when the ECB decided to start increasing its policy rates because their monetary analysis indicated upward risks to price stability. After the financial crisis in 2007 impact of news about trade and primarily output prevail.

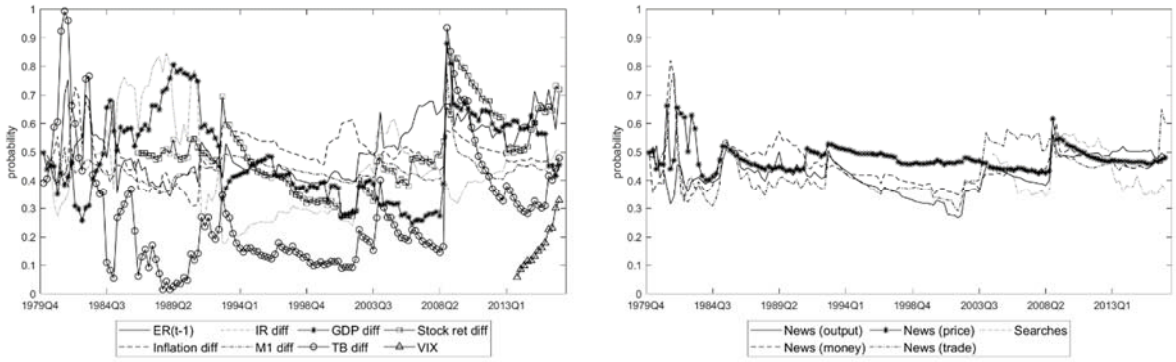
Figure 3: Time-varying probability of inclusion of predictors (extended behavioural model)



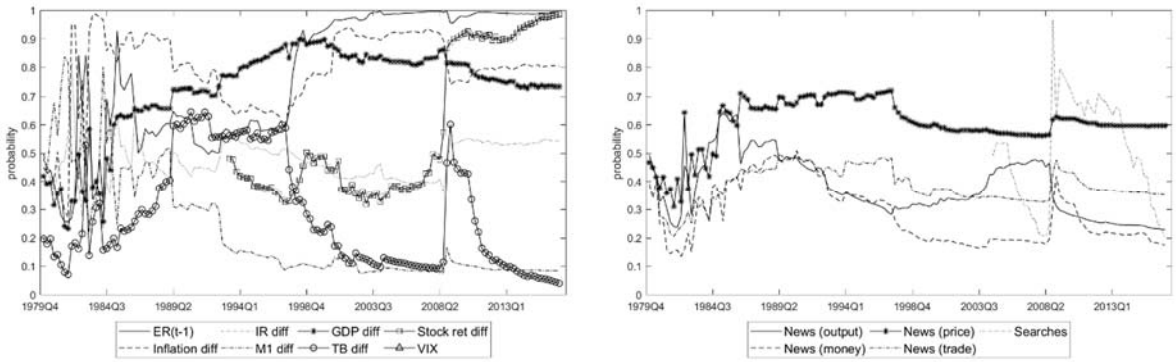
JPY/USD



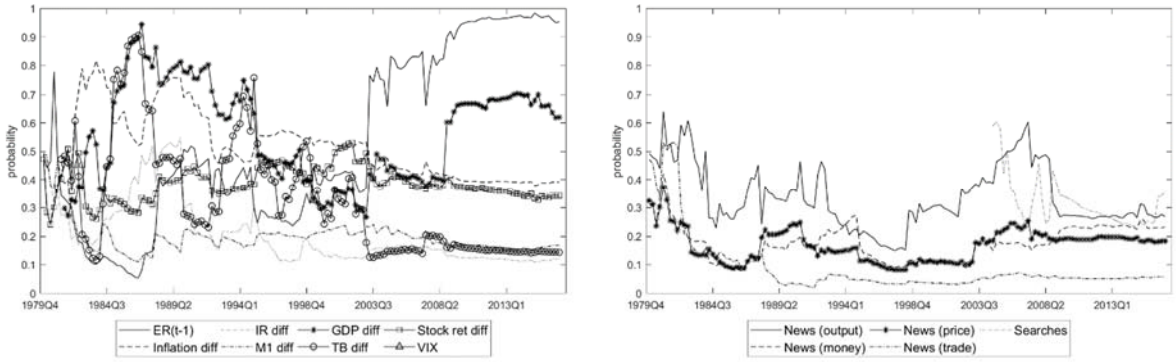
USD/GBP

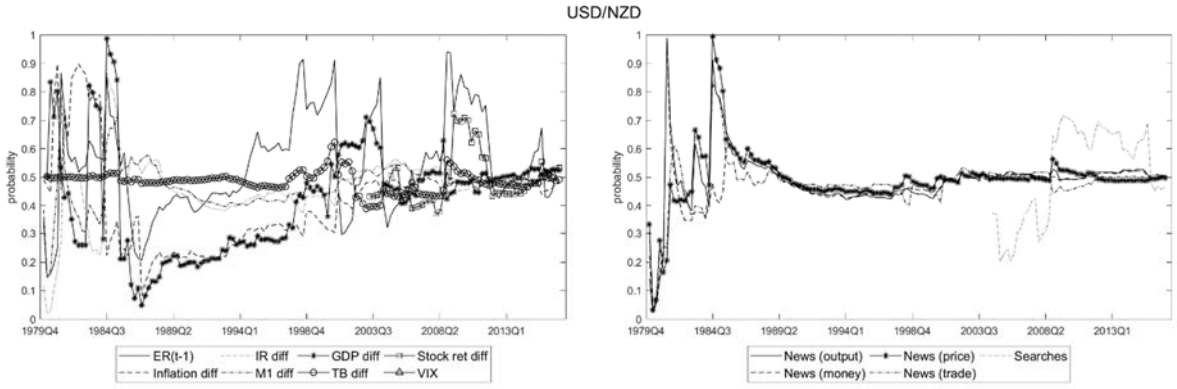


USD/AUD



CAD/USD





5. Robustness Analysis

The robustness analysis compares forecasting performance of the basic model and two behavioural models using DMA (Dynamic Model Averaging), DMS (Dynamic Model Selection) and TVP-VAR (time-varying parameter VAR) reporting MAFEs (Mean Absolute Forecast Errors), MSFEs (Mean Squared Forecast Errors), and the sum of predictive likelihoods (log(PL)). We consider the same lag (1) as in the previous analyses. Our results (Table 1) show increasing forecasting performance of behavioural models (including article news and Google searches) employing DMA and DMS methods.

Table 1: Actual and predicted data from basic and behavioural model

Currency pair	Model	TVP-VAR			DMA			DMS		
		MAFE	MSFE	log(PL)	MAFE	MSFE	log(PL)	MAFE	MSFE	log(PL)
USD/EUR	Basic	10.647	1.720	0.326	10.947	1.668	1.467	10.711	1.628	1.827
	Behavioural	11.138	1.817	0.040	10.848	1.648	1.379	10.653	1.617	1.566
	Behav. Extended	12.072	1.882	1.176	10.632	1.600	1.797	10.435	1.595	2.190
JPY/USD	Basic	25.045	2.701	0.364	23.711	2.506	4.274	26.123	2.826	6.459
	Behavioural	25.851	2.775	-0.723	23.575	2.478	3.555	24.871	2.679	4.599
	Behav. Extended	25.743	2.793	-2.024	23.561	2.464	3.389	26.180	2.839	5.613
USD/GBP	Basic	22.758	2.683	-4.482	19.315	2.287	-6.610	19.796	2.278	-4.827
	Behavioural	23.163	2.720	-4.525	18.976	2.267	-6.595	19.433	2.286	-5.190
	Behav. Extended	24.928	3.012	-5.120	19.606	2.335	-6.997	20.761	2.420	-5.027
USD/AUD	Basic	24.605	2.906	2.993	20.540	2.357	0.251	20.873	2.404	0.212
	Behavioural	25.303	3.199	3.937	19.979	2.288	-0.160	20.292	2.321	0.173
	Behav. Extended	25.899	3.263	4.884	20.339	2.326	0.076	20.953	2.377	-0.291
CAD/USD	Basic	11.604	1.460	-0.357	10.687	1.344	-0.370	10.975	1.358	-0.210
	Behavioural	11.572	1.461	-0.405	10.764	1.350	-0.349	11.104	1.367	-0.135
	Behav. Extended	12.050	1.518	-0.194	10.667	1.347	-0.317	11.183	1.373	-0.152
USD/NZD	Basic	24.959	2.971	2.412	22.694	2.560	0.504	22.971	2.582	0.872
	Behavioural	25.690	3.020	2.687	22.520	2.524	0.299	23.143	2.613	0.017
	Behav. Extended	26.278	3.083	2.176	22.829	2.594	0.860	24.076	2.799	1.103

To summarise the results of the first-step analysis, the inclusion of article news and Google searches in the prediction models using DMA/DMS methods leads to more precise predictions in case of most currency pairs. In comparison to TVP-VAR approach, forecasting errors decreased because we reduce uncertainty of model selection following assumption of selective attention, and keeping models in smaller sizes.

We also show that splitting news article into individual groups does not help to increase the forecasting performance of the all exchange rates (behavioural extended models) except CAD/USD and JPY/USD. However, the additional extension of the models could provide reasonable contribution for policymakers.

6. Discussion and Conclusions

The recent empirical models explaining the determination of exchange rate fail very often to predict the future value of exchange rate even they work not only with macroeconomic fundamentals but also incorporate the news announcements, sentiment, uncertainty, or attention. We contribute with *selective attention* hypothesis testing and show that smaller sizes models offer better forecast performance. We argue that market participants suffer from information overload and, therefore, are prone to be rationally inattentive or select only specific information. Therefore, we employ Dynamic Model Averaging and Dynamic Models Selection methods and estimate time-varying probability to include specific predictors into our models. After that we apply time-varying parameter VAR and produce one-step ahead forecasts at each point of time.

Comparison of our point forecasts with actual data confirms the importance of behavioural predictors. The forecasting performance (measured by mean absolute forecast errors, mean squared forecast errors, and the sum of predictive likelihoods) increased after we include indices constructed from article news and Google searches trends data. However, time-varying probability to include predictors into the models also confirm the role of macroeconomic fundamentals which are often dependent on the changes in domestic economic policy or world crises.

References

Akerlof, G.: Procrastination and Obedience. *The American Economic Review* **81**, 2, 1991, 1-19.

Bacchetta, P., van Wincoop, E. On the unstable relationship between exchange rates and macroeconomic fundamentals. *Journal of International Economics* **91**, 1, 2013, 18-26.

Bacchetta, P., van Wincoop, E. Can Information Heterogeneity Explain the Exchange Rate Determination Puzzle? *American Economic Review* **96**, 3, 2006, 552–576.

Bacchetta, P., van Wincoop, E. A Scapegoat Model of Exchange-Rate Fluctuations. *American Economic Review* **94**, 2, 2004, 114-118.

Bachman, R., Bayer, C.: Uncertainty business cycles – Really? *National Bureau of Economic Research Working Paper* **16862**, 2011.

Baker, S. R., Bloom, N., Davis, S. J.: Measuring Economic Policy Uncertainty. *The Quarterly Journal of Economics* **131**, 4, 2016, 1593-1636.

Balke, N. S., Ma, J., Wohar, M. E. The contribution of economic fundamentals to movements in exchange rates. *Journal of International Economics* **90**, 1, 2013, 1-16.

Beckmann, J., Czudaj, R.: Exchange rate expectations and economic policy uncertainty. *European Journal of Political Economy* **47**, March, 2017, 148-162.

Bilson, J. F. O. The Monetary Approach to the Exchange Rate: Some Empirical Evidence. *International Monetary Fund Staff Papers* **25**, 1, 1978, 48-75.

Bloom, N.: Fluctuations in Uncertainty. *Journal of Economic Perspectives* **28**, 2, 2014, 153-176.

Bloom, N.: The impact of uncertainty shocks. *Econometrica* **77**, 3, 2009, 623-685.

Breedon, F., Vitale, P.: An empirical study of portfolio-balance and information effects of order flow on exchange rates. *Journal of International Money and Finance* **29**, 3, 2010, 504-524.

Bulut, L.: Google Trends and Forecasting Performance of Exchange Rate Models. *IPEK Working Paper*, 15–05 (2015). Available at: http://econpapers.ipek.edu.tr/IpekWPArchives/wp2015/wp1505Bulut.pdf.

Burns, K., Moosa, I. A. Enhancing the forecasting power of exchange rate models by introducing nonlinearity: Does it work? *Economic Modelling* **50**, November, 2015, 27-39.

Caplin, A.: Fear as a policy instrument. In: Loewenstein, G., Read, D., Baumeister, R. (eds.): *Time and decision*. New York: Russell Sage, 2003.

Caporale, G. M., Spagnolo, F., Spagnolo, N.: Macro news and exchange rates in the BRICS. *Finance Research Letters* **21**, May, 2017, 140-143.

Carr, T. H.: A multilevel approach to selective attention: Monitoring environmental space, choosing stimuli for deep processing, and retrieving information from memory. In: Posner, M. I. (ed.): *Cognitive neuroscience of attention*. New York: Guilford Press, 2004, 56-70.

Cavusoglu, N., Neveu, A. R. The Predictive Power of Survey-Based Exchange Rate Forecasts: Is there a Role for Dispersion? *Journal of Forecasting* **34**, 2015, 337-353.

Cerra, V., Saxena, S. Ch. The monetary model strikes back: Evidence from the world. *Journal of International Economics* **81**, 2, 2010, 184-196.

Cerrato, M., Sarantis, N., Saunders, A. An investigation of customer order flow in the foreign exchange market. *Journal of Banking and Finance* **35**, 8, 2011, 1892-1906.

Chang, M-J., Su, Ch.-Y. The Dynamic Relationship between Exchange Rates and Macroeconomic Fundamentals: Evidence from Pacific Rim Countries. *Journal of International Financial Markets, Institutions and Money* **30**, May, 2014, 220-246.

Chen, S.-S., Chou, Y.-H. Revisiting the relationship between exchange rates and fundamentals. *Journal of Macroeconomics* **46**, December, 2015, 1-22.

Chen, K., Zhang, S. What's news in exchange rate dynamics: A DSGE approach. *Economic Letters* **134**, September, 2015, 133-137.

Cheung, Y.-W., Chinn, M. D., Pascual, A. G.: Empirical Exchange Rate Models of the Nineties: Are Any Fit to Survive? *Journal of International Money and Finance* **24**, 7, 2005, 1150–1175.

Chinn, M., Frankel, J. Patterns in Exchange Rate Forecasts for Twenty-Five Currencies. *Journal of Money, Credit and Banking* **26**, 4, 1994, 759-770.

Christoffersen, P. F., Diebold, F. X.: Cointegration and Long-Horizon Forecasting. *Journal of Business and Economic Statistics* **16**, 4, 1998, 450–458.

Clarida, R.H., Waldman, D.: Is Bad News About Inflation Good News for the Exchange Rate? And, If So, Can That Tell Us Anything about the Conduct of Monetary Policy? In: Campbell, J.Y. (ed.). *Asset Prices and Monetary Policy*. Chicago: University of Chicago Press, 2008.

Dabrowski, M. A., Papiez, M., Smiech, S. Exchange Rates and Monetary Fundamentals in CEE Countries: Evidence from a Panel Approach. *Journal of Macroeconomics* **41**, September, 2014, 148-159.

deOliveira, H., Denti, T., Mihm, M., Ozbek, K.: Rationally inattentive preferences and hidden information costs. *Theoretical Economics* **12**, 2, 2017, 621–654.

Dick, C. D., MacDonald, R., Menkhoff, L.: Exchange rate forecasts and expected fundamentals. *Journal of International Money and Finance* **53**, May, 2015, 235-256.

Dornbusch, R. Expectations and Exchange Rate Dynamics. *Journal of Political Economy* **84**, 6, 1976, 1161-1176.

Driskill, R. A., Sheffrin, S. S. 1981. On the mark: Comment. *American Economic Review* **71**, 5, 1068–1074.

Égert, B., Kočenda, E. The impact of macro news and central bank communication on emerging European forex markets. *Economic Systems* **38**, 1, 2014, 73-88.

Ellis, A. Foundations for optimal inattention. *Journal of Economic Theory* **173**, January, 2018, 56-94.

Engel, C., Hamilton, J. D.: Long Swings in the Dollar: Are They in the Data and Do Markets Know It? *American Economic Review* **80**, 4, 1990, 689–713.

Engel, C., Frankel, J.A.: Why Interest Rates React to Money Announcements: An Answer from the Foreign Exchange Market. *Journal of Monetary Economics* **13**, 1, 1984, 31-39.

Engel, C., Mark, N., West, K. D.: Exchange Rate Models Are Not as Bad as You Think. *NBER Working Paper* **13318** (2007).

Engel, C., West, K. D.: Accounting for Exchange Rate Variability in Present Value Models when the Discount Factor is Near One. *American Economic Review* **94**, 2, 2004, 119-125.

Engel, C., West, K. D.: Exchange Rates and Fundamentals. *Journal of Political Economy* **113**, 3, 2005, 485–517.

Evans, M. D. D. Order flows and the exchange rate disconnect puzzle. *Journal of International Economics* **80**, 1, 2010, 58-71.

Evans, M. D. D., Lyons, R. K. Order Flow and Exchange Rate Dynamics. *Journal of Political Economy* **110**, 1, 2002, 170–180.

Faust, J., Rogers, J. H., Wright, J. H.: Exchange Rate Forecasting: The Errors We've Really Made. *Journal of International Economics* **60**, 1, 2003, 35–59.

Faust, J., Rogers, J. H., Wang, S.-Y. B., Wright, J.: The high-frequency response of exchange rates and interest rates to macroeconomic announcements. *Journal of Monetary Economics* **54**, 4, 2007, 1051-1068.

Festré, A., Garrouste, P.: The 'Economics of Attention': A History of Economic Thought Perspective. *Oeconomia* **5**, 1, 2015, 3-36.

Frankel, J. A. On the Mark: A Theory of Floating Exchange Rates Based on Real Interest Differentials. *The American Economic Review* **69**, 4, 1979, 610-622.

Frankel, J. A. Tests of Monetary and Portfolio Balance Models of Exchange Rate Determination. In: John F. O. Bilson and Richard C. Marston, eds. *Exchange Rate Theory and Practice*. University of Chicago Press, 1984, 239-260.

Frankel, J. A., Rose, A. K.: Empirical Research on Nominal Exchange Rates. In: Grossman, G. M., Rogoff, K. (eds): *Handbook of International Economics*. Amsterdam: Elsevier, 1995.

Fratzscher, M., Rime, D., Sarno, L., Zinna, G. The scapegoat theory of exchange rates: the first tests. *Journal of Monetary Economics* **70**, March, 2015, 1-21.

Frenkel., J. A. A Monetary Approach to the Exchange Rate: Doctrinal Aspects and Empirical Evidence. *The Scandinavian Journal of Economics* **78**, 2, 1976, 200-224.

Gabaix, X.: A sparsity-based model of bounded rationality, applied to basic consumer and equilibrium theory. *Quarterly Journal of Economics,* 129, 4, 2014, 1661–1710.

Galai, D., Sade, O.: The "ostrich effect" and the relationship between the liquidity and the yields of financial assets. *The Journal of Business* **79**, 5, 2006, 2741-2759.

Geweke, J., Amisano, G. 2011. Hierarchical Markov Normal Mixture Models with Applications to Financial Asset Returns. *Journal of Applied Econometrics* **26**, 1–29.

Goddard, J., Kita, A., Wang, Q.: Investor Attention and FX Market Volatility. *Journal of International Financial Markets, Institutions and Money* **38**, September, 2015, 79–96.

Gourinchas, P. O., Rey, H.: International Financial Adjustment. *Journal of Political Economy* **115**, 4, 2007, 665–703.

Gul, F., Pesendorfer, W., Strzalecki, T.: Coarse competitive equilibrium and extreme prices. *American Economic Review* **107**, 1, 2017, 109–137.

Hardouvelis, G.: Economic News, Exchange Rates and Interest Rates. *Journal of International Money and Finance* **7**, 1, 1988, 23-35.

Hooper, P., Morton, J. Fluctuations in the Dollar: A Model of Nominal and Real Exchange Rate Determination. *Journal of International Money and Finance* **1**, 1982, 39-56.

Issac, A. G., Mell de, S. The real-interest-differential model after 20 years. *Journal of International Money and Finance* **20**, 4, 2001, 473-495.

Ito, T.: Foreign exchange rate expectations: micro survey data. *The American Economic Review* **80**, 3, 1990, 434-449.

Jurado, K., Ludvigson, S. C., Ng, S.: Measuring *Uncertainty. The American Economic Review* **105**, 3, 2015, 1177-1216.

Tversky, A., Kahneman, D.: Judgment under Uncertainty: Heuristics and Biases. *Science* **185**, 4157, 1974, 1124-1131.

Karlsson, N., Loewenstein, G., Seppi, D.: The ostrich effect: Selective attention to information. *Journal of Risk and Uncertainty* **38**, 2, 2009, 95-115.

Klein, M., Mizrach, B., Murphy, R. G. Managing the dollar: Has the Plaza Agreement mattered? *Journal of Money, Credit, and Banking* **23**, 2, 1991, 742-751.

Kočenda, E., Moravcová, M. Intraday Effect of News on Emerging European Forex Markets: An Event Study Analysis. *Institute of Economic Studies Working Paper* **20**, 2016. Available at: http://ies.fsv.cuni.cz/sci/publication/show/id/5518/lang/cs.

Koop, G., Korobilis, D. Forecasting inflation using dynamic model averaging. *International Economic Review* **53**, 3, 2012, 867-886.

Koop, G., Onorante, L.: Macroeconomic Nowcasting Using Google Probabilities [online]. European Central Bank, 2013. Available at: https://www.ecb.europa.eu/events/pdf/conferences/140407/OnoranteKoop_MacroeconomicNowcastingUsingGoogleProbabilities.pdf?e105896cfaba02ab33265ae4047d96be.

Kristoufek, L. What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. *PLOS ONE* **10**, 4, 2015, 1-15.

Lee, B.-K., Lee W.-N.: The effect of information overload on consumer choice quality in an on-line environment. *Psychology and Marketing* **21**, 3, 2004, 159-183.

Leitch, G., Tanner, J. E.: Economic Forecast Evaluation: Profits versus the Conventional Error Measures. *American Economic Review* **81**, 3, 1991, 580–590.

Loría, E., Sánchez, A., Salgado, U. New Evidence on the Monetary Approach of Exchange Rate Determination in Mexico 1994–2007: A Cointegrated SVAR Model. *Journal of International Money and Finance* **29**, 3, 2010, 540-554.

Lyons, R. K., Moore, M. J. An information approach to international currencies. *Journal of International Economics* **79**, 2, 2009, 211-221.

MacDonald, R., Taylor, M. P. The Monetary Approach to the Exchange Rate Rational Expectations, Long-Run Equilibrium, and Forecasting. *IMF Staff Papers* **40**, 1993, 89-107.

Maćkowiak, B., Wiederholt, M.: Business Cycle Dynamics under Rational Inattention. *The Review of Economic Studies* **82**, 4, 2015, 1502-1532.

Matějka, F., McKay, A.: Rational inattention to discrete choices: A new foundation for the multinomial logit model. *American Economic Review* **105**, 1, 2015, 272-298.

Mark, N.C. Exchange Rates and Fundamentals: Evidence on Long-Horizon Predictability. *American Economic Review* **85**, 1, 1995, 201–218.

Mark, N.C., Sul, D. Nominal Exchange Rates and Monetary Fundamentals: Evidence from a Small Post-Bretton Woods Panel. *Journal of International Economics*, **53**, 1, 2001, 29-52.

Meese, R. A. Currency Fluctuations in the Post-Bretton Woods Era. *Journal of Economic Perspectives* 4, 1, 1990, 117–134.

Messe, R. A., Rogoff, A. K. Empirical exchange rate models of the seventies: Do they fit out of sample? *Journal of International Economics,* 14, 1–2, 1983, 3–24.

Moosa, I., Burns, K.: Reappraisal of the Meese-Rogoff Puzzle. *Applied Economics* **46**, 1, 2014, 30–40.

Omrane, W. B., Savaşer, T.: The sign switch effect of macroeconomic news in foreign exchange markets. *Journal of International Financial Markets, Institutions and Money* **45**, November, 2016, 96-114.

Payne, R. Informed trade in spot foreign exchange markets: an empirical investigation. *Journal of International Economics* **61**, 2, 2003, 307–329.

Rossi, B.: Exchange rate predictability. *Journal of Economic Literature* **51**, 4, 2013, 1063-1119.

Seabold, S., Coppola, A.: Nowcasting Prices Using Google Trends: An Application to Central America. *World Bank Policy Research Working Paper* **3798**, 2015.

Raftery, A., Karny, M., Ettler, P. 2010. Online Prediction under Model Uncertainty via Dynamic Model Risk Metrics. Technical Document, Fourth Edition, available at http://www.riskmetrics.com/system/files/private/td4e.pdf.

Rapach, D. E., Strauss, J. K., Zhou, G.: Out-of-sample equity premium prediction: Combination forecasts and links to the real economy. *The Review of Financial Studies* **23**, 2, 2010, 821–862.

Rapach, D. E., Wohar, M. E. Testing the monetary model of exchange rate determination: new evidence from a century of data. *Journal of International Economics* **58**, 2, 2002, 359-385.

Rime, D., Sarno, L., Sojli, E. Exchange rate forecasting, order flow and macroeconomic information. *Journal of International Economics* **80**, 1, 2010, 72-88.

Rosa, C.: The high-frequency response of exchange rates to monetary policy actions and statements. *Journal of Banking and Finance* **35**, 2, 2011, 478-489.

Saint-Paul, G.: A "quantized" approach to rational inattention. *European Economic Review* **100**, November, 2017, 50–71.

Sarno, L., Valente, G. Exchange Rates and Fundamentals: Footloose or Evolving Relationship? *Journal of the European Economic Association* **7**, 4, 2009, 786–830.

Shannon, C. E.: *A Mathematical Theory of Communication.* Bell Labs Technical Journal **27**, 3, 1948, 379-423.

Simon, H. A. Designing organizations for an information-rich world. In: Greenberger, M. (ed.): *Computers, Communications, and the Public Interest*. Johns Hopkins University, 1971, 37-72.

Sims, C. A.: Stickiness. *Carnegie-Rochester Conference Series on Public Policy* **49**, December, 1998, 317-356.

Sims, C. A.: Implications of rational inattention. *Journal of Monetary Economics* **50**, 3, 2003, 665–690.

Sims, C. A.: Rational Inattention: Beyond the Linear-Quadratic Case. *American Economic Review* **96**, 2, 2006, 158-163.

Sims, C. A.: Rational inattention and monetary economics. In: Friedman, B. M., Woodford, M. (eds.): *Handbook of Monetary Economics* **3**, 2010, 155-181.

Smith, G. P. Google Internet Search Activity and Volatility Prediction in the Market for Foreign Currency. *Finance Research Letters* **9**, 2, 2012, 103-110.

Suhoy, T. Query indices and a 2008 downturn: Israeli data. *Bank of Israel Discussion Paper* **06**, 2009.

Tashman, L. J.: Out-of-Sample Tests of Forecasting Accuracy: An Analysis and Review. *International Journal of Forecasting* **16**, 4, 2000, 437–450.

Taylor, M. P. The Economics of Exchange Rates. *Journal of Economic Literature* **33**, 1, 1995, 13-47.

Yuan, Ch. The Exchange Rate and Macroeconomic Determinants: Time-Varying Transitional Dynamics. *North American Journal of Economics and Finance*, **22**, 2, 2011, 197-220.

Zhang, C. An information-based theory of international currency. *Journal of International Economics* **93**, 2, 2014, 286-301.

Appendix

Table A1. Definition of Macroeconomic Variables

Name and Source	Definition
GDP OECD http://stats.ukdataservice.ac.uk/Index.aspx?DataSetCode=MEI	Gross domestic product at constant prices, value, seasonally adjusted, national currency for all countries and US Dollars (fixed PPPs) for Japan (Main economic indicators, October 2017).
CPI OECD http://stats.ukdataservice.ac.uk/Index.aspx?DataSetCode=MEI	Consumer price index, index publication base (Main economic indicators, October 2017).
Interest rate OECD http://stats.ukdataservice.ac.uk/Index.aspx?DataSetCode=MEI	3-month or 90-day rates and yields for all countries except for Japan (certificates of deposit), interbank rates in % (Main economic indicators, October 2017).
M1 OECD http://stats.ukdataservice.ac.uk/Index.aspx?DataSetCode=MEI Bank of England for the United Kingdom	Monetary aggregate M1, value, seasonally adjusted, national currency (Main economic indicators, October 2017).
Export, Import OECD http://stats.ukdataservice.ac.uk/Index.aspx?DataSetCode=MEI	Export and import, value (goods), total, seasonally adjusted, national currency (Main economic indicators, October 2017).

Table A2. Search conditions

Description	Search condition
All Papers about Japan without USA	"japan" NOT "us" OR "usa" OR "u.s." OR "united states""
All selected categories about Japan without USA	"japan" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category output about Japan without USA	"japan" AND "gpd" OR "output" OR "recession" OR "production"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category money about Japan without USA	"japan" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category price about Japan without USA	"japan" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category trade about Japan without USA	"japan" AND "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
All Papers about USA without Japan	"us" OR "usa" OR "united states"" NOT "japan"
All selected categories about USA without Japan	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "japan"
Category output about USA without Japan	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "japan"
Category money about USA without Japan	"us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "japan"
Category price about USA without Japan	"us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "japan"
Category trade about USA without Japan	"us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import"" NOT "japan"
All Papers about Japan and USA	"japan" AND "us" OR "usa" OR "united states""
All selected categories about Japan and USA	"japan" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import""
Category output about Japan and USA	"japan" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production""
Category money about Japan and USA	"japan" AND "us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank""
Category price about Japan and USA	"japan" AND "us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi""
Category trade about Japan and USA	"japan" AND "us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import""

All Papers about Euroarea without USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" "euro" OR "europe"" NOT "us" OR "usa" OR "u.s." OR "united states""
All selected categories about Euroarea without USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category output about Euroarea without USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category money about Euroarea without USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category price about Euroarea without USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category trade about Euroarea without USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
All Papers about USA without Euroarea	"us" OR "usa" OR "united states"" NOT "eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe""
All selected categories about USA without Euroarea	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe""
Category output about USA without Euroarea	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe""
Category money about USA without Euroarea	"us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe""
Category price about USA without Euroarea	"us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe""
Category trade about USA without Euroarea	"us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import"" NOT "eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe""

All Papers about Euroarea and USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "us" OR "usa" OR "united states""
All selected categories about Euroarea and USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import""
Category output about Euroarea and USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production""
Category money about Euroarea and USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank""
Category price about Euroarea and USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi""
Category trade about Euroarea and USA	"eurozone" OR "euroarea" OR "euro zone" OR "euro area" OR "europe"" AND "us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import""
All Papers about UK without USA	"UK" OR "britain" OR "england" OR "kingdom"" NOT "us" OR "usa" OR "u.s." OR "united states""
All selected categories about UK without USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category output about UK without USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category money about UK without USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category price about UK without USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category trade about UK without USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
All Papers about USA without UK	"us" OR "usa" OR "united states"" NOT "UK" OR "britain" OR "england" OR "kingdom""

All selected categories about USA without UK	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "UK" OR "britain" OR "england" OR "kingdom""
Category output about USA without UK	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "UK" OR "britain" OR "england" OR "kingdom""
Category money about USA without UK	"us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "UK" OR "britain" OR "england" OR "kingdom""
Category price about USA without UK	"us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "UK" OR "britain" OR "england" OR "kingdom""
Category trade about USA without UK	"us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import"" NOT "UK" OR "britain" OR "england" OR "kingdom""
All Papers about UK and USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "us" OR "usa" OR "united states""
All selected categories about UK and USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import""
Category output about UK and USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production""
Category money about UK and USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank""
Category price about UK and USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi""
Category trade about UK and USA	"UK" OR "britain" OR "england" OR "kingdom"" AND "us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import""
All Papers about Canada without USA	"canada" NOT "us" OR "usa" OR "u.s." OR "united states""
All selected categories about Canada without USA	"canada" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category output about Canada without USA	"canada" AND "gpd" OR "output" OR "recession" OR "production"" NOT "us" OR "usa" OR "u.s." OR "united states""

Category money about Canada without USA	"canada" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category price about Canada without USA	"canada" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category trade about Canada without USA	"canada" AND "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
All Papers about USA without Canada	"us" OR "usa" OR "united states"" NOT "canada"
All selected categories about USA without Canada	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "canada"
Category output about USA without Canada	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "canada"
Category money about USA without Canada	"us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "canada"
Category price about USA without Canada	"us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "canada"
Category trade about USA without Canada	"us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import"" NOT "canada"
All Papers about Canada and USA	"canada" AND "us" OR "usa" OR "united states""
All selected categories about Canada and USA	"canada" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import""
Category output about Canada and USA	"canada" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production""
Category money about Canada and USA	"canada" AND "us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank""
Category price about Canada and USA	"canada" AND "us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi""
Category trade about Canada and USA	"canada" AND "us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import""
All Papers about Australia without USA	"australia" NOT "us" OR "usa" OR "u.s." OR "united states""
All selected categories about Australia without USA	"australia" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category output about Australia without USA	"australia" AND "gpd" OR "output" OR "recession" OR "production"" NOT "us" OR "usa" OR "u.s." OR "united states""

Category money about Australia without USA	"australia" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category price about Australia without USA	"australia" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category trade about Australia without USA	"australia" AND "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
All Papers about USA without Australia	"us" OR "usa" OR "united states"" NOT "australia"
All selected categories about USA without Australia	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "australia"
Category output about USA without Australia	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "australia"
Category money about USA without Australia	"us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "australia"
Category price about USA without Australia	"us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "australia"
Category trade about USA without Australia	"us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import"" NOT "australia"
All Papers about Australia and USA	"australia" AND "us" OR "usa" OR "united states""
All selected categories about Australia and USA	"australia" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import""
Category output about Australia and USA	"australia" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production""
Category money about Australia and USA	"australia" AND "us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank""
Category price about Australia and USA	"australia" AND "us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi""
Category trade about Australia and USA	"australia" AND "us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import""
All Papers about New Zealand without USA	"zealand" NOT "us" OR "usa" OR "u.s." OR "united states""
All selected categories about New Zealand without USA	"zealand" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category output about New Zealand without USA	"zealand" AND "gpd" OR "output" OR "recession" OR "production"" NOT "us" OR "usa" OR "u.s." OR "united states""

Category money about New Zealand without USA	"zealand" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category price about New Zealand without USA	"zealand" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "us" OR "usa" OR "u.s." OR "united states""
Category trade about New Zealand without USA	"zealand" AND "trade" OR "export" OR "import"" NOT "us" OR "usa" OR "u.s." OR "united states""
All Papers about USA without New Zealand	"us" OR "usa" OR "united states"" NOT "zealand"
All selected categories about USA without New Zealand	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import"" NOT "zealand"
Category output about USA without New Zealand	"us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production"" NOT "zealand"
Category money about USA without New Zealand	"us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank"" NOT "zealand"
Category price about USA without New Zealand	"us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi"" NOT "zealand"
Category trade about USA without New Zealand	"us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import"" NOT "zealand"
All Papers about New Zealand and USA	"zealand" AND "us" OR "usa" OR "united states""
All selected categories about New Zealand and USA	"zealand" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production" OR "money" OR "interest rate" OR "monetary" OR "central bank" OR "price" OR "inflation" OR "deflation" OR "cpi" OR "trade" OR "export" OR "import""
Category output about New Zealand and USA	"zealand" AND "us" OR "usa" OR "united states"" AND "gpd" OR "output" OR "recession" OR "production""
Category money about New Zealand and USA	"zealand" AND "us" OR "usa" OR "united states"" AND "money" OR "interest rate" OR "monetary" OR "central bank""
Category price about New Zealand and USA	"zealand" AND "us" OR "usa" OR "united states"" AND "price" OR "inflation" OR "deflation" OR "cpi""
Category trade about New Zealand and USA	"zealand" AND "us" OR "usa" OR "united states"" AND "trade" OR "export" OR "import""

Can the Machine Learn Capital Structure?

Jack Strauss
University of Denver. USA

Abstract

We use machine-learning tools to analyze and predict capital structure using data from approximately 180,000 firms. LASSO, Random Forecast, Gradient Boosting, Neural net methods and a general linear model are used to select variables over a 1970-2000 in-sample and training period. We then construct out-of-sample mean squared forecast errors from 2001-2016. We show that Random Forest methods significantly, substantially and consistently outperform the benchmark linear model as well as a benchmark Lasso model. We then examine variable importance, and show that more than a dozen variables reliably predict corporate leverage over the past fifteen years. Our work highlights the importance of incorporating nonlinearities for predicting capital structure. And yes the machine can learn and predict capital structure.

Evaluating Auto-encoder and Principal Component Analysis for Feature Engineering in Electronic Health Records

Shruti Kaushik[1,a], Abhinav Choudhury[1,b], Nataraj Dasgupta[2,c], Sayee Natarajan[2,d], Larry A. Pickett[2,e], and Varun Dutt[1,f]

[1]Applied Cognitive Science Laboratory, Indian Institute of Technology Mandi, Himachal Pradesh, India – 175005
[2]RxDataScience, Inc., USA - 27709

[a]shruti_kaushik@students.iitmandi.ac.in, [b]abhinav_choudhury@students.iitmandi.ac.in, [c]nd@rxdatascience.com, [d]sayee@rxdatascience.com, [e]larry@rxdatascience.com, and [f]varun@iitmandi.ac.in

Abstract. Feature engineering is an important mechanism where we transform and represent the high-dimensional data into a lower-dimensional space. These representations can then be used to efficiently train machine-learning models. Auto-encoders are widely used in research for unsupervised feature learning. However, the application of auto-encoders for electronic health records (EHRs) containing features with binary values (binary-valued features) has been less studied. The primary objective of this research was to compare an auto-encoder with principal component analysis (PCA), a popular feature engineering technique, for feature selection in different (US and Indian) EHR datasets containing binary-valued features. The US dataset contained thousands of binary-valued features, and the Indian dataset contained nineteen binary-valued features. Results revealed that feature selection by the auto-encoder followed by different classification algorithms gave the highest accuracy on both the datasets compared to feature selection by PCA. We highlight the implications of using auto-encoders for learning features in EHR datasets.

Keywords: Auto-encoders, principal component analysis, dimensionality reduction, machine learning, classification, EHR, features.

1 Introduction

The use of electronic health records (EHRs) has increased among hospitals, clinics, and patients' care settings [1]. These records may store patients' visit information, demographic details, diagnoses, lab test results, and prescription information [2, 16]. In general, EHR datasets contain several features which may help accurately predict healthcare outcomes.

Prior research has proposed several machine learning (ML) algorithms for predicting different healthcare outcomes [10]. The accurate prediction of healthcare outcomes, however, may require selection of relevant features (i.e., feature engineering) in data. In fact, keeping irrelevant and redundant features could mislead ML algorithms [5]. Thus, one may need to perform feature engineering before implementing ML algorithms [5].

Some of the feature engineering techniques include filters, wrappers, and embedded methods, where these methods help reduce the number of features in data [5]. However, beyond these techniques, there exist other techniques that may create new features from the original features present in datasets [4]. For example, principal component analysis (PCA) is a feature engineering technique that performs a linear combination of original features to create a new set of features in lower dimensional feature space [15]. Similarly, auto-encoders, a neural network with the same inputs and outputs, is another feature engineering technique where new features are a non-linear combination of original features [3]. Prior research has used different auto-encoder-based approaches for features selection on image and magnetic resonance image datasets [6-8]. Research has also compared auto-encoder-based and PCA-based feature engineering techniques for classifying neuro-images [6]. The main advantage of PCA and auto-encoder techniques over the filters, wrappers, and embedded methods is that the former allows all original features to contribute to the transformed features; whereas, the latter approaches may eliminate some of the original features from datasets.

In the real-world, EHR datasets may contain several features with binary absent/present values (i.e., binary-valued features) [11]. For example, EHRs may contain diagnosis codes, procedure codes, and other demographic variables as absent or present features corresponding to patients [11]. In fact, features in EHR datasets could be converted into binary-valued features, where rows are linked to unique patients and columns are the different binary (absent/present) features. Although auto-encoder-based and PCA-based feature engineering techniques have been used in healthcare-related image analyses; however, to the best of authors' knowledge, a comparison of these techniques on EHRs containing several binary-valued features has not been explored in literature. In this research, we address this literature gap by evaluating auto-encoder-based and PCA-based feature-engineering approaches on real-world EHR datasets. Here, we compare different feature-engineering techniques by evaluating their ability to classify records post feature engineering.

The primary objective of this paper is to evaluate PCA and auto-encoders in their feature engineering capabilities across two EHR datasets containing binary-valued features. One dataset involves the purchase of two pain medications in the US, and the other dataset involves the purchase of five general-purpose medications in a large district hospital in Himachal Pradesh, India. We perform classification of records post feature engineering by relying upon three standard ML algorithms including, naive Bayes classifier [12], logistic regression [13], and support vector machine (SVM) [14].

In what follows, we first provide a brief review of related literature. Next, we explain the methodology of applying various feature engineering techniques. In Section IV, we present our experimental results and compare classification accuracies post feature engineering using PCA and auto-encoders. Finally, we discuss our results and conclude our paper by highlighting the main implications of this research and its future scope.

2 Background

Prior research has evaluated PCA as a feature engineering technique in image retrieval tasks [4]. For example, reference [4] compared PCA and linear discriminant analysis (LDA) in content-based image retrieval task and found PCA to perform better compared to LDA. Also, researchers have compared PCA with other feature engineering techniques like the gain ratio, fuzzy rough features (FRF) selection, and correlation-based feature selection in a breast cancer dataset [18]. Results show that the FRF approach outperformed PCA in terms of better classification accuracy [18].

Prior research has also evaluated auto-encoders as a feature engineering technique involving magnetic resonance images [6] and image datasets [7]. For example, reference [6] used stacked auto-encoders for feature engineering and compared auto-encoders with the LASSO-based methods, PCA, and two-sample t-test approaches for prediction of Alzheimer's disease from magnetic resonance images. Results revealed better performance with auto-encoders compared to PCA [6]. Similarly, reference [7] has used different variants of auto-encoders for feature engineering on popular image datasets [7]. Moreover, researchers have also used various ML techniques like decision tree, naive Bayes, SVM, neural networks, and regression approach for performing classification post feature engineering [6, 11].

Although a number of image-based applications have utilized PCA-based and auto-encoder-based feature engineering methods, an evaluation of these feature engineering approaches has not been done on EHR datasets containing several binary-valued features. Overall, on EHR datasets, we expect auto-encoders to perform better compared to PCA in feature engineering and post classification. This expectation is based upon prior literature presented above as well as the fact that auto-encoders are nonlinear feature engineering techniques compared to PCA, which is a linear feature engineering technique [6].

3 Method

3.1 Data

We used two datasets for feature engineering and subsequent classification. The first dataset (I) was the Truven MarketScan® health dataset[1] containing patients' insurance claims in the US [16]. This dataset contained approximately 45,000 unique patients. Between January 2011 and December 2015, these patients formed the following consumer groups across two common pain medications: consumers of medicine A (55.2% patients), consumers of medicine B 39.98% patients), and consumers of both medicine A and B (4.82% patients).[2] The dataset contains patients' demographic variables (age, gender, region, and birth year), clinical variables (admission type, diagnoses made, and procedures performed), the name of medicines, and medicines' refill counts per patient. There were a total of 15,081 features (including class) present against each patient in the Truven dataset. Out of these features, 15,075 features were present/absent binary-valued diagnoses and procedure codes. The list of other 6 (non-

[1] Truven Market scan dataset links paid claims and detailed patient information over time.

[2] Due to the non-disclosure agreement, we have anonymized the original names of these medications.

binary-valued) features is shown in Table 1. Five out of 6 features contained demographic information and the last feature contained the class label.

We first separated the 15,075 features (diagnoses and procedure codes) from the demographic features (listed in Table 1). We then applied PCA and auto-encoder on 15,075 diagnoses and procedure codes to select the relevant features and then combined the selected features with the other 6 non-binary-valued demographic features. From 15,075 binary features, we transformed the number of features in the following sequence: 1000, 5000, and 10000. We varied the number of features to investigate whether the classification accuracy changed as one changed the number of features. These new features, along with the 6 features (mentioned in Table 1) were then used to classify patients into three consumer classes, as discussed above.

Table 1. Description of Input Features for Dataset (I)

Features	Description
Gender	Male, Female
Age-group	0-17, 18-34, 35-44, 45-54, 55-64
Region	Northeast, northcentral, south, west, unknown
Type of admission	Surgical, medical, maternity and newborn, psych and substance abuse, unknown
Refill count	Count in number
Pain medication (Class)	A, B, Both

The second dataset (II) was collected from a government hospital in Mandi district, Himachal Pradesh, India. This dataset contained five general-purpose medications, which were the top-most five medications prescribed by the doctors in this hospital. The dataset contained approximately 30,000 unique patients who consumed five medications (A' to E') between June 2016 and January 2018. The dataset contains 20% records of patients who consumed A', 16.4% records of patients who consumed B', 19.2% records of patients who consumed C', 18.5% records of patients who consumed D', and 25.9% records of patients who consumed E'. There were a total of 21 features in this dataset, including the class label (A' to E' corresponding to different medications). Table 2 shows the description of these 21 features. Out of the 21 features, the first 19 features were binary-valued (see Table 2). Overall, both datasets contained binary-valued features across a majority of their attributes with a similar number of unique patients.

We performed feature engineering only on the first 19 binary-valued features to select the relevant features. From these 19 binary-valued features, we transformed the number of features in the following sequence: 5, 10, and 15. Then, we combined the transformed features with the Quantity feature and the Class label (the last two attributes in Table 2) to classify the patients according to the medication they consumed (A' to E').

Table 2. Description of Input Features for Dataset (II)

Features	Description
Age-group (0-18)	Contains binary value (0/1)
Age-group (19-39)	Contains binary value (0/1)
Age-group (40-59)	Contains binary value (0/1)

Age-group (60+)	Contains binary value (0/1)
Male OPD	Contains binary value (0/1)
Female OPD	Contains binary value (0/1)
Medicine OPD	Contains binary value (0/1)
Skin OPD	Contains binary value (0/1)
Eye OPD	Contains binary value (0/1)
ENT OPD	Contains binary value (0/1)
Surgical OPD	Contains binary value (0/1)
Orthopedic OPD	Contains binary value (0/1)
Dental OPD	Contains binary value (0/1)
Gyne OPD	Contains binary value (0/1)
Psychiatry OPD	Contains binary value (0/1)
Skin OPD	Contains binary value (0/1)
Pediatrics OPD	Contains binary value (0/1)
Emergency OPD	Contains binary value (0/1)
Pulmonary Medicine	Contains binary value (0/1)
Quantity	Total number of capsules
Class	Name of medicine

Note: 1 indicates that the patient belongs to a specific feature. OPD means Out Patient Department.

3.2 Principal Component Analysis (PCA)

PCA is a feature engineering technique that does not directly select features as present in the dataset [15]. However, PCA aims to reduce the dimensionality of a dataset containing several correlated features by transforming the original feature space into a new feature space in which all the features are uncorrelated [15]. PCA finds the principal components in data, where these components are the directions where the data is most spread out or the directions with the most variance in data. The process of finding the principal components is described below:

1. Take the complete dataset of $d + 1$ dimensions and discard the class attribute such that our dataset becomes d dimensional.
2. Calculate the mean for each dimension of the dataset.
3. Calculate the covariance matrix of the whole dataset.
4. Calculate the eigenvectors and the corresponding eigenvalues.
5. Sort the eigenvectors by decreasing eigenvalues and choose p eigenvectors with the largest eigenvalues to form a $d \times p$ dimensional matrix W.
6. Use this $d \times p$ eigenvectors matrix to transform the original features onto the new subspace.

Thus, implementing PCA means finding the eigenvalues and eigenvectors of the features' correlation matrix in data [15]. Eigenvectors and eigenvalues exist in pairs. Eigenvector gives the direction, and corresponding eigenvalues (which is a number) tells how much variance is present in the data in that direction. For dataset (I), we selected the top-most 1000, 5000, and 10000 eigenvectors and transformed the features in the direction of these eigenvectors. After this step, the new features were combined with the 6 other features (listed in Table 1) for the classification task. While on dataset (II), we selected the top-most 5, 10, and 15 eigenvectors and transformed whole data in the direction of these eigenvectors. The new (transformed) features were then combined with quantity and class features (listed in Table 2) to perform the classification of medications consumed.

3.3 Auto-encoder

An auto-encoder is an unsupervised machine learning technique that can learn representations from data [3]. Auto-encoders work by compressing the input into a latent-space representation (lower dimensional space) and then reconstructing the input from this representation.

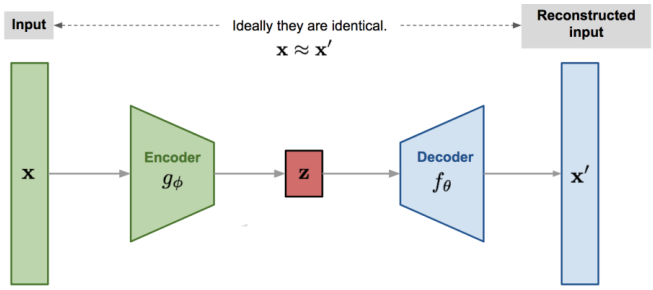


Fig. 1. The architecture of an auto-encoder [3]

In an auto-encoder, we take an unlabeled dataset and frame it as a supervised learning problem tasked with outputting x', a reconstruction of the original input x (Fig. 1). The auto-encoder consists of two parts: encoder and decoder. Given the unlabeled input dataset $\{x_n\}_{n=1}^{N}$, the encoder maps input $x \in R^d$ to $z \in R^p$ *where* $p < d$, where d are the total number of features in data and p are the reduced number of features in the latent space. The encoding process is defined as follows:

$$z_n = g_\Phi(W_1 x_n + b_1) \qquad (1)$$

Where g_Φ is the encoding function, W_1 is the weight matrix of the encoder, b_1 is the bias vector, and z_n is known as the latent representation. Once the input has been encoded, the decoder tries to reconstruct the input x form the latent representation z_n and maps it to the output $x' \in R^d$. The decoder process is defined as follows:

$$x'_n = f_\theta\,(W_2 z_n + b_2) \qquad (2)$$

Where f_θ is the decoding function, W_2 is the weight matrix of the decoder, and b_2 is the bias vector. This network is then trained by minimizing the reconstruction error, $L(x, x')$, which measures the differences between our original input and the consequent reconstruction.

$$L(x, x') = \| \mathrm{X} - \mathrm{X}' \|^2 \qquad (3)$$

The stacked auto-encoder consists of multiple layers of nodes in which the outputs of each layer are wired to the inputs of the successive layer (Fig. 2).

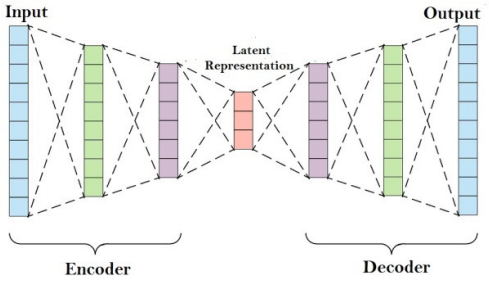


Fig. 2. The architecture of a stacked auto-encoder [3]

Given dataset (I) $\{x_n\}_{n=1}^{45000}$, the encoder maps input $x \in R^{15075}$ to $z \in R^p$ *where p is set to* $1000, 5000,$ and 10000. In case of dataset (II) $\{x_n\}_{n=1}^{30000}$, the encoder maps input $x \in R^{19}$ *to* $z \in R^p$ *the p is set to* $5, 10,$ and 15. Once the auto-encoder has been trained, we save the latent representation and combine it with the other features to perform classification. For training the auto-encoder on both datasets, we tried different batch sizes and finally used a batch size of 64. Furthermore, 90% of data was used for training, and the remaining 10% data was used for testing. The auto-encoder was trained for 50 epochs on both datasets to obtain the encoded dimensions (new features). We used Adadelta as an optimizer and mean square error as the optimizer function [19]. Table 3 shows the architecture of the stacked auto-encoders used to obtain a different set of features. These architectures were selected after a trial-and-error evaluation of the test loss from different auto-encoder architectures (the objective is to minimize the test loss).

On dataset (I), the 1000 encoded dimensions were achieved with 15075 neurons in the input layer, 8000, 4000, 2000, and 1000 neurons in the encoder layers, and 2000, 4000, 8000, and 15075 neurons in the decoder layers. The 5000 encoded dimensions were achieved with 15075 neurons in the input layer, 11000, 7000, and 5000 neurons in the encoded layers, and 7000, 11000, 15075 neurons in the decoded layers. Similarly, the 10000 encoded dimensions were achieved with 15075 neurons in the input layer, 13000, 10000 neurons in the encoded layers, and 13000, 15075 neurons in the decoded layers.

On dataset (II), the 5 encoded dimensions were achieved with 19 neurons in the input layer, 16, 12, 8, and 5 neurons in the encoder layers, and 8, 12, 16, and 19 neurons in the decoder layers. The 10 encoded dimensions were achieved with 19 neurons in the input layer, 16, 13, and 10 neurons in the encoder layers, and 13, 16, and 19 neurons in the decoder layers. Similarly, the 15 encoded dimensions were achieved with 19 neurons in the input layer, 17, 15 neurons in the encoder layers, and 17, 19 neurons in the decoder layers.

Table 3. Description of Stacked Auto-encoder

Dataset	Total features (binary-valued)	Encoding dimension	Total number of encoder and decoder layers	Test Loss
I	15075	1000	8 (4 encoder, 4 decoder)	0.010
		5000	6 (3 encoder, 3 decoder)	0.002
		10000	4 (2 encoder, 2 decoder)	0.009
II	19	5	8 (4 encoder, 4 decoder)	0.030
		10	6 (3 encoder, 3 decoder)	0.018
		15	4 (2 encoder, 2 decoder)	0.008

3.4 Classification Algorithms

We combined the new transformed features from PCA and auto-encoders with other features in dataset I and II (see Tables 1 and 2). Both datasets were then divided into two parts for classification using naive Bayes, logistic regression and SVM: 70% of the data was used for training, and 30% of the data was used for testing.

3.4.1 Naive Bayes

Naive Bayes is a probabilistic classifier that is based on the Bayes theorem. It is called naive because it assumes a strong independence assumption between features [12]. It assumes that the value of a specific feature is independent of the value of any other feature, given the target (class) label. Despite this assumption, naive Bayes has been quite successful in solving practical problems in text classification, medical diagnosis and system performance management [12]. The classifier attempts to maximize the posterior probability in determining the class of a transaction.

Suppose, vector y = $(y_1, y_2, \ldots, y_n)$ represent the features in the problem with n denoting the total number of features and k be the possible number of classes C_k. Naive Bayes is a conditional probability model which can be decomposed as [12]:

$$p(C_k/y) = \frac{p(C_k)\, p(y/C_k)}{p(y)} \quad (4)$$

Under the independence assumption, the probabilities of the features are defined as follows [20]:

$$p(C_k/y_1, \ldots, y_n) = \; p(C_k) \prod_{i=1}^{n} p(y_i/C_k) \quad (5)$$

This most likely class is then picked based on the maximum a posteriori (MAP) decision rule [20] as follows:

$$C_k = argmax_{k\in 1\ldots K}\; p(C_k) \prod_{i=1}^{n} p(y_i/C_k) \quad (6)$$

3.4.2 Logistic Regression

Logistic regression is a linear classifier which can be used for modeling the relationship between one dependent binary variable (Y) and one or more independent variables (X) [13]. It models the posterior probabilities of the 2 classes in an instance. Let p represents the probability of occurrence of a class event ($p = P(Y = y)$, where Y is the class label possessing a value y), which depends on independent variables ($X_1, X_2, \ldots X_n$). We use the following equation for modeling the probability:

$$p = \frac{e^{a_0 + a_1X_1 + a_2X_2 + \cdots + a_nX_n}}{1 + e^{a_0 + a_1X_1 + a_2X_2 + \cdots + a_nX_n}} \quad (7)$$

Where a_0 is the bias or intercept term, and $a_1, a_2, \ldots, a_n$ are the coefficients for the independent variables ($X_1, X_2, \ldots X_n$). Since there are 3 classes in dataset (I) and 5 classes in dataset (II), we performed one-vs-all classification using logistic regression. One-vs-all classification was implemented by training multiple logistic regression classifiers, one for each of the K classes in the training dataset. Hence, we trained 3

different logistic regression classifiers for dataset (I) and 5 different logistic regression classifiers for dataset (II). Once the one-vs-all predictions had been made for all classes, the classifier picked the class with the highest probability.

3.4.3 Support Vector Machines

Support vector machines (SVM) are supervised classification techniques which can handle large features space [14]. SVMs are the binary classifiers which can be utilized for multi-class classification tasks as well. They build a hyperplane or a set of hyperplanes in a high dimensional space which can be used for classification and regression-based tasks. SVMs can classify linearly separable as well as non-linearly separable data [14]. If the data is linearly separable, then SVM uses the linear hyperplane to perform classification. However, for the non-linear data, rather than fitting a non-linear curve, it transforms the data into high dimensional space to perform classification. SVM uses the kernel functions, e.g., radial basis function (RBF kernel) to transform data into the high dimensional plane for classifying the non-linear data [14]. SVM uses gamma and C parameters to perform classification. Gamma defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. The C parameter defines the cost of misclassification. A large C gives low bias and high variance; whereas, a small C gives higher bias and low variance. In this paper, we used SVM with the RBF kernel to classify the patients. We chose gamma as 1/number of features while implementing SVM. We chose C = 1 in the SVM across both auto-encoder and PCA. The SVM also performed one-vs-all classification for classifying the three classes on dataset (I) and five classes on dataset (II) (this process was similar to the one followed in logistic regression).

3.4.4 Random Chance Classification

We also ran the Monte Carlo [17] simulations 5000 times to generate a random guess for each of the classes on both datasets. Since all the classes are not equally likely, specifically in the case of dataset (I). Therefore, while running the Monte Carlo simulations, we kept the same probability for each class as present in the actual datasets. This was done to check if the classification algorithms gave better accuracy than a random guess and by how much percentage the feature engineering techniques helped in improving the accuracy.

4 Results

Fig. 3 and Fig. 4 show the accuracy from different classifiers on test data with all features and with features from feature engineering using auto-encoder (AE) and PCA (LR, SVM, NB, AVG, and RC refer to the logistic regression, support vector machine, naive Bayes, average accuracy across all classifiers, and the random chance, respectively). On dataset (I), the best accuracy (= 63.01%) was obtained with 10005 features on test data when these features were selected by the stacked auto-encoder

and classified by logistic regression. On dataset (II), the best accuracy (= 63.08%) was obtained with 16 features on test data when these features were selected by the stacked auto-encoder and classified by SVM. On dataset (I), decreasing features from 15080 to 1005 (a 93% decrease) decreased the average accuracy from 55.11% to 52.49% in the worst case (a meagre 2.62% decrease). On dataset (II), decreasing features from 20 to 6 (a 70% decrease) decreased the average accuracy from 56.42% to 47.91% in the worst case (a meagre 8.51% decrease). Among all classifiers, the naive Bayes algorithm was most affected by the decrease in the number of features in both datasets. Furthermore, the average accuracy of 5000 runs from the random chance algorithm (Monte Carlo) came out to be 46.5% and 20% on dataset (I) and (II), respectively. Thus, both LR and SVM algorithms performed better compared to the random chance algorithm (Monte Carlo). Overall, we witnessed only a small reduction in the average accuracy due to feature reduction.

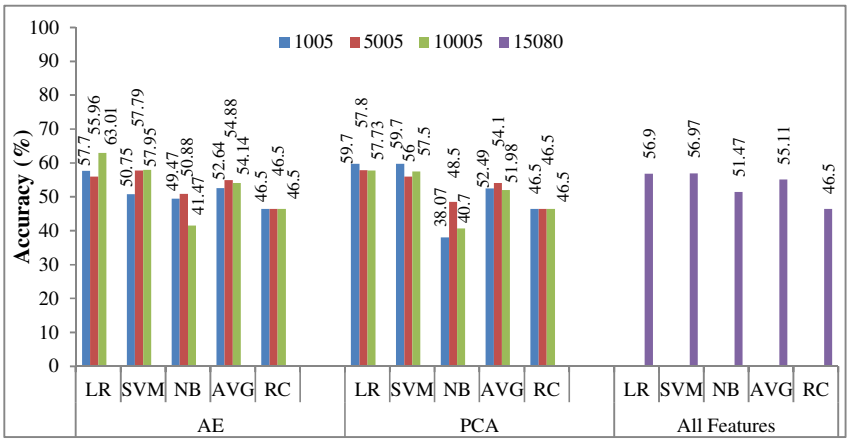


Fig. 3. Test classification accuracy on dataset (I)

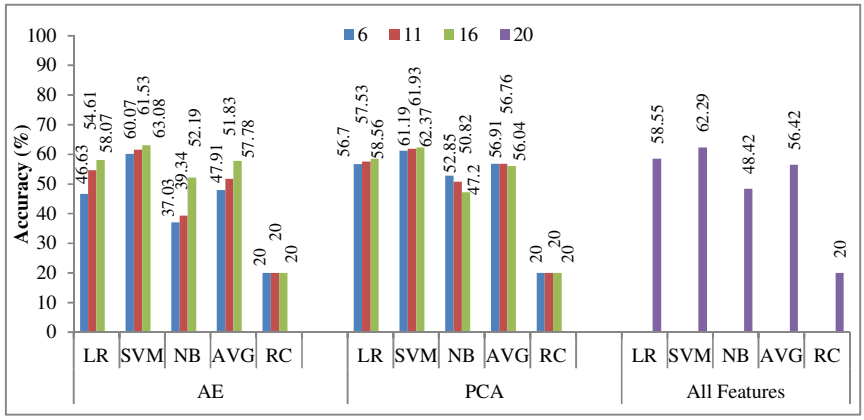


Fig. 4. Test classification accuracy on dataset (II)

5 Discussion and Conclusions

Feature engineering may be needed in EHR datasets with binary-valued features when the number of such features is large as feature engineering may likely help reduce the complexity of machine-learning algorithms [5]. The primary objective of this research was to evaluate two popular features engineering techniques (PCA and auto-encoders) for classifying patients according to their medicine consumption across two EHR datasets involving several binary-valued features.

First, feature engineering using auto-encoders gave better accuracies compared to feature engineering using PCA. A likely reason for this finding is that neural networks are capable of learning non-linear relationships from data compared to PCA, which is a linear feature selection technique [6]. Perhaps, the ability to learn non-linear relationships led-to better feature engineering from auto-encoders compared to PCA.

Second, we found that the naive Bayes classifier's accuracy was most affected by feature engineering. A likely reason for this finding is that the naive Bayes algorithm treats all the features independently and gives them equal importance. Thus, decreasing the number of features dents the accuracy of this algorithm compared to other algorithms that may not treat all features with equal weights.

Third, there was only a meagre decrease in the average accuracy across classifiers after feature engineering on binary-valued attributes. Overall, this result is promising, and it shows that feature engineering on large EHR datasets with binary-valued features is an ecologically valid exercise. Furthermore, there were significant improvements across LR and SVM algorithms compared to the Monte Carlo simulations in both datasets. Again, these results show that classification using LR and SVM approaches seems to be effective across both linear and non-linear feature-engineering methods.

Finally, there are some other feature-engineering approaches, such as XGBoost [20] and Bayesian belief networks [21]. Thus, as part of our future work, we plan to extend our current investigation to these approaches on binary-valued EHR datasets.

Acknowledgement. This project was supported by grants (awards: #IITM/CONS/RxDSI/VD/16 and # IITM/CONS/PPLP/VD/03) to Varun Dutt.

References

1. J. E and Y. N.: Electronic Health Record Adoption and Use among Office-based Physicians in the U.S., by State: 2015 National Electronic Health Records Survey. The Office of the National Coordinator for Health Information Technology, Tech. Rep. (2016).
2. P. B. Jensen, L. J. Jensen, and S. Brunak: Translational genetics: Mining electronic health records: towards better research applications and clinical care. Nature Reviews - Genetics, Vol. 13, pp. 395–405 (2012).
3. I. Goodfellow, Y. Bengio, and A. Courville: Deep learning. MIT Press, (2016).
4. Shereena, V. B., & David, J. M.: COMPARATIVE STUDY OF DIMENSIONALITY REDUCTION TECHNIQUES USING PCA AND LDA FOR CONTENT BASED IMAGE RETRIEVAL. Computer Science & Information Technology, pp. 41 (2015).

5. Motoda, H. and Liu, H.: Feature selection, extraction and construction. Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol, 5, pp.67-72 (2002).
6. Shi, S., & Nathoo, F.: Feature Learning and Classification in Neuroimaging: Predicting Cognitive Impairment from Magnetic Resonance Imaging. In 2018 4th International Conference on Big Data and Information Analytics (BigDIA), IEEE, pp. 1-5 (2018).
7. Meng, Q., Catchpoole, D., Skillicom, D., & Kennedy, P. J.: Relational autoencoder for feature extraction. In 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 364-371 (2017).
8. Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P.: Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. IEEE journal of biomedical and health informatics, 22(5), pp. 1589-1604 (2018).
9. Kaushik, S., Choudhury A., Mallik K., Moid A., and Dutt V.: Applying Data Mining to Healthcare: A Study of Social Network of Physicians and Patient Journeys. In Machine Learning and Data Mining in Pattern Recognition. Springer International Publishing, New York, pp. 599-613. (2016).
10. Sharma, R., Singh, S.N. and Khatri, S.: Medical data mining using different classification and clustering techniques: a critical survey. In Computational Intelligence & Communication Technology (CICT), Second International Conference on IEEE, pp. 687-691 (2016).
11. Kaushik, S., Choudhury, A., Dasgupta, N., Natarajan, S., Pickett, L. A., & Dutt, V.: Evaluating Frequent-Set Mining Approaches in Machine-Learning Problems with Several Attributes: A Case Study in Healthcare. In International Conference on Machine Learning and Data Mining in Pattern Recognition, Springer, pp. 244-258 (2018).
12. Langley, P. and Sage, S.: Induction of selective Bayesian classifiers. In Proceedings of the Tenth international conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc., pp. 399-406 (1994).
13. Peng, C.Y.J., Lee, K.L. and Ingersoll, G.M.: An introduction to logistic regression analysis and reporting. The journal of educational research, 96(1), pp.3-14 (2002).
14. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J. and Scholkopf, B.: Support vector machines. IEEE Intelligent Systems and their applications, 13(4), pp.18-28 (1998).
15. Song, F., Guo, Z. and Mei, D.: Feature selection using principal component analysis. In System science, engineering design and manufacturing informatization (ICSEM), international conference on IEEE, Vol. 1, pp. 27-30 (2010).
16. Danielson, E.: Health research data for the real world: the MarketScan® Databases. Ann Arbor, MI: Truven Health Analytics (2014).
17. Doubilet, P., Begg, C. B., Weinstein, M. C., Braun, P., & McNeil, B. J.: Probabilistic sensitivity analysis using Monte Carlo simulation: a practical approach. Medical decision making, 5(2), pp. 157-177 (1985).
18. El-Hasnony, I. M., El Bakry, H. M., & Saleh, A. A.: Comparative study among data reduction techniques over classification accuracy. International Journal of Computer Applications, 122(2) (2015).
19. Zhang, N., Lei, D., & Zhao, J. F.: An Improved Adagrad Gradient Descent Optimization Algorithm. In 2018 Chinese Automation Congress (CAC), IEEE, pp. 2359-2362 (2018).
20. Zheng, H., Yuan, J., & Chen, L.: Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. Energies. 10(8), 1168 (2017).
21. Cheng, J., & Greiner, R.: Learning bayesian belief network classifiers: Algorithms and system. In Conference of the Canadian Society for Computational Studies of Intelligence. Springer, Berlin, Heidelberg, pp. 141-151 (2001).

Improving The Management of Public Transport Through Modeling and Forecasting Passenger Occupancy Rate

Túlio Vieira, Paulo Almeida, Magali Meireles, and Renato Ribeiro

Intelligent Systems Laboratory, CEFET-MG, BRAZIL
Institute of Mathematical Sciences and Informatics, PUC-Minas, BRAZIL
Transport Engineering Department, CEFET-MG, BRAZIL
{tuphfv@lsi.cefetmg.br
pema@lsi.cefetmg.br
,magali@pucminas.br
renato@transporte.eng.br

Abstract. The improvement of public transport in large cities is a fundamental factor for the quality of life. Poor transportation leads to an increased of greenhouse gases generation, hinders access to essential services and emphasizes the difference between social classes. One possible way to improve traffic in large cities is to encourage people to use public transport. By improving the quality of public transport systems and reducing tariffs, more people can use it as a means of getting around in urban centers. This article performs an analysis between different strategies (Neural Recurrent Network using LSTM and GRU, Convolutional Neural Network and ARIMA models) to model the variation of the occupancy rate (PTO) of the metropolitan buses in order to improve the planning and management of public transport. Results show that ARIMA models present better results to PTO forecasting and to describe the behavior of time series. This kind of approach can be used, in practice, to adjust the number of buses and population demand, for a given period.

Keywords: Recurrent Neural Networks, Convolutional Neural Networks, ARIMA Models, Time Series Forecasting

1 Introduction

Many large cities around the World suffer from the problems generated by poor quality urban transport systems. Poor quality transportation systems directly affect the quality of life of an entire urban population [1]. A failing urban transport system causes an increase in greenhouse gases, an increase in the risk of accidents, hinders access to essential services (health, education and leisure) [2] and disrupts the development of economic activities [3].

According to [4], projections indicate that by 2030, there will be more than 2 billion motor vehicles on the streets. Most of these vehicles will be in large

cities of developing countries. This massive use of private vehicles as a means of urban transport worsens the conditions of transport and the problems faced by large cities. Figure 1 shows a comparison of urban growth and several other indicators for the city of Belo Horizonte, capital of the state of Minas Gerais in Brazil, between 2002 and 2012.

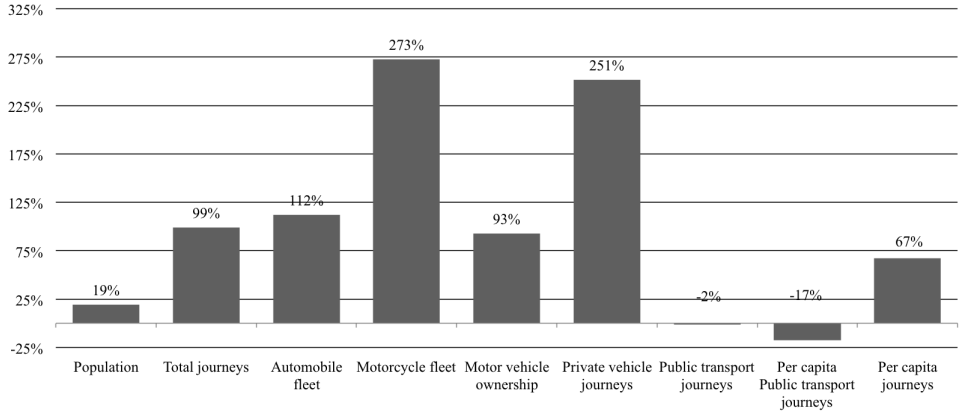


Fig. 1: Population growth and other mobility indicators of Belo Horizonte, Brazil, between 2002 and 2012 [5]

An alternative way to improve transport quality to and ensure sustainable development is to prioritize the use of public transport and to integrate this with other modes of transport [6]. The use of public transport for long-distance travel in cities contributes to improving people's quality of life by reducing air pollution, decreasing congestion, increasing mobility and access to essential services, and promoting social equality [7]. As can be seen in Figure 1, the increase in the population and number of cars in 10-year period was not accompanied by an increased use of public transport. This scenario worsens the problems faced by large cities in developing countries.

According to research carried out by the National Association of Urban Transportation Companies (NTU), in several capitals of Brazil in 2017, the number of passengers using public transport has been decreasing in recent years. This is the opposite of what happens in more developed countries. This fact can be explained by the poor quality of the service provided, the high price of bus tickets and the increase in the average time spent to reach destinations, when compared to the time spent on private transportation. Also, the increase in the number of cars on streets means that travel times increase even more for everyone. Public policies aimed at improving the quality of public transport are very important to make this mode of transport more attractive, especially in emerging countries.

In order to reverse this scenario and to promote the use of public transport, it is necessary to improve the quality of the urban transport service offered

to the population. In turn, to improve the quality of public transport service, it is necessary to improve planning and management. Through planning and management of the public transport system, it is possible to identify main routes of travel, to reduce travel and waiting times of passengers, to make more efficient use of vehicles fleet and to reduce overcrowding the vehicles. With better quality of service, residents of urban areas will tend to use more public transport.

Planning and management of public transport can be carried out by collecting and analyzing urban population data. For efficient planning, quality data are required on the behavior of daily population movements. As movements within urban centers change over the years, it is necessary to know the behavior of these movements so that good predictions about the dynamics of urban travel can be made. Therefore, predicting demand and population movements is fundamental for the efficient planning of public transport.

One way to predict the behavior of the daily displacements of a population is to estimate an Origin-Destination (OD) matrices. After building this matrix, it is possible to map which are the main attraction zones and trip distribution at the urban area. Another important tool to understand the dynamics of displacements is the public transport occupancy (PTO) rate. PTO measures the number of people inside a bus during a full trip. By using PTO, public transport operators can reduce the costs of the offered service, plan travel schedules, and avoid bus overcrowding. On the other hand, to collect conventional PTO data is an expensive and very time consuming task.

In this sense, to predict PTO in urban buses in an alternative way can be very useful to improve planning and management of public transport. This article aims to compare different techniques used to model the variation of PTO in urban buses during a complete trip, starting from digital data collection. We compare different computational intelligence techniques used as estimators of the temporal variation of PTO during a trip: Recurrent Neural Networks (RNN) using Long-Short Term Memory (LSTM), Gated Recurrent Unit (GRU) and Convolutional Neural Networks (CNN). In practical experiments, three different bus lines from the city of Belo Horizonte, in the state of Minas Gerais, Brazil, were used as test-beds to the proposed approaches.

The remainder of this article is organized as follows. Section 2 presents some work related to urban transport modeling. Section 3 discusses database collection and used techniques. Results achieved, statistical analysis and discussions are shown in Section 4. Section 5 draws some conclusion about this work.

2 Related Work

The search for models that can be used as predictors for urban transport demand is a recurring theme in the literature. Several researchers have been trying to generate efficient models to help in the improvement of transport systems and, thus, to improve the life of inhabitants of big cities. In this section, we will present some approaches that seek to model the displacement behavior of urban populations.

One way to improve urban transportation is to find techniques that can be used to model urban vehicular traffic. With such approaches, it is possible to reduce congestion and air pollution. To develop urban traffic modeling solutions, on these cases, is very common [8], [9], [10].

The author of [8] uses urban traffic modeling based on Hybrid Petri Net to reduce traffic jam by controlling traffic lights cycles structure and duration. [9] uses a cellular automata approach to model urban traffic. The results show that different mechanisms of traffic control have a significant bearing on traffic dynamics and inter vehicle spacing distribution. [10] uses a hybrid model that combines Artificial Neural Networks and a statistical strategy to provide one hour forecast of urban traffic flow rates. The reported results show that this approach is promising for predicting vehicle flow.

Another way that leads to a construction model that can be used for better traffic planning in large cities is to model and to predict OD matrices. These models are often used to model the dynamics of movements and to generate a more adequate planning of traffic in big cities. Some approaches that develop models to estimate OD matrices are [11–13].

[11] suggests the use of Markov models to estimate OD matrix. The results show that the adopted strategy can be used in practical applications for urban transport planning. [12] performs a comparison between different methods used to construct OD matrices. Also, he compares traditional survey methods and gravity models with strategies that use smartphone data to estimate those matrices. [13] proposes a model to generate OD matrices that does not use transfers and identification of passengers. They use only boarding and unboarding counts to estimate places of origin and destination of passengers.

There are few approaches to estimate PTO in the technical literature. Some of them focus on the elaboration of methods to automatically find PTO and, thus, to provide data for public transport planning. Examples are [14], [15] and [16]. [14] proposes a system that allows estimating passengers' flow, based on mobile crowd-sensing. They use binary classifiers to indicate if a bus is full or not. [15] estimates the number of passengers boarded via Wi-Fi (Wireless Fidelity) prob request. With this kind of data, it is possible to estimate the variation of PTO during a trip. The strategy used by [16] is to combine data collected by pass-through cards and GPS (Global Positioning System) data, to estimate PTO.

3 Proposed Approach and Practical Experiments

The approach here proposed is similar to that one used by [15] and [16]. First, we digitally collect passengers data by means of smartphones Wi-Fi probe requests, during a full bus trip. At the same time, we collect GPS information on that trip and mix up these data to estimate the times and positions each smartphone was first detected, and last seen. After the trip ends, collected data are retrieved to a central server, where they are processed to generate PTO estimation. Finally, the raw PTO estimation is matched to actual PTO, measured during the bus trip,

Table 1: Example of Dataset.

Door	**1**		**2**		**3**		**Total**		**PTO**
Time	**In**	**Out**	**In**	**Out**	**In**	**Out**	**In**	**Out**	
5:59:45	33	0	0	0	0	0	33	0	33
6:00:06	5	0	0	0	0	0	5	0	38
6:00:52	5	0	0	0	0	0	5	0	43
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
7:42:19	0	0	0	5	0	0	0	5	11
7:42:58	0	0	0	3	0	2	0	5	6
7:44:33	0	1	0	5	0	3	0	6	0

to adjust mathematical models that will be able to compensate the deviation between raw estimation and actual PTO.

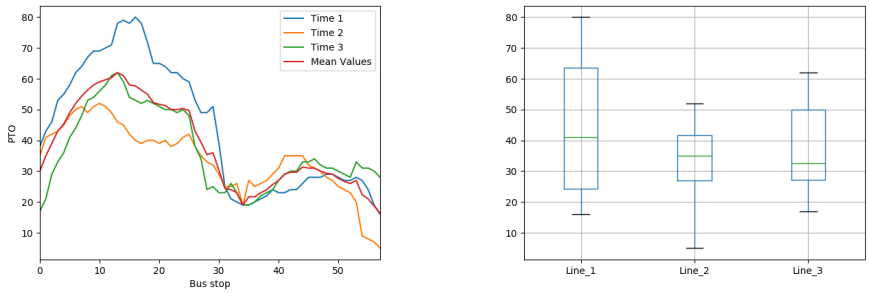
Between June 16th and 20th, 2018, a quantitative survey was carried out on public transportation trips of three different metropolitan lines in the city of Belo Horizonte. Time schedules were set to obtain data representing the morning peak period (between 6:00 am and 8:00 am), the afternoon peak period (between 4:30 p.m. and 6:30 p.m.) and off-peak period. Thus, it is possible to have samples with more fidelity to passenger flow characteristics for each line. This survey was performed by manually counting the flow of passengers entering and leaving the bus at each stop point, during a full trip. An example of the collected data can be seen in Table 1.

Table 1 contains data showing the number of people entering and leaving the bus during a full trip. At each moment the bus goes through a stop, passenger flow and the time stamp in which this stop occurred are noted. With this data, it is possible to estimate the temporal variation of PTO for that line. Figure 2a shows an example of hourly variation of PTO for the data presented on Table 1.

The data presented in Figure 2a vary according to the time of the trip (peak or off peak period), the day of the week, month and line searched. Figure 2a shows the variation of the PTO for three different times in the same searched line. As this variance exists, for each line surveyed, the mean of the hourly hourly values of the PTO during the same period was calculated for June 16 to 20, 2018. This average PTO value will be used to compare the different obtained.

4 Results and Discussion

Initially, in order to verify the existence or not of outliers in our research data, we used boxplots, presented at Figure 2b. From the diagram, using the indication of outlier data with more than 1.5 interquartile range, it is possible to conclude that no outliers were found on collected data. For the computational intelligence algorithms tested, random initial weights were used, so we repeated the algorithms execution for 100 times to find unbiased compensated prediction models of PTO.



(a) PTO from same bus line (3 times). (b) Boxplot from different lines.

Fig. 2: Dataset sample.

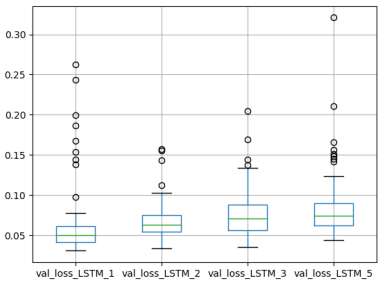
For RNN networks, the configurations with 1, 2, 3 or 5 delays of observations were used to predict the next PTO. These configurations were employed using LSTM and GRU approaches. We used 2 hidden layers with 64 and 32 neurons, respectively, loss function MSE and dropout of 20% in all tests. Figure 3a presents the comparative boxplot between these 3 configurations used in the LSTM algorithm. Table 2 presents statistical data after 100 runs of each configuration (1, 2, 3 or 5 delays). This table contains normality (Shapiro-Wilk) and post hoc (Nemenyi) tests. By the analysis of Figure 3a and Table 2, it is possible to evaluate that "statistical evidences exist that there is difference between performances" of RNN with LSTM using 1 or 2, 3 e 5 observations to predict the next PTO value. We found out that the higher the number of observations is, the worse the performance of the algorithm is. Figure 4 shows the autocorrelation (ACF) function of original series, with 1 difference and 2 differences.

Table 2: Statistics for LSTM.

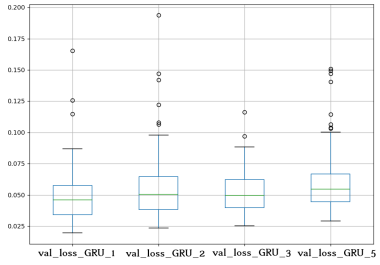
		p_value: Nemenyi test			
	p_value: Shapiro-Wilk	**LSTM_1**	**LSTM_2**	**LSTM_3**	**LSTM_5**
LSTM_1	2.3e-15	-1	0.001	0.001	0.001
LSTM_2	0.791		-1	0.885	0.578
LSTM_3	0.87			-1	0.900
LSTM_5	0.71				-1

Figure 3b and Table 3 present the same tests performed for RNN using GRU. As it can be seen at Table 3, using 1 delay, the result of 100 runs does not represent a normal function. But using 2, 3 or 5 delay, we can obtain a normal distribution of the loss function.

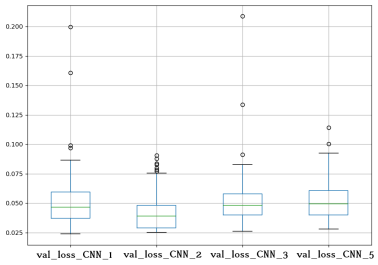
For CNN algorithm, 1D convolution window with length 3, 64 filters and average pooling was used. Figure 3c shows a boxplot for 4 different input con-



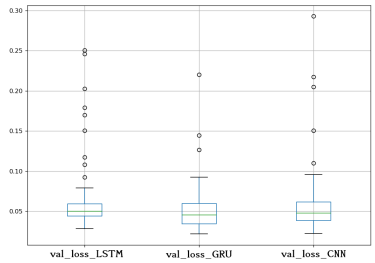
(a) Boxplot for RNN using LSTM.



(b) Boxplot for RNN using GRU.



(c) Boxplot for CNN.



(d) Boxplot LSTM vs GRU vs CNN.

Fig. 3: Boxplot for all tests.

figurations (1, 2, 3 or 5). Table 4 shows the statistical analysis for Figure 3c. Again, it can be seen the same behavior as the previous case.

Figure 3d and Table 5 present a comparison between results obtained using 1 delay for each algorithm (RNN LSTM, RNN GRU and CNN) in the time series forecast. Table 5 also displays the values of R^2 score for each prediction. Figure 5 shows the results obtained with the use of the best weights found for the case of 1 delay. Although the highest value of R^2 was found with the use of CNN algorithm, we did not find statistical evidence on which approach presents the best results for PTO forecasting.

Finally, we compared the models obtained by computational intelligence algorithms with a parametric approach using Autoregressive Integrated Moving Average (ARIMA). For ARIMA model, the ARIMA(0,2,1) configuration was used because, by Figure 4 and the test results presented in Table 6, this was the best configuration found. Using ARIMA (0,2,1), we found the prediction shown by Figure 6. To confirm the validity of the results, we performed a residue test, presented in Figure 7. Analyzing these results, it is possible to see that ARIMA model approach is statistically valid. Comparing the results obtained through computaional intelligence models with the ARIMA approach, we verified that

Table 3: Statistics for RNN using GRU.

		p_value: Nemenyi test			
	p_value: Shapiro-Wilk	**GRU_1**	**GRU_2**	**GRU_3**	**GRU_5**
GRU_1	3.8e-9	-1	0.452	0.452	0.004
GRU_2	0.820		-1	0.900	0.221
GRU_3	0.949			-1	0.221
GRU_5	0.828				-1

Table 4: Statistics for CNN.

		p_value: Nemenyi test			
	p_value: Shapiro-Wilk	**CNN_1**	**CNN_2**	**CNN_3**	**CNN_5**
CNN_1	4.6e-12	-1	0.001	0.900	0.900
CNN_2	0.857		-1	0.001	0.001
CNN_3	0.655			-1	0.900
CNN_5	0.92				-1

Table 5: Statistics for RNN LSTM vs RNN GRU vs CNN.

			p_value: Nemenyi test		
	p_value: Shapiro-Wilk	**R2 score**	**LSTM_1**	**GRU_1**	**CNN_1**
LSTM_1	7.3e-16	0.827	-1	0.235	0.860
GRU_1	1.9e-10	0.893		-1	0.885
CNN_1	2.9e-15	0.883			-1

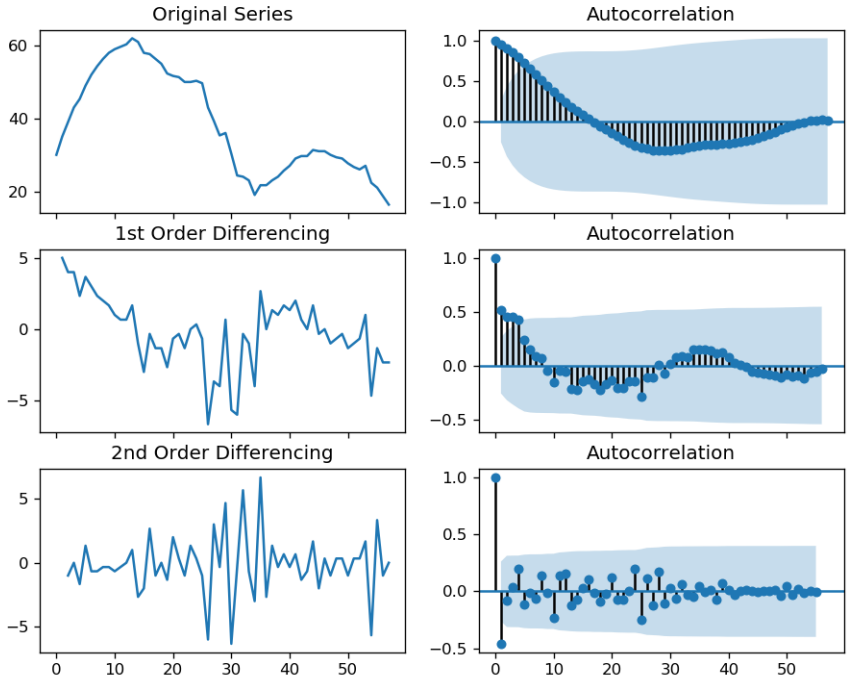


Fig. 4: ACF for the mean PTO (Figure 2a).

both can be used as predictors for PTO. For PTO forecasting application, we prefer to use ARIMA models because, with the ARIMA model, it is possible to obtain a time series equation. Thus, this equation can be used in the mathematical modeling of PTO variation during a trip.

5 Conclusion

To estimate the variation of PTO during a trip can be important to better planning and management of urban public transportation. By means of PTO, it

Table 6: Statistics of ARIMA(0,2,1) model.

	Performance		**Coef**	**Std err**	**P_value**
AIC	224.447	**Const.**	-0.1152	0.086	0.185
BIC	230.523	**MA**	-0.6337	0.091	0.000

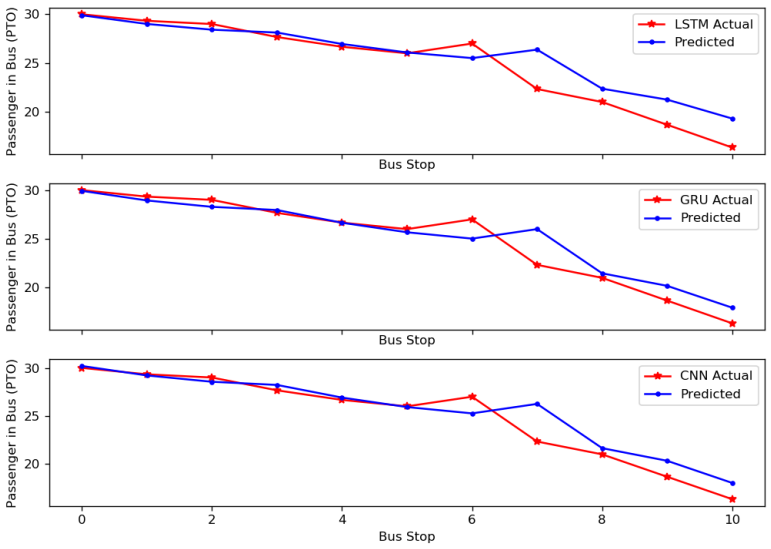


Fig. 5: Prediction using RNN LSTM, RNN GRU and CNN.

is possible to adjust the number of buses to the population demand. Thus, it is possible to have more realistic bus ticket prices, to reduce bus overcrowding and, overall, to improve the quality of provided service, and to attract more users. With more users on public transportation, we will have a smaller number of cars on the streets. This reduces greenhouse gas emissions and noise pollution. Therefore, the quality of life of the city's residents is also improved.

After comparing different configurations of computational intelligence algorithms (RNN with LSTM / GRU and CNN), we noticed that as the number of delays used as input increases, the quality of the prediction decreases. This behavior suggests a low autocorrelation between time series data. Comparing the 3 approaches (RNN with LSTM / GRU and CNN), statistically, there is no difference when they use only 1 input delay. So we can use any of those to predict PTO. All strategies reached R^2 close to 0.9.

Applying the ARIMA model, we also achieveed results very close to those achieved by computational intelligence approaches. The advantage of using ARIMA is that we can find an explicit mathematical model that describes the behavior of the time series. This feature is important because it is possible to understand the variation of PTO during a trip. Therefore, for the prediction of PTO, it is suggested to adopt the ARIMA technique.

In order to establish the best model for PTO forecasting, more research is needed, because several factors can influence time series behavior. Thus, inves-

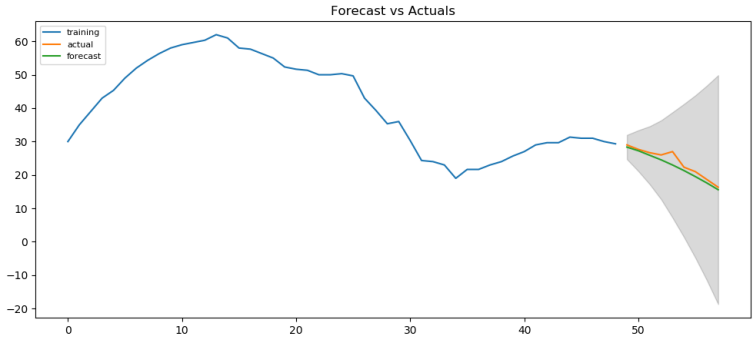


Fig. 6: Prediction using ARIMA(0,2,1).

tigating other non-linear approaches may be interesting to predict the variation of PTO during a trip.

Acknowledgments. Authors would like to thank CAPES Foundation, CEFET-MG, CNPq and FAPEMIG for the financial support to this project.

References

1. T. Litman, "Transportation and public health," *Annual review of public health*, vol. 34, pp. 217–233, 2013.
2. M. D. d. Santos, M. F. Silva, L. A. Velloza, and J. E. Pompeu, "Lack of accessibility in public transport and inadequacy of sidewalks: effects on the social participation of elderly persons with functional limitations," *Revista Brasileira de Geriatria e Gerontologia*, vol. 20, no. 2, pp. 161–174, 2017.
3. K. M. Gwilliam, *Cities on the move: a World Bank urban transport strategy review*. The World Bank, 2002.
4. D. Sperling and D. Gordon, *Two billion cars: driving toward sustainability*. Oxford University Press, 2010.
5. R. G. Ribeiro, *Estudo dos Deslocamentos Urbanos da Classe Mdia Brasileira na Regio Metropolitana de Belo Horizonte*. PhD thesis, COPPE/UFRJ Rio de Janeiro, RJ, Brasil, 2015.
6. D. Banister, "The sustainable mobility paradigm," *Transport policy*, vol. 15, no. 2, pp. 73–80, 2008.
7. T. Litman, *Evaluating public transportation health benefits*. Victoria Transport Policy Institute, 2012.
8. M. Voinescu, A. Udrea, and S. Caramihai, "On urban traffic modelling and control," *Journal of Control Engineering and Applied Informatics*, vol. 11, no. 1, pp. 10–18, 2009.
9. O. K. Tonguz, W. Viriyasitavat, and F. Bai, "Modeling urban traffic: a cellular automata approach," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 142–150, 2009.

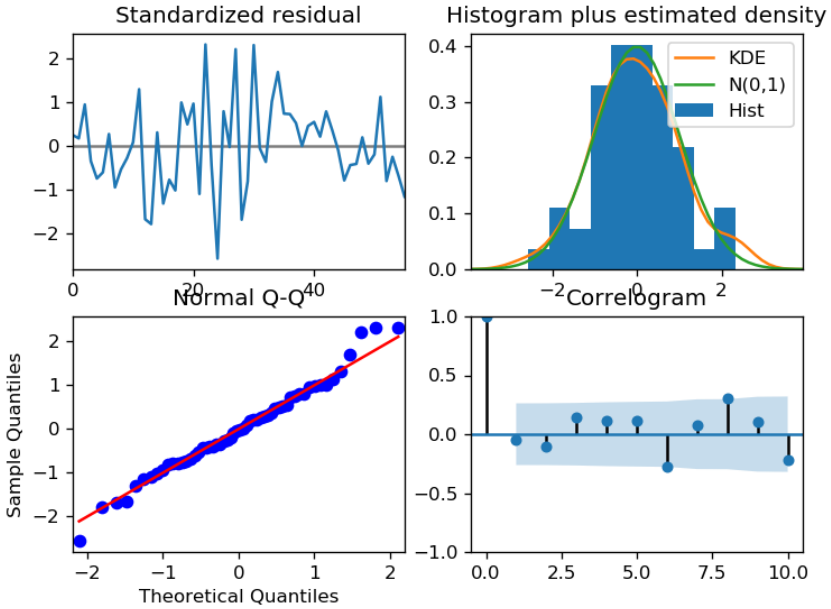


Fig. 7: Residues test for ARIMA(0,2,1).

10. F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.
11. V. Khabarov and A. Tesselkin, "Method for estimating origin-destination matrices using markov models," in *2016 11th International Forum on Strategic Technology (IFOST)*, pp. 389–393, IEEE, 2016.
12. R. Tolouei, S. Psarras, and R. Prince, "Origin-destination trip matrix development: Conventional methods versus mobile phone data," *Transportation research procedia*, vol. 26, pp. 39–52, 2017.
13. R. R. Cura, R. Stickar, C. Delrieux, F. Tohmé, L. Ordinez, and D. Barry, "Modeling the origin-destination matrix with incomplete information," in *International Conference on Ubiquitous Computing and Ambient Intelligence*, pp. 121–127, Springer, 2017.
14. S. Brandon, "Estimating passenger flow and occupancy on board public transport buses through mobile participatory and opportunistic sensing," Master's thesis, University of Dublin, Trinity College, 2015.
15. L. Mikkelsen, R. Buchakchiev, T. Madsen, and H. P. Schwefel, "Public transport occupancy estimation using wlan probing," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 302–308, IEEE, 2016.
16. J. Zhang, D. Shen, L. Tu, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, "A real-time passenger flow estimation and prediction method for urban bus transit systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3168–3178, 2017.

Applications of Statistical and Machine Learning Methods for Predicting Time-Series Performance of Network Devices

Naveksha Sood[1,a], Usha Rani[2,b], Srikanth Swaminathan[2,c], George Abraham[2,d], Dileep A. D.[1,e], and Varun Dutt[1,f]

[1]Indian Institute of Technology Mandi, Himachal Pradesh, India – 175005
[2]NMSworks Software Pvt. Ltd., India
[a]naveksha_sood@projects.iitmandi.ac.in, [b]usha@nmsworks.co.in,
[c]ssrikanth@nmsworks.co.in, [d]george@nmsworks.co.in,
[e]addileep@iitmandi.ac.in, and [f]varun@iitmandi.ac.in

Abstract Prediction of performance of network devices, which control the flow of data, is of utmost importance to be able to manage the network efficiently. In this paper, we used a statistical auto-regressive integrated moving average (ARIMA) model and a machine learning (ML) multi-layer perceptron (MLP) model to forecast the performance of a network device 15-minutes ahead in time. Forecasting was done with one-feature (univariate) and multiple features (multivariate). Also, we evaluated the effects of highly correlated features on our predictions in MLP and ARIMA models. Results revealed that the ARIMA model performed better compared to the MLP model for univariate data; however, the MLP model performed better compared to the ARIMA model with exogenous variables for multivariate data. In addition, keeping highly correlated features in the models improved model predictions on multivariate data. We highlight the real-life implications of using statistical and ML models for time-series forecasting of network data.

Keywords: Network Devices • Auto-Regressive Integrated Moving-Average Model • Multi-layer Perceptron Model • Time-series Forecasting • CPU Utilization

1 Introduction

A rapid advancement in digital infrastructure and services has led to an increased popularity of the Internet over the years [1]. In fact, telecommunication networks allow numerous users to connect to the Internet ensuring high-speed access and security. In particular, broadband technologies address the bandwidth gap at the last mile and these technologies are extensively deployed to provide high speed Internet access to subscribers [2]. Service providers may use network management systems (NMSs) to constantly monitor the network devices for fault and performance to ensure network health and satisfactory customer service [3]. A predictive algorithm that augments NMSs and provides information regarding a possible aberrance in the network can help in effective fault avoidance and resource allocation. The primary objective of this work is to provide predictive performance management capability via machine-learning and statistical models for a broadband network owned by a large telecom operator.

The main components in a broadband network are Digital Subscriber Line (DSL) modems, Optical Network terminals or WiFi routers, and these components connect to the access layer via technologies such as DSL Access Multiplexers (DSLAMs), Optical Line Terminals or Ethernet Local Area Network switches [4]. These nodes are aggregated using Resilient Packet Ring switches and Synchronous Digital Hierarchy rings, on to the broadband network edge [4]. The Broadband Network Gateway (BNG) is an important device (a router) at the network edge, and the BNG performs subscriber management including session and circuit aggregation, authentication/authorization/accounting, policy and traffic management functions in addition to routing [5]. The BNG device statistics are monitored by NMSs and provide the service provider critical information regarding customer service and experience [5]. Given the important role the BNG plays with respect to customer experience, this work focuses on prediction of BNG performance parameters, to enable service providers to take a proactive approach to impending failures or performance problems in the network. In this paper,

we develop and test a statistical modelling technique (ARIMA) and a machine learning (ML) modelling technique (MLP) to predict the performance of BNG devices over time as a univariate and multivariate time-series. Univariate time-series modelling techniques infer the pattern over time and illustrates how a performance measure, e.g., CPU utilization depends on its value at previous time instances [6]. In addition, multivariate time-series modelling techniques consider the effect of other variables on the performance measure [6]. Overall, statistical and ML modelling has shifted the traditional reactive approach to handle faults to a more proactive approach of preventing faults in a network [7].

Prior research has been done to compare traditional network management protocols for fault management [8]. Prior research has also performed a classification of regular and anomalous functioning of a telecom network using hidden state Markov model (HSMM) [9]. Various studies have tried to classify and predict CPU utilization patterns of a device and data centre servers, and traffic patterns on networks to predict resource exhaustion [10-15]. Some studies have increased the scope of their work to multiple devices [16-17]. Researchers have also worked on predicting the performance of networks [18-20]. However, to the best of author's knowledge, only a handful of studies have tried to compare statistical and ML algorithms for performance prediction of network devices.

In this research, we address this literature gap by using statistical and ML modelling for time-series performance data of a BNG device to predict the performance of the device at future time instances over a one-month period. Specifically, we develop an ARIMA model and its extension ARIMA model with exogenous variables (ARIMAX) to model and predict the performance of a device ahead in time. We compare the results of a statistical ARIMA model with an MLP model for univariate and multivariate prediction ahead in time. Also, we evaluate the effect of considering only the highly correlated features on our model predictions.

In the following sections, we present prior research involving performance data modelling and failure prediction. Second, we elaborate on the methodology used, explaining the functioning of the ARIMA and MLP models. Third, we present how models were calibrated using a genetic algorithm framework. Next, we compare the results for both ARIMA and MLP models. Finally, we discuss the implication of our results in real world and detail future research directions in this research program on modelling of network device data.

2 Background

Statistical and ML algorithms may help to predict the performance of network devices and this prediction may give useful insights to the service providers for managing the network more efficiently.

Prior research has investigated traditional network management protocols such as simple network management protocol (SNMP) and Common Object Request Broker Architecture (CORBA), both of which worked on passive fault corrective measures [8]. With the advent of ML, researchers are working on proactive approaches to classify the performance of a large telecom network and predict failures [9]. For example, reference [9] used HSMM after an elaborate data pre-processing technique and found the prediction accuracy to improve by a large amount.

Some researchers have tried to predict resource utilization of devices in a cloud environment and datacentre; and network traffic patterns to aid in better performance of the network [10-15]. For example, reference [10] used a deep recurrent network with LSTM units to predict server load and performance of two servers of a datacentre.

Reference [11] used a LSTM-based network to predict the multivariate mobile network traffic data to facilitate proactive resource allocation. These authors compared their LSTM-based network with an ARIMA model and a feed-forward neural network (FFNN) model and reported LSTM to be performing significantly well. Reference [12] compared statistical modelling techniques such as triple exponential smoothing (TES) and seasonal ARIMA (SARIMA) with ML modelling techniques like multi-layer perceptron (MLP) and LSTM to predict univariate network traffic data of a University. These authors found the SARIMA model to be performing better compared to the MLP and LSTM models. Reference [13] compared various techniques such as Predictive Elastic Resource Scaling (PRESS), ARIMA, Non-Linear Autoregressive neural network (NARNN), LSTM, and a Bidirectional LSTM (BLSTM) to predict multivariate time-series data of load on a machine in a cloud environment multiple steps ahead in time. They explored various feature selection techniques like Pearson's correlation, Spearman's correlation, and Granger's causality to choose the independent variables. Reference [14] compared ARFIMA, Bayesian models, Kalman Filter, and clustering to predict the CPU utilization of a server to aid dynamic server virtualization. Reference [15] have proposed a Time-aware Residual Network (T-ResNet) to model the CPU utilization of a cloud server.

Some researchers have extended their work to more than one device to make performance predictions [16-17]. Reference [16] have used LSTM-based neural networks to predict the workload of a task by fitting the model to previous workloads of many tasks. These authors performed clustering and achieved better results compared to those obtained by using pre-existing techniques. Reference [17] trained a random forest classifier on the time-series data of 250 systems to predict events in the functioning of a software, achieving 81% classification accuracy.

Various studies have been done on predicting the performance of networks [18-20]. Reference [18] used a Random Forest model on online streaming data using Apache Spark software to predict the appearance of failures in future. Reference [19] used a multivariate Recurrent Neural Network (RNN) model to predict the failures in a broadband network. These authors considered the effect of external events such as weather forecasts and maintenance work on the occurrence of faults, which turned out to significantly improve the accuracy. Reference [20] predicted the disruption and degradation of services of a home network, which yielded a prediction accuracy of 75%.

Although, several researchers have worked to classify and predict the performance of a device or a network using specific techniques [9, 17-20], but limited work has been done to compare statistical and ML modelling approaches on univariate and multivariate time-series data concerning the performance of network devices. In this paper, we address this literature gap by developing ARIMA and MLP models to predict the performance of a network device. We hope these ARIMA and MLP models help avoid faults and bottlenecks in broadband networks.

3 Method

3.1 Data

Data used in this study are from a Network Management System managing a large telecom service provider in India. The NMS database contains a log of various performance measures recorded for every 15-minutes for approximately one-month period between 9th August, 2018 and 7th September, 2018. Table 1 summarizes the performance measures collected for a BNG device and their significance.

Table 1. Performance Measures of BNG Device

Performance Measure	Description	Units

CPU Utilization	Percentage of CPU resources being used	Percentage
Active Count	Number of active users.	Number
Authenticate Count	Number of users waiting to be authenticated.	Number
Total Memory	Total memory of the BNG device	Bytes
Average Temperature	Average temperature of various sensors	Celsius
Total In Bandwidth	Total traffic coming into the BNG device	Bytes
Total Out Bandwidth	Total traffic going out of the BNG device	Bytes

The NMS is configured to generate "threshold alarms" when a performance parameter of interest exceeds or falls short of a pre-configured threshold value. The first step in the comparison process involved anonymizing data, cleaning it by removing outliers and filtering out erroneous records. Table 2 provides the central tendency measures, across features in data.

Table 2. Statistical Characteristics of Data[1]

Performance Measure	Max Value	Min Value	Mean	Variance
CPU Utilization	62	8	18	69
Active Count	1.64×10^{04}	7.25×10^{03}	1.15×10^{04}	8.00×10^{06}
Authenticate Count	130	0	4	25
Total Memory	47	42	43	4
Average Temperature	37	32	35	0.66
Total In Bandwidth	6.92×10^{08}	0	2.89×10^{08}	3.17×10^{16}
Total Out Bandwidth	6.87×10^{08}	0	2.88×10^{08}	3.07×10^{16}

[1] Due to the non-disclosure agreement, device IDs were anonymised.

For this study, CPU utilization is being used as a dependent variable for univariate and multivariate modelling and other performance measures are added as independent variables depending on their correlation coefficients for multivariate modelling. The models were built for a BNG device that recorded the maximum number of alarms. A total of 2785 records were available for the device, of which 2088 (75%) have been used for training and the remaining 557 (25%) for testing. Fig. 1 shows the trend of CPU utilization for 3 weeks of training data and last 1 week of test data.

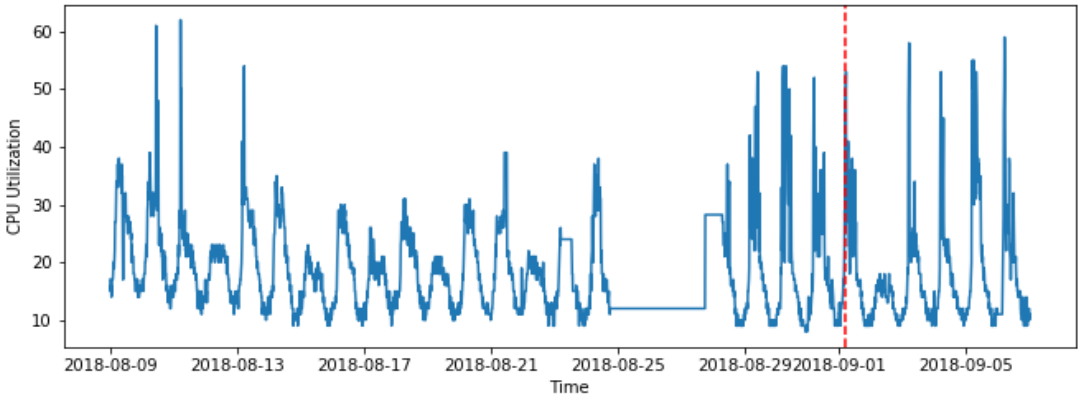


Fig. 1. CPU Utilization of a BNG device over one-month period

Next, we detail the statistical and ML algorithms that were considered for our evaluation.

3.2 Statistical and ML Algorithms

Auto-Regressive Integrated Moving-Average (ARIMA)

ARIMA is statistical modelling and forecasting technique used to capture the temporal structures of a univariate time-series data [21]. ARIMA model generally works on stationary data, however, it can handle the data with a trend with the help of transforming techniques such as differencing. Also, data with seasonal component can be handled by an extension of the ARIMA model called Seasonal ARIMA (SARIMA) [21]. It can also be extended to include multiple features as exogenous variables (ARIMAX) [21].

ARIMA, according to its acronym, can be broken down into three parts:

Stationarity of Time-Series: A time-series with constant values over time for mean, variance and auto-correlation is said to be stationary in time. In other words, the properties of time-series are not dependent on time. Most statistical forecasting methods assume that a time-series can be made approximately stationary using mathematical transformations such as differencing [22]. A time-series with t observations, on differencing will yield a time-series with t-1 observations, as:

$$y_t{}' = y_t - y_{t-1} \quad (1)$$

Auto-Regressive Models: In an auto-regressive model, we assume that the value of a variable at time t is a function of its values at p time instances before it [21]. Thus, an auto-regressive model is defined as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + \epsilon_t \quad (2)$$

where p is the auto-regressive trend parameter, ϵ_t is white noise and $y_{t-1}, y_{t-2} \cdots y_{t-p}$ denote the CPU utilization at previous time periods.

Moving-Average Models: A moving-average model assumes that the prediction at time t is a function of error between actual value at time t-1 and the moving average in a regression model [21]. A moving-average model is defined as:

$$y_t = c + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-} \quad (3)$$

where q is the moving-average trend parameter, ϵ_t is white noise and $\epsilon_{t-1}, \epsilon_{t-2} \ldots \epsilon_{t-q}$ are the error terms at previous time periods.

If we combine auto-regression and a moving-average model on stationary data, we obtain a non-seasonal ARIMA model, which is defined as:

$$y'_t = c + \phi_1 y'_{t-1} + \ldots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (4)$$

ARIMAX builds upon an ARIMA model and incorporates exogenous variable. An ARIMAX model assumes that the prediction at time t is not only a function of it's values at previous time periods but also a function of exogenous variable(s) [22]. Thus, an ARIMAX model is defined as:

$$y'_t = \beta x_t + c + \phi_1 y'_{t-1} + \ldots + \phi_p y'_{t-p} + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q} + \epsilon_t \quad (5)$$

where x_t is the value of exogenous variable at time t.

Multilayer Perceptron (MLP)

Multi-layer perceptron is a class of feed-forward neural network (FFNN), with at least three layers: an input layer, an output layer and one or more hidden layers which can be assumed to make up the actual engine of the MLP. Each neuron of one layer of an MLP is connected to each neuron of the next layer and every connection has some weight associated with it as depicted by fig. 2.

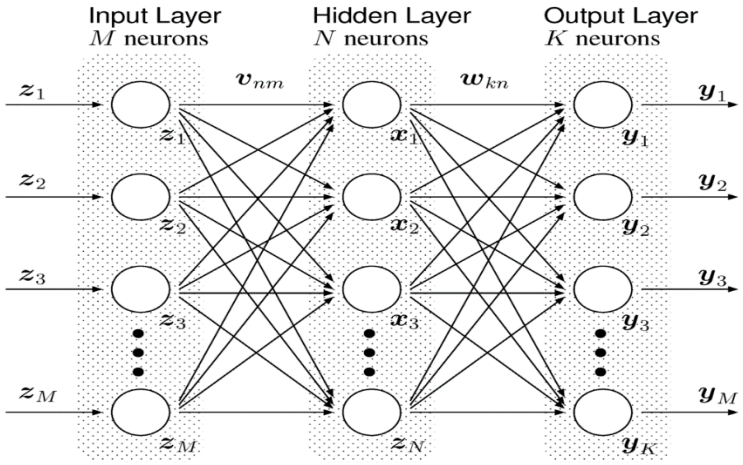


Fig. 2. Architecture of a Multi-Layer Perceptron [23]

It follows a supervised learning technique of back-propagation, where during the training phase, weights are randomly initialized, the model is given input-output pairs in small batches, input is applied to the model and the error between predictions and ground truth for each batch is calculated [24]. All the neurons except for the neurons of input layer have an activation function which also introduces stochasticity in the model.

At output node 'n' error in t^{th} training point is denoted by:

$$e_n(n) = d_n(t) - y_n(t) \tag{6}$$

where d is the actual value and y is the predicted value. With the help of optimizer being used by the model, the node weights are changed such that the error in the entire output is minimized, which is defined as:

$$\varepsilon(t) = \frac{1}{2}\sum_n e_n^2\,(t) \tag{7}$$

Using the gradient descent method, the change in each weight is:

$$\Delta w_{kn}(t) = -\eta\,\frac{\partial\varepsilon(t)}{\partial v_{k(t)}}\,y_n(t) \tag{8}$$

where y_n is the output of the previous neuron and 'η' is the learning rate of MLP [25]. As this process is repeated in batches for the entire training set, it constitutes one epoch, and hence the process is iterated for many epochs, giving us optimal weights.

In the next section, we detail how the two models were calibrated.

4 Model Calibration

For both the ARIMA and MLP models, free parameters were optimized using a Genetic Algorithm (GA) program. GA is a parameter optimization technique based on bio-inspired operators such as mutation, crossover and selection. It is an iterative procedure where at each iteration a random set of parameters is selected as parents and are evolved to make children. One iteration is called a generation and over successive generations, population evolves towards an optimal solution [26]. Parameters in different models were varied and the variation in parameter values ensured that the optimization captured the optimal parameter values with high confidence by minimizing the Root Mean Square Error (RMSE) between actual and predicted value. In GA, the crossover and mutation rates were kept at their default values of 80% and 1% respectively. The GA was evolved over 70 generations having 45 population per generation. The stopping criteria were defined as no change in the fitness function for the last 12 generations. The predictions were estimated using one-step ahead walk-forward validation method, where actual data at time t is used to make the prediction on time t+1, allowing the model to use all available data to make the most accurate prediction [27].

Highly co-related features were chosen based on Pearson's Correlation. Pearson's correlation is the linear correlation between two variables say, CPU utilization and active count. It can range between -1 to 1. A high positive value of Pearson's correlation coefficient (PCC) shows a strong positive relation between the two variables, i.e., as the active user count increases, the CPU utilization of the network device also increases. A lower value of PCC depicts a weak positive correlation. Likewise, negative PCC, depicts a negative correlation, i.e., if the value of one variable increases, the value of another variable decreases. PCC is defined as:

$$\rho_{X,Y} = \frac{COV(X,Y)}{\sigma_X \sigma_Y} \tag{9}$$

Where COV is covariance, σ_X, σ_Y is the standard deviation of variable X and Y.

4.1 ARIMA Model

First, augmented Dickey-Fuller (ADF) test was performed to confirm the stationarity of data and it revealed the time-series to be stationary.
Using GA free parameters p, d, and q were optimized. Parameters p and q were varied as the integer values in [0, 6], and d was kept 0.

4.2 Multi-layer Perceptron Model

The architecture of MLP model was optimized using GA as well. Various parameters used to define an MLP model such as number of layers, number of nodes, batch size, look back period, epochs and others, were varied in the range given in Table 3. Since MLP models use activation functions which introduce stochastic nature in the model, the entire process of training and prediction for a particular combination of parameters was repeated 10 times and the average root mean square error between modelled and actual data was minimized.

Table 3. Parameter Optimization of MLP

Parameter	Range of Values
Look Back Period	1, 2, 3, 4, 7, 14, 21
Number of Hidden Layers	1, 2, 3, 4, 5

Number of Nodes	16, 32, 64, 128, 256
Number of Epochs	40, 80, 120, 160, 200
Batch Size	1, 2, 3, 4, 5
Activation Functions (in each layer except the input layer)	Rectified Linear(ReLu), Linear, Sigmoid, Tangent Hyperbole (tanh)
Optimizer	Adaptive Moment Estimation(Adam), Root mean Square Prop(RMSProp)

5 Results

5.1 Univariate Time-Series

Table 4 shows the training and testing root mean square error (RMSE) values in the ARIMA and MLP models for the univariate CPU utilization time-series. The ARIMA model performed better compared to the MLP model during both training and test.

Table 4. Results for univariate time-series

Model	Train Error	Test Error
ARIMA	3.114	4.458
MLP	3.457	4.989

Fig. 3 shows the ground truth and predictions for ARIMA and MLP models. The calibrated ARIMA model had the lag values p as 5, q as 0, and d as 0 (the time-series was stationary). The calibrated MLP model had 1 hidden layer with 128 neurons, 1 time-step as the look back period, 4 batch size, and 120 epochs. It used the ADAM optimizer and ReLU activation functions. As shown in Fig. 3, the ARIMA model possessed a slight edge over the MLP model in predicting the CPU utilization.

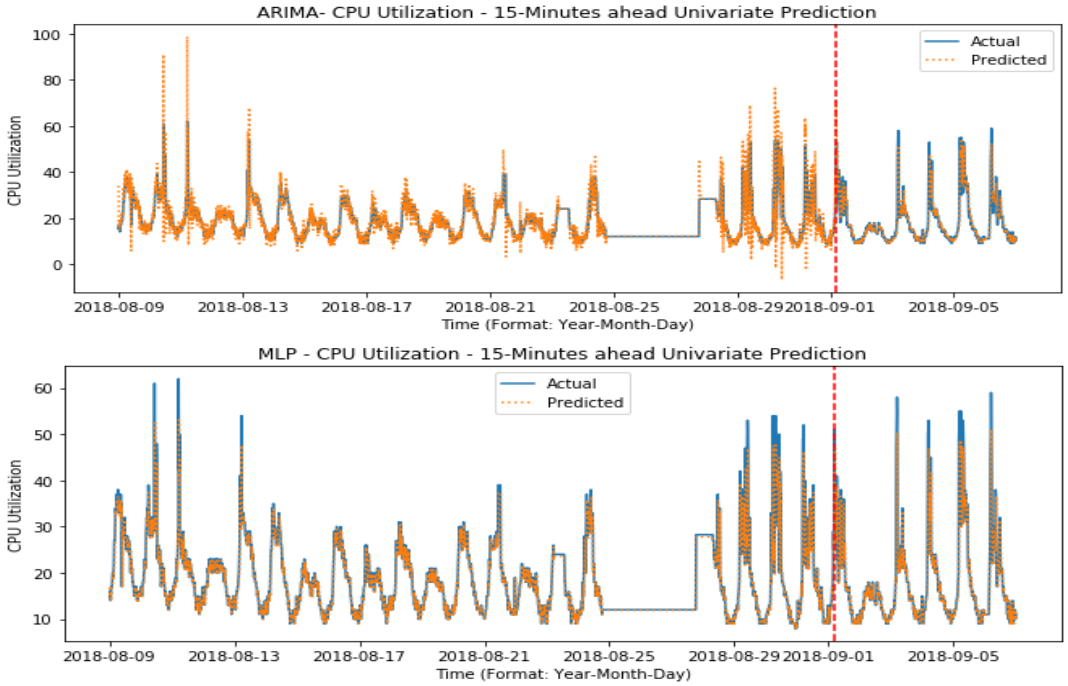


Fig. 3. Univariate ARIMA and MLP results during training and test

5.3 Multivariate Time-Series with all features

Table 5 shows the training and testing root mean square error (RMSE) values in the ARIMA and MLP models for the multivariate CPU utilization time-series considering all features in data (in ARIMA model, the features entered as exogenous variables). The ARIMA model performed worse compared to the MLP model during both training and test. Also, the ARIMA model showed over-fitting in data, where the test RMSE was greater than the training RMSE.

Table 5. Results for multivariate time-series with all the features

Model	Train Error	Test Error
ARIMA	2.998	20.179
MLP	2.818	5.902

Fig. 4 shows the ground truth and predictions for multivariate time-series from the ARIMA and MLP models (these figures correspond to the RMSE shown in Table 5). The calibrated ARIMA model had the lag values p as 3, q as 2, and d as 0 (the time-series was stationary). The calibrated MLP model had 2 hidden layers with 256 neurons, 1 time-step as the look back period, 8 batch size, and 80 epochs. It used the ADAM optimizer for training and tanh as the activation function.

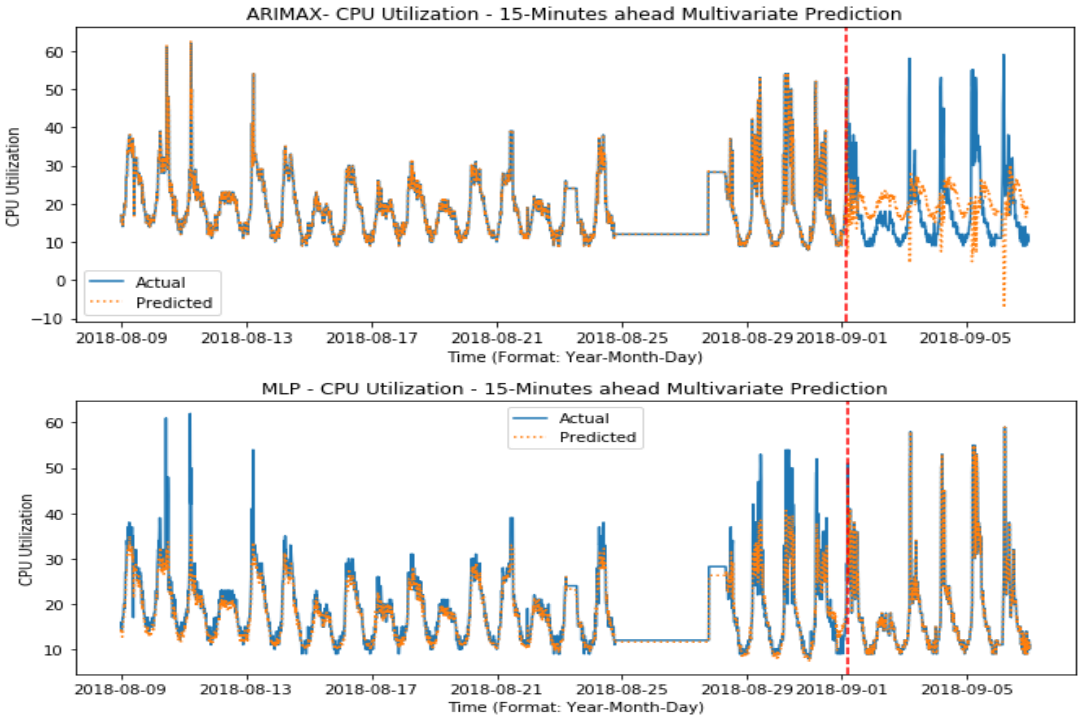


Fig. 4. Multivariate ARIMA and MLP results during training and test

5.4 Multivariate Time-Series with highly correlated features

As the multivariate RMSEs were poorer compared to univariate RMSEs during test from both models, we performed feature-engineering, the art of selecting relevant features, in data via Pearson's correlation. Table 6 shows Pearson's correlation between different features and CPU utilization in training data arranged in the descending order of their correlation coefficients and their respective *p*-values. Among all features, Active User Count, Total Out Bandwidth, and Total In Bandwidth possessed strong correlations with CPU utilization (>= 0.5) and these features were retained in data for ARIMA and MLP models.

Table 6. Pearson's correlation coefficient and *p*-values

Feature X correlated with Feature Y	r^2	*p*-value
Active User Count with CPU utilization	0.700	0.000
Total Out Bandwidth with CPU utilization	0.693	0.000
Total In Bandwidth with CPU utilization	0.691	0.000
Authenticate User Count with CPU utilization	0.222	0.000
Average Temperature with CPU utilization	0.169	0.000
Total Memory with CPU utilization	-0.034	0.107

[2] *r* refers to the Pearson correlation coefficient. Strongly correlated features are highlighted in bold letters.

Table 7 shows the training and testing root mean square error (RMSE) values in the ARIMA and MLP models for the multivariate CPU utilization time-series with only the three strongly correlated features (in ARIMA model, the three features entered as exogenous variables). Overall, during test, there was an improvement in the RMSEs of multivariate models with strongly correlated features compared to the multivariate models with all features. The MLP model performed better compared to the ARIMA model both during training and test.

Table 7. Results for multivariate time-series with three strongly co-related features

Model	**Train Error**	**Test Error**
ARIMA	3.014	13.401
MLP	2.977	5.441

Fig. 5 shows the ground truth and predictions for multivariate time-series from the ARIMA and MLP models with strongly correlated features (these figures correspond to the RMSE shown in Table 7).

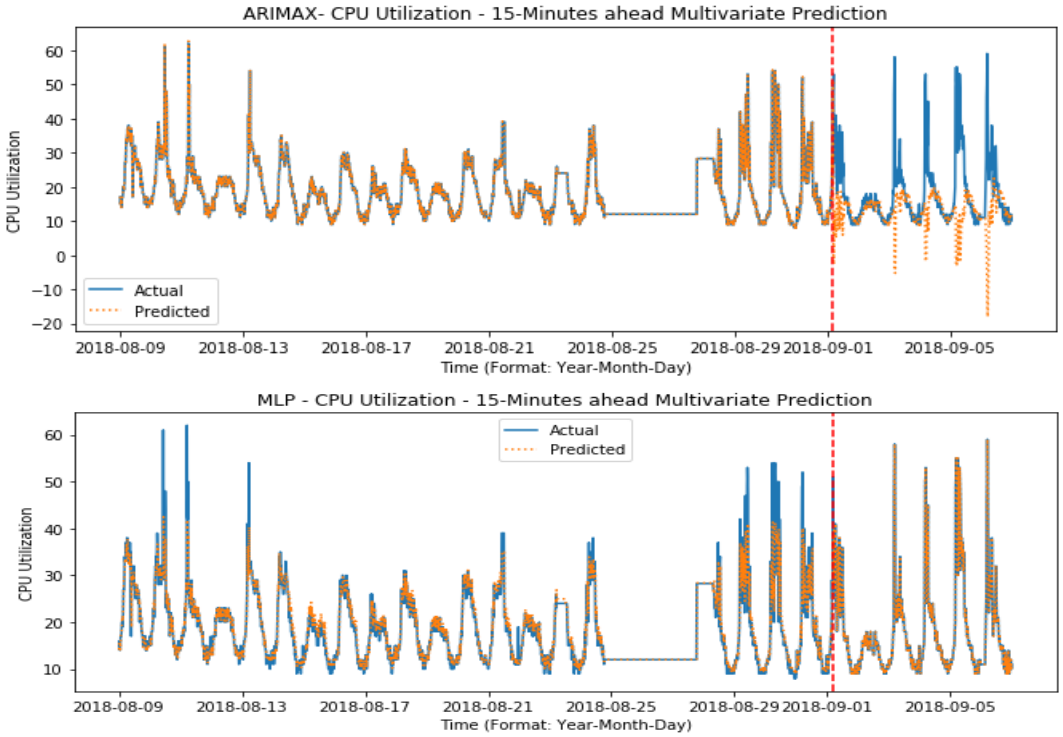


Fig. 5. Multivariate ARIMA and MLP results during training and test with strongly correlated features

6 Discussion and Conclusions

Network management services (NMSs) help network operators in finding faults during network operations. A predictive algorithm could help NMSs provide information regarding potential network aberrances ahead of their occurrence. The primary objective of this work was to provide predictive performance management capability via machine-learning and statistical models for a broadband network. Specifically, we developed a statistical ARIMA model and a machine-learning MLP model on univariate and multivariate CPU utilization data of a network device in a broadband network. Results revealed that univariate models performed better compared to multivariate models, where the univariate ARIMA model performed better compared to the univariate MLP model. Although the results of the multivariate models were poorer compared to the univariate models, the multivariate MLP model outperformed the multivariate ARIMA model. Overall, using only the strongly correlated features helped the multivariate models perform better compared to their multivariate versions with all features.

First, univariate models performed better compared to multivariate models. A likely reason for this result could be that other features in data do not add value beyond the CPU utilization feature. Thus, it may be better to predict the future CPU utilization value using the prior CPU utilization values compared to predicting the future CPU utilization value using prior CPU utilization values and other features.

Second, the univariate ARIMA model performed better compared to the univariate MLP model. A likely reason for this result could be that ARIMA model is lightweight with only three parameters and fewer parameters allow the model to understand the temporal structures in a stationary univariate time series. In contrast, the MLP model has several parameters and it may need relatively larger datasets to be trained properly.

Third, the multivariate MLP model outperformed the multivariate ARIMA model. A likely reason for these results could be that a linear ARIMA model lacks the power to understand different correlations among various features and handle a multi-dimensional data; whereas, a MLP model, which has several layers and several nodes per layer, can do so efficiently.

Fourth, the correlation study further gave us the insight into relationships among the features. These correlation results consolidate the logical reasoning that a strong correlation between features and CPU utilization in broadband network devices are helpful in modelling and forecasting values. Alternatively, presence of too many features, where some are less correlated, may spoil predictions.

Our results have various implications. First, ARIMA models may be a better time-series prediction technique for performance measure predictions in broadband network devices. Also, ARIMA with fewer parameters, may take less training time and provide robust performance. Second, collecting a number of performance measures about network devices (like temperature and memory utilization) may not necessarily lead to better model performance. In fact, collecting a single performance measure (like CPU utilization) may suffice to predict its future values.

There are several things to try as part of our future work in predictive modelling. First, as a part of our future work, we would like to investigate how models developed for a device can be generalized to many devices in the network. Next, it may be worthwhile to compare the correlation-based feature-engineering technique with other feature-engineering techniques like LASSO and RIDGE regression [28] and auto-encoders [29]. Furthermore, it may be interesting to investigate whether an increase in the dataset size causes an improvement in the prediction performance of the MLP model compared to the ARIMA model in network data. Finally, it may also be interesting to compare the approaches proposed in this paper with the existing traditional network management approaches (e.g., SNMP and CORBA [8]). We would like to pursue some of these ideas as part of our research program on prediction of network-related performance measures.

Acknowledgement: The project was supported by the grants (awards: #IITM/MHRD(UAY)/AD/115) to Dileep A.D. We thank NMSworks for their financial and computational support in this research.

7 References

[1] Internet World Stats (2019). *Internet usage statistics*.
[2] Broadband Technology (2019, June 14). *Gale Encyclopedia of E-Commerce*.
[3] Network Management System (NMS). *Techopedia Inc.*
[4] Telecommunications Network and Service Architectures Principles, Concepts and Architectures (2010). *Etudes et Formations en Telecommunications (EFORT).*
[5] Broadband Network Gateway Overview. *Cisco ASR 9000 Series Aggregation Services Router Broadband Network Gateway Configuration Guide, Release 4.3x.*
[6] Iwok I.A., Okpe A.S. (2016). A Comparitive Study between Univariate and Multivariate Linear Stationary Time Series Models. *American Journal of Mathematics and Statistics*.
[7] Proactive vs Reactive Artificial Intelligence (2019, March 25). *Princeton Data Labs*.
[8] Gu Q., Marshall A. (2004). Network Management performance analysis and scalability tests: SNMP vs CORBA. *Managing Next Generation Convergence Networks and Services, IEEE/IFIP Network Operations and Management Symposium*.
[9] Salfner F., Tshirpke S. (2008). Error Log Preprocessing for Accurate Failure Prediction. *USENIX Workshop on the Analysis of System Logs (WASL).*
[10] Huang Z., Peng J., Lian H., Guo J., Qiu W. (2017). Deep Recurrent Model for Server Load and Performance Prediction in Data Center. *Complexity.*
[11] Trinh H.D., Giupponi L., Dini P.(2018). Mobile Traffic Prediction from Raw Data Using LSTM Network. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*
[12] Sood N., Kaushik S., Nigam A., Kailasam S., Dileep A.D., Dutt V. (2019). Applications of Statistical and Machine Learning Time-Series Methods for Predicting Internet Usage. *International Conference on Machine Learning and Data Mining*
[13] Gupta S., Dileep A.D., Gonsalves T.A. (2018). A joint feature selection framework for multivariate resource usage prediction in cloud servers using stability and prediction performance. *The Journal of Supercomputing.*
[14] Kudinova M, Melekhova A., Verinov A. (2015). CPU Utilization prediction methods overview. *The 11th Central & Eastern European Software Engineering Conference, Russia.*
[15] Chen S., Shen Y., Zhu Y. (2018). Modelling conceptual characteristics of virtual machines for CPU Utilization Prediction, *Conceptual Modelling.*
[16] Yu Y., Jindal V., Yen I., Bastani F. (2016). Integrating Clustering and Learning for improved Workload Prediction in the Cloud. *9th International Conference on Cloud Computing (CLOUD).*
[17] Schorgenhumer A., Kahlhofer M., Chalupar P. (2018). Using Multi-System Monitoring Time Series to Predict Performance Events. *9th Symposium on Software Performance (SSP).*
[18] Duenas J.C., Navarro J.M., Parada H.A., Andion J., Cuadrado F. (2018). Applying Event Stream Processing to Network Online Failure Prediction. *IEEE Communications Magazine.*
[19] Deljac Z., Randic M., Krcelic G. (2016). A Multivariate Approach to Predicting Quantity of Failures in Broadband Netowrks Based on a Recurrent Neural Network. *Journal of Network and Systems Management.*
[20] Akyamec A., Phadke C., Kushnir D., Uzunalioglu. (2015). Predicting Home Network problems using Diverse Data. *36th IEEE Sarnoff Symposium.*
[21] Asteriou D., Hall S. G. (2011). ARIMA Models and the Box-Jenkins Methodology, *Applied Econometrics.*
[22] Hyndman R.J., Athanasopoulos G. (2014). Forecasting: Principles and Practice.
[23] Isokawa T., Nishimora H., Matsui N. (2012). Quaternionic Multilayer Perceptron with local Analyticity. *Higher Dimensional Neural Networks: Quaternionic and Complex.*
[24] Rumelhart D. E., Hinton G.E., Williams R.J. (1986). Learning Internal Representations by Error Propagation. *Parallel distributed processing: explorations in the microstructure of cognition, Volume: 1.*
[25] Haykin S. (1999). Multilayer Perceptrons. *Neural Networks and Learning Machines.*
[26] Man K.F., Tang K.S., and Kwong S. (1996). Genetic Algorithms: Concepts and Applications. *IEEE Transactions on Industrial Electronics, Volume: 43, Issue: 5.*
[27] Kirkpatrick II C.D., Dahlquist J.R. (2016). Technical Analysis: The Complete Resource for Financial Market Technicians 3rd Edition.
[28] Rahman A., Thevaraja M., Gabriel M.E. (2019). Recent Developments in Data Science: Comparing Linear, Ridge and Lasso Regressions Techniques Using Wine Data. *The International Conference on Digital Image & Signal Processing.*
[29] Zhang C., Cheng X., Liu J., He J., Liu G. (2018). Deep Sparse Autoencoder for Feature Extraction and Diagnosis of Locomotive Adhesion Status. *Journal of Control Science and Engineering.*

Keyword Index

Author Index