Contributions to Statistics

Olga Valenzuela
Fernando Rojas
Héctor Pomares
Ignacio Rojas *Editors*

Theory and Applications of Time Series Analysis

Selected Contributions from ITISE 2018

Springer

Contributions to Statistics

The series **Contributions to Statistics** contains publications in theoretical and applied statistics, including for example applications in medical statistics, biometrics, econometrics and computational statistics. These publications are primarily monographs and multiple author works containing new research results, but conference and congress reports are also considered.

Apart from the contribution to scientific progress presented, it is a notable characteristic of the series that publishing time is very short, permitting authors and editors to present their results without delay.

More information about this series at http://www.springer.com/series/2912

Olga Valenzuela · Fernando Rojas ·
Héctor Pomares · Ignacio Rojas
Editors

Theory and Applications of Time Series Analysis

Selected Contributions from ITISE 2018

Springer

Editors
Olga Valenzuela
Faculty of Sciences
University of Granada
Granada, Spain

Fernando Rojas
ETSIIT, CITIC-UGR
University of Granada
Granada, Spain

Héctor Pomares
ETSIIT, CITIC-UGR
University of Granada
Granada, Spain

Ignacio Rojas
ETSIIT, CITIC-UGR
University of Granada
Granada, Spain

ISSN 1431-1968
Contributions to Statistics
ISBN 978-3-030-26035-4 ISBN 978-3-030-26036-1 (eBook)
https://doi.org/10.1007/978-3-030-26036-1

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The word forecasting is commonly associated by many people with some enigmatic concepts such as astrology (not to be confounded with astronomy), crystal balls, or tarot cards. There is, however, a different way to make forecasts based on scientific analyses from past and present information. This information can be numerical or categorical. It can even be expressed in linguistic terms by experts in the field. In this book, we try to provide some very recent contributions toward this scientific way to make forecasts and which have in common that this past and present information is arranged as a set of measurements collected in different instants of time (normally at fixed time intervals). These contributions to what we call in this book "Time Series Analysis and Forecasting" have been classified into different parts according to their content. The first three parts of the book contain more theoretical contributions, some related to pure statistical methods, some that also make use of state-of-the-art computational intelligence methodologies and finally some more related to econometrics. On the other hand, in the last parts, we provide more practical contributions with the intention of providing the readers with the view that this field, although with a very sophisticated and powerful theory behind, has as final aim the practical application. There exists practically no discipline in this world which cannot benefit from contributions in the "Time Analysis and Forecasting" field.

The origin of this book stems from the International Conference on Time Series and Forecasting, ITISE 2018, held in Granada (Spain) in September 2018. Our aim with the organization of ITISE 2018 was to create a friendly 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 statistics, mathematical models, econometrics, engineering, and computer science in the field of time series analysis and forecasting.

The list of topics in the successive Call for Papers has also evolved, resulting in the following list for the last 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.
    - Data preprocessing methods: data decomposition, seasonal adjustment, singular spectrum analysis, and detrending methods.
2. **Econometric and Forecasting**
    - Econometric models.
    - Economic and econometric forecasting.
    - Real macroeconomic monitoring and forecasting.
    - Advanced econometric methods.
3. **Advanced Methods and Online Learning in Time Series**
    - Adaptivity for stochastic models.
    - Online 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.
4. **High Dimension and Complex/Big Data**
    - Local versus global forecast.
    - Techniques for dimension reduction.
    - Multiscaling.
    - Forecasting from complex/big data.
5. **Forecasting in Real Problems**
    - Health forecasting.
    - Atmospheric science forecasting.
    - Telecommunication forecasting.
    - Hydrological forecasting.
    - Traffic forecasting.
    - Tourism forecasting.
    - Marketing forecasting.
    - Modeling and forecasting in power markets.

- Energy forecasting.
- Climate forecasting.
- Financial forecasting and risk analysis.
- Forecasting electricity load and prices.
- Forecasting and planning systems.
- Applications in other disciplines.

High-quality candidate papers from the Conference ITISE2018 (26 contributions) were invited to submit an extended version of their conference paper to be considered for this special publication in the book series of Springer: Contributions to Statistics. For the selection procedure, the information/evaluation of the chairman of every session, in conjunction with the review comments and the summary of reviews, were taken into account.

So, now we are pleased to have reached the end of the whole process and present the readers with these final contributions that we hope will provide a clear overview of the thematic areas covered by the ITISE 2018 conference, ranging from theoretical aspects to real-world applications of Time Series Analysis and Forecasting.

It is important to note that for the sake of consistency and readability of the book, the presented papers have been classified into the following parts:

- **Part: Advanced Statistical Methods for Time Series Analysis and Forecasting**
  The main objective of this chapter is to present advanced statistical methodologies and theories that could be used with time series. It also aims at bringing into existence recent and becoming developments in computational mathematics that could be used in the field of time series. In particular, six contributions have been selected for this chapter. The first contribution provides us with a methodology to identify nonstationary autoregressive processes with time-varying orders and time-varying degrees of nonstationarity, including its extension to multivariate autoregressive processes. The second contribution deals with how to take advantage of the information provided by different estimators for a given big data problem with inhomogeneities, i.e., data is neither i.i.d. (exhibiting outliers or not belonging to same distribution) nor stationary (time-varying effects may be present). Not surprisingly, the three most accurate forecasting methods of the recently celebrated M4 competition are precisely hybrid approaches formed by a combination of different estimators, thus proving that researching in this direction should be encouraged. The next contribution presents a new extension for the general case of linear and nonlinear data of the Granger causality technique to detect causal relationship between time series based on local approximations of the time delay embedding reconstruction of the time series' state space by a linear regression model. The next two contributions try to shed some light into how complex systems work. For this purpose, in the first one, the authors have developed a GUI-based computing environment which allows for building forecasts based on System Dynamics model, which is the part of the Systems Theory devoted to

understanding the dynamic behavior of complex systems. In the second one, the authors make use of order patterns recurrence plots to visually tell apart chaotic systems from other non-chaotic ones. Finally, the last contribution of this part presents a new freely available Matlab toolbox, called SSpace, that implements linear, nonlinear, and non-Gaussian State-Space systems. The contribution demonstrates the toolbox's potential with several examples.

- **Part: Advanced Computational Intelligence Methods for Time Series Analysis and Forecasting**
  Although time series analysis can be considered a discipline originated within the statistical area, in the last decades many computational intelligence methods or machine learning approaches have been proposed to solve time series-related problems. In fact, new and further computational intelligence approaches, their efficiency, and their comparison to statistical methods and other fact-checked computational intelligence methods are significant topics in academic and professional projects. It is not uncommon the existence of time series forecasting competitions which try to elucidate which of the two main research streams is better. For instance, the above-commented M4-Competition for the first time made explicit mention to machine learning forecasting methods. Within this topic, five contributions have been selected for this book. Just related to the comment we made in the previous paragraphs, the first of the contributions also deals with an ensemble of estimators but this time it is an ensemble of machine learning models (deep neural networks). In this case, the authors extend the concept of snapshot ensembles to the field of time series forecasting. The idea of this concept is to combine the models obtained through the different local minima that the optimization algorithm finds in its search for the global one. The next very interesting contribution is about detecting areas in a time series, stationary or nonstationary, where it can be asserted that the data belong to a different distribution than before. To solve this problem, the authors propose a method called Wavelet-Based Least Squares Density-Difference which is based on a least squares method applied to the distance between two wavelet expanded densities extracted from the time series. The third contribution of this second part of the book presents the very computationally efficient virtual leave-one-out methodology aimed at selecting the best neural network structure for time series prediction, and shows how to apply this method in the practical case of time series data extracted from crime-related police reports. Finally, the last contribution is related to how to implement existing algorithms in fast computing platforms such as Field Programmable Gate Arrays (FPGAs). In this case, the authors deal with the hardware implementation of Echo State Networks, which are a special case of Recurrent Neural Networks in which the synaptic weights between neurons are kept fixed and only the connections from the network to a measurement output layer are modified by learning.

- **Part: Econometric Models, Financial Forecasting, and Risk Analysis**
  One of the most prominent applications of time series modeling and forecasting lies within the field of Econometrics. This chapter aims at presenting some recent developments of time series research applied to financial and futures data with the original idea of focusing on studies that develop and apply recent nonlinear econometric models to reproduce financial market dynamics and to capture financial data properties with the hope of eventually predicting the next economic bubble. Five contributions have been selected to that end. The first one introduces a new class of long-memory model for the estimation of volatility of stock returns, which takes into account long-memory heteroskedasticity of the financial time series. The second contribution shows that under appropriate assumptions on the fractional integration orders the transfer function corresponding to a Vector Autoregressive Fractionally Integrated Moving Average (VARFIMA) or a Fractionally Integrated Vector Autoregressive Moving Average (FIVARMA) process can be estimated consistently using Canonical Variate Analysis. The third contribution studies how deep long short-term memory neural networks can be applied to robustly forecast interest rates of different maturities and tenors. Since deep networks need a lot of data to learn from, the authors solve this problem by generating data based on fitted time series models. They complete their presentation by applying support vector machines to predict trends in the term structures. The next contribution studies the default intensities estimated from credit default swap spreads by the dynamic Nelson–Siegel model with a time-varying decay parameter. They show that for the German and U.S. credit default swap markets the decay parameters change over time and the magnitude of the decay parameter is positively related to the level of default intensities. The last contribution of this part of the book deals with the problem of how to obtain an accurate measure of the current globalization process we are currently undergoing in the world. To that end, the authors use the concept of permutation entropy which essentially measures the entropy of a set of time series based on the analysis of their permutation patterns. The main difference between this concept and the Shannon entropy is that the former is a symbolic entropy focused on patterns rather than on a probability distribution function, which makes it useful in an analysis of short time series.

  The next three parts of the book are dedicated to specific applications of time series analysis. The contributions provided can be classified into the following main parts.

- **Part: Time Series Analysis in Earth Sciences**
  This part makes particular emphasis on the application of time series analysis applied to earth-sciences-related data. For example, the first contribution analyzes high-resolution time series from a long-term monitoring campaign in the Guadalquivir River Estuary in the south of Spain in order to predict how harmful floods can be under certain circumstances. The second contribution uses the smoothed Lomb–Scargle periodogram to study the precipitation and

temperature data recorded during the last decades from 707 meteorological stations in Andalusia. They eventually obtain a very interesting picture of the spatial distribution of the climatic cycles from where many conclusions can be extracted. The third contribution is about online forecasting of ambient temperature and solar irradiation. The method is based on an adaptive ARX model which can be tuned to changing weather conditions without relying on external inputs and it obtains an outstanding performance improvement with respect to the weather forecasting services' prediction. Finally, the last paper analyzes data taken from the storm that occurred at the Spanish coast of the Mediterranean Sea at the end of January 2017 and which produced severe coastal floods. The authors manage to accurately model and predict the space-time evolution of sea wave heights during that event using a combination of spatiotemporal random field theory and the Bayesian maximum entropy method.

- **Part: Energy Time Series Forecasting**
  This part makes particular emphasis on the application of time series analysis, modeling, and forecasting applied to energy-related data. By energy, we refer to any kind of energy, such as electrical, solar, microwave, wind, and so on. The first contribution presents several adaptive methods for forecasting solar heat production and heat demand of consumers based on weather forecasts. Apart from the good results obtained, these methods are explicitly developed so that they are easy to implement in simple computers such as programmable logic controllers commonly used in the industry. The second one deals with making short-term forecasts (48-h horizon) of wind power production so as to know how the pricing rates of wind-generated electricity are going to evolve. To that end, the authors propose several direct and indirect methods based on different machine learning algorithms with very promising results.
- **Part: Time Series Analysis and Prediction in Other Real Problems**
  This last part is dedicated to other real applications of time series analysis, modeling, and forecasting different from those especially mentioned before. The idea is to state explicitly that applications of time series analysis reach practically any scientific discipline imaginable. Four very different contributions were selected for this last part. The first one uses queuing theory to model the custom inspection process in the Helsinki–Vantaa Airport so as to predict its capacity and assess whether it can deal with the estimated increase in the number of passengers in the following years. The second one studies the use of different models such as ARIMA, Prophet (launched by Facebook in 2017) Multilayer Perceptrons, and Long Short-Term Memory Neural Networks to predict Internet data consumption and mobile phone card recharges for different prediction horizons. The results are worth reading. The next contribution studies how different pattern similarity-based forecasting methods perform to estimate up to 1 week ahead, the future demand of a thermal unit in a power plant. The following and last contribution of this book takes advantage of the fact that the ECG signal is somewhat modulated by the respiration of a person. So, with the

use of an Empirical Mode Decomposition approach to R-peak detection and an Independent Component Analysis to separate out the respiration signal in the frequency domain, the authors manage to robustly estimate the breathing rate of that person.

Last but not least, we would like to point out that this edition of ITISE was organized by the University of Granada together with the Spanish Chapter of the IEEE Computational Intelligence Society. The Guest Editors would also like to express their gratitude to all the people who supported them in the compilation of this book, and especially to the contributing authors for their submissions, the chairmen of the different sessions, and to the anonymous reviewers for their comments and useful suggestions in order to improve the quality of the papers.

We wish to thank our main sponsors as well: the Department of Computer Architecture and Computer Technology, the Faculty of Science of the University of Granada, the Research Centre for Information and Communications Technologies (CITIC-UGR), and the Ministry of Science and Innovation for their support and grants. Finally, we wish also to thank Prof. Alfred Hofmann, Vice President Publishing—Computer Science, Springer-Verlag and Dr. Veronika Rosteck, Springer, Editor, for their interest in editing a book series of Springer based on the best papers of ITISE 2018.

We hope the readers of this book can make the most of these selected contributions.

Granada, Spain
January 2019

Olga Valenzuela
Fernando Rojas
Héctor Pomares
Ignacio Rojas

Contents

Advanced Statistical Methods for Time Series Analysis and Forecasting

Advanced Computational Intelligence Methods for Time Series Analysis and Forecasting

Econometric Models, Financial Forecasting and Risk Analysis

Time Series Analysis in Earth Sciences

Energy Time Series Forecasting

Time Series Analysis and Prediction in Other Real Problems

Advanced Statistical Methods for Time Series Analysis and Forecasting

Identification of Nonstationary Processes Using Noncausal Bidirectional Lattice Filtering

Maciej Niedźwiecki and Damian Chojnacki

Abstract The problem of off-line identification of a nonstationary autoregressive process with a time-varying order and a time-varying degree of nonstationarity is considered and solved using the parallel estimation approach. The proposed parallel estimation scheme is made up of several bidirectional (noncausal) exponentially weighted lattice algorithms with different estimation memory and order settings. It is shown that optimization of both settings can be carried out by means of minimization of the locally evaluated accumulated forward/backward prediction error statistic.

Keywords Identification of nonstationary processes · Selection of model order · Selection of estimation memory

Introduction

Autoregressive analysis is a popular modeling tool, used to solve practical problems in many different areas, such as biomedicine [1–3], geophysics [4–6], telecommunications [7, 8], etc. When the analyzed processes are nonstationary, identification of their autoregressive models can be carried out using local estimation techniques, such as the well-known sliding-window (SWLS) or exponentially weighted (EWLS) least squares approaches. Local estimation algorithms are often called finite-memory since

This work was partially supported by the National Science Center under the agreement UMO-2015/17/B/ST7/03772. Calculations were carried out at the Academic Computer Centre in Gdańsk.

M. Niedźwiecki (✉) · D. Chojnacki
Faculty of Electronics, Telecommunications and Informatics, Department of Automatic Control, Gdańsk University of Technology, ul. Narutowicza 11/12, Gdańsk, Poland
e-mail: maciekn@eti.pg.edu.pl

D. Chojnacki
e-mail: damian.chojnacki@pg.edu.pl

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_1

they rely on the limited (or effectively limited) number of signal samples. Owing to this property they are capable of tracking time-varying signal parameters.

Two important decisions that must be taken when identifying the time-varying autoregressive model are the choice of the number of estimated autoregressive coefficients, i.e., the model order, and selection of the size of the local analysis interval, i.e., the estimation memory. Both decisions may have important quantitative (estimation accuracy) and qualitative (estimation adequacy) implications.

In this paper we will focus on noncausal estimation techniques, which can be applied when the analyzed signal is prerecorded and can be analyzed off-line. Noncausality means that at any given time instant t the local parameter estimates can be based on both "past" observations (collected prior to t) and "future" observations (collected after t). When applied to identification of nonstationary processes, noncausal estimators can significantly reduce the estimation bias (due to elimination of the so-called estimation delay, typical of all causal algorithms [9]).

In the proposed approach, which is a modification of the method described in [10], noncausal estimates are obtained by combining results yielded by the exponentially weighted least squares lattice/ladder algorithms [11] running forward and backward in time, respectively. The problem of model order and estimation memory adaptation is solved using the parallel estimation approach. In this approach several competing algorithms, with different order and memory settings, are operated simultaneously and compared according to their locally assessed predictive abilities.

The proposed technique is computationally attractive and yields time-varying models with guaranteed uniform stability property which is important in such applications as parametric spectrum estimation or process simulation.

Nonstationary Autoregressive Processes

Suppose that the analyzed discrete time signal $\{y(t)\}$, $t = \dots, -1, 0, 1, \dots$, can be described or at least approximated by the following time-varying autoregressive (AR) model

$$\begin{gathered} y(t) = \sum_{i=1}^{n} a_{i,n}(t) y(t-i) + e_n(t) = \boldsymbol{\varphi}_n^{\mathrm{T}}(t)\boldsymbol{\alpha}_n(t) + e_n(t) \\ \mathrm{var}[e_n(t)] = \rho_n(t) \end{gathered} \tag{1}$$

where $\boldsymbol{\varphi}_n(t) = [y(t-1), \dots, y(t-n)]^{\mathrm{T}}$ denotes regression vector, $\boldsymbol{\alpha}_n(t) = [a_{1,n}(t), \dots, a_{n,n}(t)]^{\mathrm{T}}$ denotes the vector of autoregressive coefficients, and $\{e_n(t)\}$ denotes white noise with a time-dependent variance $\rho_n(t)$. In the sequel we will assume that the entire history of the signal $\{y(t), t = 1, \dots, T_0\}$ is available, along with the "boundary" conditions $\{y(1-i), y(T_0+i), i = 1, \dots, N\}$, where N denotes the maximum model order that will be considered.

When the driving noise variance $\rho_n(t)$ is bounded, $\boldsymbol{\alpha}_n(t)$ is a “sampled” version of a sufficiently smooth continuous time parameter trajectory, and at all time instants t all zeros of the characteristic polynomial $A[z, \boldsymbol{\alpha}_n(t)] = 1 - \sum_{i=1}^{n} a_{i,n}(t)z^{-i}$ are uniformly bounded away from the unit circle in the complex plane, the process (1) is uniformly exponentially stable [12]. According to the theory developed by Dahlhaus [13], under the conditions specified above $\{y(t)\}$ belongs to the class of locally stationary processes with uniquely defined instantaneous spectral density function given by

$$S_n(\omega, t) = \frac{\rho_n(t)}{|A[e^{j\omega}, \boldsymbol{\alpha}_n(t)]|^2} \tag{2}$$

where $j = \sqrt{-1}$ and $\omega \in (-\pi, \pi]$ denotes the normalized angular frequency.

Equivalent Parametrizations of a Stationary Autoregressive Process

It is known that a zero-mean stationary AR process characterized by the set $\mathcal{P}_n = \{\rho_n, a_{1,n}, \dots, a_{n,n}\}$ (further referred to as direct parametrization) can be equivalently specified in terms of autocorrelation coefficients $\mathcal{R}_n = \{r_0, r_1, \dots, r_n\}$ where $r_i = \mathrm{E}[y(t)y(t-i)]$ (autocorrelation parametrization), or in terms of partial autocorrelation coefficients $\mathcal{Q}_n = \{r_0, q_1, \dots, q_n\}$ where q_i is the normalized autocorrelation between $y(t)$ and $y(t-i)$ with the linear dependence on the intermediate variables $y(s), t-i < s < t$ removed (lattice parametrization).

All three parametrizations are equivalent, i.e., given any of them, one can determine the remaining two using invertible mappings

$$\begin{aligned}
\mathcal{P}_n &= F[\mathcal{R}_n], \quad \mathcal{R}_n = F^{-1}[\mathcal{P}_n] \\
\mathcal{R}_n &= G[\mathcal{Q}_n], \quad \mathcal{Q}_n = G^{-1}[\mathcal{R}_n] \\
\mathcal{Q}_n &= H[\mathcal{P}_n], \quad \mathcal{P}_n = H^{-1}[\mathcal{Q}_n].
\end{aligned}$$

Description of these mappings can be found, e.g., in [14].

Causal Lattice Algorithm

The exponentially weighted least squares normalized lattice/ladder algorithm proposed by Lee et al. [11], further referred to as EWLMF algorithm, is a time- and order-recursive estimation procedure known for its low computational cost and numerical robustness. The EWLMF algorithm is a lattice approximation of the

EWLS algorithm. The EWLS algorithm, equipped with the forgetting constant λ_k, $0 < \lambda_k < 1$, provides a direct signal parametrization

$$\widehat{\mathcal{P}}_{n|k}(t) = \{\widehat{\rho}_{n|k}(t), \widehat{a}_{1,n|k}(t), \ldots, \widehat{a}_{n,n|k}(t)\}$$

where

$$\begin{aligned}\widehat{\boldsymbol{\alpha}}_{n|k}(t) &= [\widehat{a}_{1,n|k}(t), \ldots, \widehat{a}_{n,n|k}(t)]^{\mathrm{T}} \\ &= \arg\min_{\boldsymbol{\alpha}_n} \sum_{i=0}^{t-1} \lambda_k^i [y(t-i) - \boldsymbol{\varphi}_n^{\mathrm{T}}(t-i)\boldsymbol{\alpha}_n]^2 \end{aligned} \tag{3}$$

$$\widehat{\rho}_{n|k}(t) = \frac{1}{L_k(t)} \sum_{i=0}^{t-1} \lambda_k^i [y(t-i) - \boldsymbol{\varphi}_n^{\mathrm{T}}(t-i)\widehat{\boldsymbol{\alpha}}_{n|k}(t)]^2 \tag{4}$$

and $L_k(t) = \sum_{i=0}^{t-1} \lambda_k^i$ denotes the effective width of the applied exponential window. The explicit solution of (3) and (4) can be obtained in the form

$$\widehat{\boldsymbol{\alpha}}_{n|k}(t) = \widehat{\mathbf{R}}_{n|k}^{-1}(t)\widehat{\mathbf{r}}_{n|k}(t), \;\; \widehat{\rho}_{n|k}(t) = \widehat{r}_{0|k}(t) - \widehat{\mathbf{r}}_{n|k}^{\mathrm{T}}(t)\widehat{\boldsymbol{\alpha}}_{n|k}(t) \tag{5}$$

where

$$\begin{aligned}\widehat{\mathbf{R}}_{n|k}(t) &= \frac{1}{L_k(t)} \sum_{i=0}^{t-1} \lambda_k^i \boldsymbol{\varphi}_n(t-i)\boldsymbol{\varphi}_n^{\mathrm{T}}(t-i) \\ \widehat{\mathbf{r}}_{n|k}(t) &= \frac{1}{L_k(t)} \sum_{i=0}^{t-1} \lambda_k^i y(t-i)\boldsymbol{\varphi}_n(t-i) \\ \widehat{r}_{0|k}(t) &= \frac{1}{L_k(t)} \sum_{i=0}^{t-1} \lambda_k^i y^2(t-i) = \widetilde{r}_{0|k}(t).\end{aligned}$$

The EWLMF algorithm estimates the normalized partial autocorrelation coefficients directly from the data, yielding the lattice signal parametrization

$$\widetilde{\mathcal{Q}}_{n|k}(t) = \{\widetilde{r}_{0|k}(t), \widetilde{q}_{1|k}(t), \ldots, \widetilde{q}_{n|k}(t)\}$$

The estimates $\widetilde{q}_{1|k}(t), \ldots, \widetilde{q}_{n|k}(t)$ are usually called reflection coefficients. Due to appropriate normalization, the estimates provided by the EWLMF algorithm obey the condition

$$|\widetilde{q}_{i|k}(t)| < 1, \;\; \forall t, i = 1, \ldots, n \tag{6}$$

which guarantees that the corresponding “frozen” AR models are at all times stable. Denote by

$$\widetilde{\mathcal{P}}_{n|k}(t) = H^{-1}[\widetilde{\mathcal{Q}}_{n|k}(t)] = \{\widetilde{\rho}_{n|k}(t), \widetilde{a}_{1,n|k}(t), \ldots, \widetilde{a}_{n,n|k}(t)\}$$

the direct parametrization that is an equivalent of the lattice parametrization yielded by the EWLMF algorithm. Since the EWLS algorithm does not guarantee model stability, it is clear that $\widehat{\mathcal{P}}_{n|k}(t) \neq \widetilde{\mathcal{P}}_{n|k}(t)$. We note, however, that both parametrizations become identical if the matrix $\widehat{\mathbf{R}}_{n|k}(t)$ and the vector $\widehat{\mathbf{r}}_{n|k}(t)$ appearing in (5) are replaced with

$$\widetilde{\mathbf{R}}_{n|k}(t) = \begin{bmatrix} \widetilde{r}_{0|k}(t) & & \widetilde{r}_{n-1|k}(t) \\ \vdots & \ddots & \vdots \\ \widetilde{r}_{n-1|k}(t) & & \widetilde{r}_{0|k}(t) \end{bmatrix}, \quad \widetilde{\mathbf{r}}_{n|k}(t) = \quad \left[\widetilde{r}_{1|k}(t) \ldots \widetilde{r}_{n|k}(t)\right]^{\mathrm{T}}$$

where

$$\widetilde{\mathcal{R}}_{n|k}(t) = \{\widetilde{r}_{0|k}(t), \widetilde{r}_{1|k}(t), \ldots, \widetilde{r}_{n|k}(t)\} = G[\widetilde{\mathcal{Q}}_{n|k}(t)]$$

denotes an autocorrelation parametrization equivalent to $\widetilde{\mathcal{Q}}_{n|k}(t)$. Therefore, the parametrization $\widetilde{\mathcal{P}}_{n|k}(t)$ can be regarded as a stable approximation of $\widehat{\mathcal{P}}_{n|k}(t)$.

Noncausal Lattice Algorithm

To obtain noncausal estimator of $\rho_n(t)$ and $\boldsymbol{\alpha}_n(t)$ we will combine results yielded by two lattice algorithms—the forward-time $(-)$ EWLMF algorithm equipped with a forgetting constant λ_{k^-}, providing the estimates

$$\widetilde{\mathcal{Q}}^{-}_{n|k}(t) = \{\widetilde{r}_{0|k^-}(t), \widetilde{q}_{1|k^-}(t), \ldots, \widetilde{q}_{n|k^-}(t)\}$$

and the backward time $(+)$ EWLMF algorithm equipped with a forgetting constant λ_{k^+} providing the estimates

$$\widetilde{\mathcal{Q}}^{+}_{n|k}(t) = \{\widetilde{r}_{0|k^+}(t), \widetilde{q}_{1|k^+}(t), \ldots, \widetilde{q}_{n|k^+}(t)\}.$$

We will not assume that the forward and backward time EWLMF algorithms are equipped with the same forgetting constants. Setting $k^- \neq k^+$, one can fuse long memory forward time estimation results with short memory backward time ones or *vice versa*. Such asymmetric variants may be useful in the presence of abrupt parameter changes. Let $\pi = \{k^-, k^+\}$, $T_-(t) = \{1, \ldots, t-1\}$ and $T_+(t) = \{1, \ldots, T_0 - t\}$. The combined estimate can be obtained using a three-step procedure.

1. First, one can determine the autocorrelation parametrizations corresponding to $\widehat{\mathcal{Q}}^-_{n|k}(t-1)$ and $\widehat{\mathcal{Q}}^+_{n|k}(t+1)$

$$\widetilde{\mathcal{R}}^\pm_{n|k}(t\pm 1) = G[\widetilde{\mathcal{Q}}^\pm_{n|k}(t\pm 1)] = \{\widetilde{r}_{0|k^\pm}(t\pm 1), \widetilde{r}_{1|k^\pm}(t\pm 1), \ldots, \widetilde{r}_{n|k^\pm}(t\pm 1)\}$$

   Since parametrizations $\widetilde{\mathcal{Q}}^-_{n|k}(t-1)$ and $\widetilde{\mathcal{Q}}^+_{n|k}(t+1)$ are stable, the covariance matrices made up of the estimates $\{\widetilde{r}_{i|k^-}(t), i = 0, \ldots, n\}$ and $\{\widetilde{r}_{i|k^+}(t), i = 0, \ldots, n\}$ must be positive definite [14].
2. Second, the two-sided autocorrelation parametrization

$$\widetilde{\mathcal{R}}_{n|\pi}(t) = \{\widetilde{r}_{0|\pi}(t), \widetilde{r}_{1|\pi}(t), \ldots, \widetilde{r}_{n|\pi}(t)\}$$

   can be obtained using the formula

$$\widetilde{r}_{i|\pi}(t) = \mu_-(t)\widetilde{r}_{i|k^-}(t-1) + \mu_+(t)\widetilde{r}_{i|k^+}(t+1), \quad i = 0, \ldots, n \tag{7}$$

   where $\mu_\pm(t) = L^\pm_{k^\pm}(t\pm 1)/L_\pi(t)$, $L^\pm_{k^\pm}(t\pm 1) = \sum_{i\in T_\pm(t)} \lambda^{i-1}_{k^\pm}$ and $L_\pi(t) = L^-_{k^-}(t-1) + L^+_{k^+}(t+1)$. Note that since the sequence $\{\widetilde{r}_{i|\pi}(t), i = 0, \ldots, n\}$ is a convex combination of $\{\widetilde{r}_{i|k^-}(t-1), i = 0, \ldots, n\}$ and $\{\widetilde{r}_{i|k^+}(t+1), i = 0, \ldots, n\}$, the parametrization $\widetilde{\mathcal{R}}_{n|\pi}(t)$ is at all times stable.
3. Finally, based on $\widetilde{\mathcal{R}}_{n|\pi}(t)$, one can obtain the direct parametrization

$$\widetilde{\mathcal{P}}_{n|\pi}(t) = F[\widetilde{\mathcal{R}}_{n|\pi}(t)] = \{\widetilde{\rho}_{n|\pi}(t), \widetilde{a}_{1,n|\pi}(t), \ldots, \widetilde{a}_{n,n|\pi}(t)\}$$

The doubly exponentially weighted Lee-Morf-Friedlander (E^{2}WLMF) algorithm described above differs from the one proposed in [10] in one important aspect—unlike [10] the obtained parameter estimates do not depend (in a deterministic sense) on the "central" sample $y(t)$.

Similarly as in the case of the EWLMF estimate, one can show that the E^{2}WLMF estimate $\widetilde{\boldsymbol{\alpha}}_{n|\pi}(t) = [\widetilde{a}_{1,n|\pi}(t), \ldots, \widetilde{a}_{n,n|\pi}(t)]^{\mathrm{T}}$ can be regarded as a "stable approximation" of the estimate obtained using the noncausal doubly exponentially weighted least squares (E^{2}WLS) algorithm

$$\begin{aligned} \widehat{\boldsymbol{\alpha}}_{n|\pi}(t) &= [\widehat{a}_{1,n|\pi}(t), \ldots, \widehat{a}_{n,n|\pi}(t)]^{\mathrm{T}} \\ &= \arg\min_{\boldsymbol{\alpha}_n} \Big[\sum_{i=1}^{t-1} \lambda^{i-1}_{k^-}\{y(t-i) - [\boldsymbol{\varphi}^-_n(t-i)]^{\mathrm{T}}\boldsymbol{\alpha}_n\}^2 \\ &\qquad + \sum_{i=1}^{T_0-t} \lambda^{i-1}_{k^+}\{y(t+i) - [\boldsymbol{\varphi}^+_n(t+i)]^{\mathrm{T}}\boldsymbol{\alpha}_n\}^2 \Big] \end{aligned}$$

where $\boldsymbol{\varphi}^\pm_n(t) = [y(t\pm 1), \ldots, y(t\pm n)]^{\mathrm{T}}$. Actually, note that

$$\widehat{\boldsymbol{\alpha}}_{n|\pi}(t) = \Big[\mu_-(t)\widehat{\mathbf{R}}^-_{n|k^-}(t-1) + \mu_+(t)\widehat{\mathbf{R}}^+_{n|k^+}(t+1)\Big]^{-1} \\ \times \Big[\mu_-(t)\widehat{\mathbf{r}}^-_{n|k^-}(t-1) + \mu_+(t)\widehat{\mathbf{r}}^+_{n|k^+}(t+1)\Big] \tag{8}$$

where

$$\widehat{\mathbf{R}}^\pm_{n|k^\pm}(t\pm 1) = \frac{1}{L^\pm_{k^\pm}(t\pm 1)} \sum_{i\in T_\pm(t)} \lambda^{i-1}_{k^\pm} \boldsymbol{\varphi}^\pm_n(t\pm i)[\boldsymbol{\varphi}^\pm_n(t\pm i)]^{\mathrm{T}}$$

$$\widehat{\mathbf{r}}^\pm_{n|k^\pm}(t\pm 1) = \frac{1}{L^\pm_{k^\pm}(t\pm 1)} \sum_{i\in T_\pm(t)} \lambda^{i-1}_{k^\pm} y(t\pm i)\boldsymbol{\varphi}^\pm_n(t\pm i).$$

Similarly, since $\widetilde{\boldsymbol{\alpha}}_{n|\pi}(t)$ must obey Yule-Walker equations defined in terms of $\{\widetilde{r}_{i|\pi}(t), i = 0, \dots, n\}$ [14], it holds that

$$\widetilde{\boldsymbol{\alpha}}_{n|\pi}(t) = \Big[\mu_-(t)\widetilde{\mathbf{R}}^-_{n|k^-}(t-1) + \mu_+(t)\widetilde{\mathbf{R}}^+_{n|k^+}(t+1)\Big]^{-1} \\ \times \Big[\mu_-(t)\widetilde{\mathbf{r}}^-_{n|k^-}(t-1) + \mu_+(t)\widetilde{\mathbf{r}}^+_{n|k^+}(t+1)\Big]$$

where

$$\widetilde{\mathbf{R}}_{n|k^\pm}(t\pm 1) = \begin{bmatrix} \widetilde{r}_{0|k^\pm}(t\pm 1) & & \widetilde{r}_{n-1|k^\pm}(t\pm 1) \\ \vdots & \ddots & \vdots \\ \widetilde{r}_{n-1|k^\pm}(t\pm 1) & & \widetilde{r}_{0|k^\pm}(t\pm 1) \end{bmatrix}$$

$$\widetilde{\mathbf{r}}_{n|k^\pm}(t\pm 1) = \Big[\widetilde{r}_{1|k^\pm}(t\pm 1) \dots \widetilde{r}_{n|k^\pm}(t\pm 1)\Big]^{\mathrm{T}}.$$

Hence, the estimates $\widehat{\boldsymbol{\alpha}}_{n|\pi}(t)$ and $\widetilde{\boldsymbol{\alpha}}_{n|\pi}(t)$ coincide if the quantities $\widehat{\mathbf{R}}^\pm_{n|k^\pm}(t\pm 1)$ and $\widehat{\mathbf{r}}^\pm_{n|k^\pm}(t\pm 1)$ are replaced in (8) with $\widetilde{\mathbf{R}}^\pm_{n|k^\pm}(t\pm 1)$ and $\widetilde{\mathbf{r}}^\pm_{n|k^\pm}(t\pm 1)$, respectively.

Model Order and Estimation Memory Adaptation

Based on $\widetilde{\mathcal{P}}_{n|\pi}(t)$, the parametric estimate of the instantaneous spectral density function $S_n(\omega, t)$ can be obtained in the form

$$\widetilde{S}_{n|\pi}(\omega, t) = \frac{\widetilde{\rho}_{n|\pi}(t)}{|A[e^{j\omega}, \widetilde{\boldsymbol{\alpha}}_{n|\pi}(t)]|^2} \tag{9}$$

where $\widetilde{\boldsymbol{\alpha}}_{n|\pi}(t) = [\widetilde{a}_{1,n|\pi}(t), \dots, \widetilde{a}_{n,n|\pi}(t)]^{\mathrm{T}}$.

Selection of the order n of the autoregressive model, and the choice of forgetting factors $\lambda_{k^\pm}$ plays an important role in parametric spectral analysis. If the order is underestimated some important features of the resonant structure of $\{y(t)\}$ may not be revealed, while when it is overestimated some nonexistent resonances may be indicated. In both cases one may arrive at false qualitative conclusions. The optimal choice of λ_{k^-} and λ_{k^+}, i.e., the one that trades off the bias and variance components of the mean squared parameter estimation error, depends on the rate of parameter variation—forgetting factors should be smaller (which corresponds to shorter memory) when process parameters are subject to fast changes, and larger (which corresponds to longer memory) when parameters vary slowly with time.

Our solution to the order/memory optimization problem will be based on parallel estimation. Consider several E^{2}WLMF algorithms with different order and memory settings, running in parallel. Denote by $\mathcal{N} = \{1, \dots, N\}$ the set of all model orders that will be considered, and by Π the set of all considered pairs $\pi = \{k_-, k_+\}$. The data-adaptive version of (9) can be expressed in the form

$$\widetilde{S}_{\widehat{n}(t)|\widehat{\pi}(t)}(\omega, t) = \frac{\widetilde{\rho}_{\widehat{n}(t)|\widehat{\pi}(t)}(t)}{|A[e^{j\omega}, \widetilde{\boldsymbol{\alpha}}_{\widehat{n}(t)|\widehat{\pi}(t)}(t)]|^2} \tag{10}$$

where

$$\{\widehat{n}(t), \widehat{\pi}(t)\} = \{\widehat{n}(t), \widehat{k}_-(t), \widehat{k}_+(t)\} = \arg\min_{\substack{n\in\mathcal{N}\\ \pi\in\Pi}} J_{n|\pi}(t) \tag{11}$$

and $J_{n|\pi}(t)$ denotes the local decision statistic.

The proposed selection criterion takes advantage of the fact that, unlike the estimates considered in [10], the estimates $\widetilde{\boldsymbol{\alpha}}_{n|\pi}(t)$ are not functions of $y(t)$ and therefore they can be used to compute unbiased forward and backward prediction errors

$$\varepsilon^{\pm}_{n|\pi}(t) = y(t) - [\boldsymbol{\varphi}^{\pm}_n(t)]^{\mathrm{T}} \widetilde{\boldsymbol{\alpha}}_{n|\pi}(t).$$

Consequently, one can adopt for $J_{n|\pi}(t)$ the following prediction error (PE) statistic

$$J_{n|\pi}(t) = \sum_{i=-M}^{M} [\varepsilon^{-}_{n|\pi}(t-i)]^2 + \sum_{i=-M}^{M} [\varepsilon^{+}_{n|\pi}(t+i)]^2 \tag{12}$$

where $M \in [20, 50]$ is the parameter that controls the size of the local decision window $[t-M, t+M]$ centered around t.

Computational Complexity

Denote by $K_\pi \le K(K+1)/2$ the number of forward–backward pairs $\pi = (k^-, k^+)$ included in Π. For the assumed maximum model order N the per sample computational load (the number of multiply–add operations) of the proposed parallel estimation scheme is pretty low and is approximately equal to

$$l(N) = 2KA(N) + 2KB(N) + K_\pi C(N)$$

where $A(N) = 30N$ denotes the load of the ELMF algorithm (given that the Newton-Raphson method is used to evaluate square roots), $B(N) = 2N + N^2$ denotes the load of the G transform (computation of autocorrelation coefficients based on reflection coefficients), and $C(N) = 2 + 4N + N^2$ is the load of the F transform (computation of autoregressive coefficients based on autocorrelation coefficients). Note that the first stage of processing is computationally the cheapest one and that the only quantities that have to be memorized during the forward/backward sweep of the EWLMF algorithms are the forward/backward reflection coefficients.

Extension to Multivariate Autoregressive Processes

Unlike the univariate case, every zero-mean stationary multivariate AR process has two (usually different) direct parametrizations $\mathcal{P}_n^- = \{\boldsymbol{\rho}_n^-, \mathbf{A}_{1,n}^-, \dots, \mathbf{A}_{n,n}^-\}$ and $\mathcal{P}_n^+ = \{\boldsymbol{\rho}_n^+, \mathbf{A}_{1,n}^+, \dots, \mathbf{A}_{n,n}^+\}$, corresponding to the forward-time and backward time AR models, respectively:

$$\mathbf{y}(t) = \sum_{i=1}^{n} \mathbf{A}_{i,n}^{\pm}\mathbf{y}(t \pm i) + \mathbf{e}_n^{\pm}(t), \quad \mathrm{cov}[\mathbf{e}_n^{\pm}(t)] = \boldsymbol{\rho}_n^{\pm} \tag{13}$$

where $\mathbf{y}(t) = [y_1(t), \dots, y_m(t)]^{\mathrm{T}}$ is the m-dimensional vector of signal components and $\mathbf{A}_{i,n}^{\pm}$, $i = 1, \dots, n$, denote $m \times m$ matrices of forward/backward autoregressive coefficients. Based on (13), the spectral density (matrix) function of $\{\mathbf{y}(t)\}$ can be evaluated using the formula

$$\begin{aligned}\mathbf{S}_n(\omega) &= \left[\mathcal{A}\left(e^{-j\omega}, \boldsymbol{\alpha}_n^-\right)\right]^{-1} \boldsymbol{\rho}_n^- \left[\mathcal{A}\left(e^{j\omega}, \boldsymbol{\alpha}_n^-\right)\right]^{-\mathrm{T}} \\ &= \left[\mathcal{A}\left(e^{-j\omega}, \boldsymbol{\alpha}_n^+\right)\right]^{-1} \boldsymbol{\rho}_n^+ \left[\mathcal{A}\left(e^{j\omega}, \boldsymbol{\alpha}_n^+\right)\right]^{-\mathrm{T}}\end{aligned} \tag{14}$$

where $\boldsymbol{\alpha}_n^{\pm} = [\mathbf{A}_{1,n}^{\pm}, \dots, \mathbf{A}_{n,n}^{\pm}]$ and

$$\mathcal{A}\left(z, \boldsymbol{\alpha}_n^{\pm}\right) = \mathrm{i} - \sum_{i=1}^{n} \mathbf{A}_{i,n}^{\pm} z^{-i}.$$

Similar to the univariate case, the process $\{\mathbf{y}(t)\}$ has a unique autocorrelation parametrization

$$\mathcal{R}_n = \{\mathbf{R}_0, \mathbf{R}_1, \dots, \mathbf{R}_n\}$$

where $\mathbf{R}_i = \mathrm{E}[\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t-i)]$, and unique lattice parametrization

$$\mathcal{Q}_n = \{\mathbf{R}_0, \mathbf{Q}_1, \dots, \mathbf{Q}_n\}$$

where $\mathbf{Q}_i$, $i = 1, \ldots, n$, denote the matrices of normalized partial autocorrelation (reflection) coefficients.

Noncausal identification of a multivariate AR process can be carried out in an analogous way to that described in section "Noncausal Lattice Algorithm". First, at each time instant t, one can use the multivariate version of the EWLMF algorithm to evaluate the matrices of forward-time and backward-time normalized reflection coefficients $\widetilde{\mathbf{Q}}^{\pm}_{i|k}(t \pm 1)$, $i = 1, \ldots, N$, $k \in \mathcal{K}$. Then, for all selections of n and π, one can evaluate the two-sided parametrizations $\widetilde{\mathcal{R}}_{n|\pi}(t)$, $n \in \mathcal{N}$, $\pi \in \Pi$ and, after solving the corresponding Yule-Walker equations—the two-sided direct parametrizations

$$\mathcal{P}^{\pm}_{n|\pi}(t) = \{\widetilde{\boldsymbol{\rho}}^{\pm}_{n|\pi}(t), \widetilde{\mathbf{A}}^{\pm}_{1,n|\pi}(t), \ldots, \widetilde{\mathbf{A}}^{\pm}_{n,n|\pi}(t)\}.$$

Finally, the best local combination of n and π can be selected using the decision rule (11) after adopting

$$J_{n|\pi}(t) = \det\left[E_{n|\pi}(t)\right] \tag{15}$$

$$E_{n|\pi}(t) = \sum_{i=-M}^{M} \boldsymbol{\varepsilon}^{-}_{n|\pi}(t-i)[\boldsymbol{\varepsilon}^{-}_{n|\pi}(t-i)]^{\mathrm{T}} + \sum_{i=-M}^{M} \boldsymbol{\varepsilon}^{+}_{n|\pi}(t-i)[\boldsymbol{\varepsilon}^{+}_{n|\pi}(t-i)]^{\mathrm{T}}$$

where

$$\boldsymbol{\varepsilon}^{\pm}_{n|\pi}(t) = \mathbf{y}(t) - \sum_{i=1}^{n} \widetilde{\mathbf{A}}^{\pm}_{i,n|\pi}(t)\mathbf{y}(t \pm i).$$

Note that (15) is a natural extension of the univariate statistic (12). Another option (computationally less demanding) is choosing $J_{n|\pi}(t)$ in the form

$$J_{n|\pi}(t) = \mathrm{tr}\left[E_{n|\pi}(t)\right]. \tag{16}$$

Once the quantities $\widehat{n}(t)$ and $\widehat{\pi}(t)$ are established, the estimate of the instantaneous spectral density matrix can be obtained in the form

$$\begin{aligned}\widetilde{\mathbf{S}}_{\widehat{n}(t)|\widehat{\pi}(t)}(\omega, t) = \left\{\mathcal{A}\left[e^{-j\omega}, \widetilde{\boldsymbol{\alpha}}^{\pm}_{\widehat{n}(t)|\widehat{\pi}(t)}(t)\right]\right\}^{-1} \widetilde{\boldsymbol{\rho}}^{\pm}_{\widehat{n}(t)|\widehat{\pi}(t)}(t) \times \\ \times \left\{\mathcal{A}\left[e^{j\omega}, \widetilde{\boldsymbol{\alpha}}^{\pm}_{\widehat{n}(t)|\widehat{\pi}(t)}(t)\right]\right\}^{-\mathrm{T}}\end{aligned} \tag{17}$$

where

$$\widetilde{\boldsymbol{\alpha}}^{\pm}_{n|\pi}(t) = [\widetilde{\mathbf{A}}^{\pm}_{1,n|\pi}(t), \ldots, \widetilde{\mathbf{A}}^{\pm}_{n,n|\pi}(t)].$$

Simulation Results

To verify the proposed order and estimation memory selection rule, a nonstationary variable-order autoregressive process was generated. Process generation was based on four time-invariant AR anchor models M_1, M_2, M_3 and M_4, of orders 2, 4, 6 and 8, respectively. The characteristic polynomial $A_i(z)$ of the model M_i had i pairs of complex conjugate zeros, given by $z_k^{\pm} = 0.995e^{\pm jk\pi/5}$, $k = 1, \dots, i$. Two simulation scenarios were considered, incorporating either smooth parameter changes (scenario A) or abrupt parameter changes (scenario B).

In the first case, depicted in Fig. 1, the generated signal $\{y(t), t = 1, \dots, T_0\}$ had stationary periods, during which it was governed by anchor models, and nonstationary periods, when the generating model was obtained by morphing one anchor model into another one. Transition from M_{i-1} to M_i was realized by moving, with a constant speed, the i-th pair of complex conjugate zeros from their initial zero positions

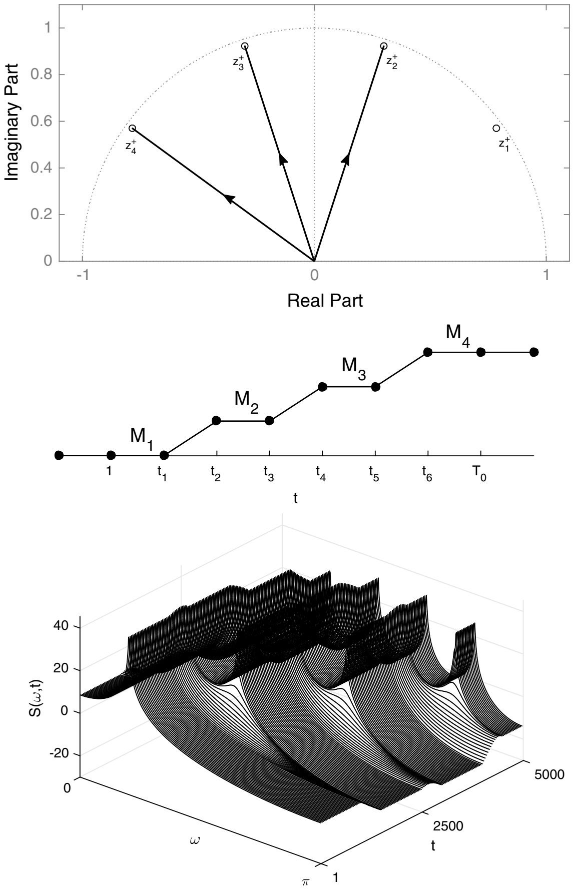


Fig. 1 Trajectories of zeros of the characteristic polynomial (top figure), simulation scenario A corresponding to smooth parameter variation (middle figure), and the corresponding time-varying spectral density function (bottom figure)

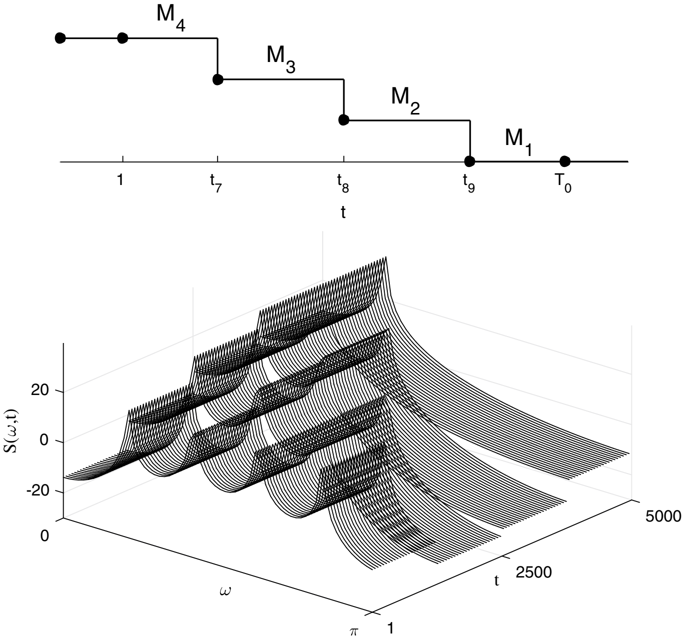


Fig. 2 Simulation scenario B corresponding to abrupt parameter changes (top figure), and the corresponding time-varying spectral density function (bottom figure)

towards the unit circle—see Fig. 1. The simulation scenario is symbolically depicted in Fig. 1. Note that according to this scenario the order of the generating model was gradually increased from two to eight.

In the second case, illustrated in Fig. 2, the model M_{i-1} was instantaneously switched to M_i. In this case the order of the model was gradually decreased from eight to two.

The adopted value of T_0 was equal to 5000 and the breakpoints, marked with bullets in Figs. 1 and 2, had the following time coordinates: $t_1 = 1000$, $t_2 = 1500$, $t_3 = 2500$, $t_4 = 3000$, $t_5 = 4000$, $t_6 = 4500$ (for type-A changes), and $t_7 = 1250$, $t_8 = 2750$, $t_9 = 4250$ (for type-B changes). The parallel estimation scheme was made up of 4 E^{2}WLMF algorithms combining results yielded by $K = 3$ forward/backward EWLMF trackers equipped with forgetting constants $\lambda_1 = 0.95$, $\lambda_2 = 0.99$ and $\lambda_3 = 0.995$. The four combinations of forward/backward forgetting constants were: $(0.99, 0.99)$, $(0.995, 0.995)$, $(0.995, 0.95)$, and $(0.95, 0.995)$, which corresponds to $\pi_1 = (2, 2)$, $\pi_2 = (3, 3)$, $\pi_3 = (3, 1)$ and $\pi_4 = (1, 3)$, respectively. The parameter M, which determines the width of the local decision window, was set to 50.

Two measures of fit were used to evaluate identification results: the mean squared parameter tracking error and the Itakura-Saito spectral distortion measure, both averaged over $t \in [1, T_0]$ and 100 independent realizations of $\{y(t)\}$. Table 1 compares results yielded by three unidirectional $(\lambda_1, \ldots, \lambda_3)$ and four bidirectional

Table 1 Averaged Itakura-Saito distortion measures (left tables) and mean square parameter estimation errors (right tables) in two cases described in the text.

Smooth parameter variation

n/N	λ_1	λ_2	λ_3	π_1	π_2	π_3	π_4	Adaptive
1	4,600	4,266	4,199	4,185	4,131	4,193	4,170	4,155
2	3,183	2,751	2,796	2,551	2,552	2,697	2,644	2,603
3	3,093	2,619	2,660	2,398	2,397	2,559	2,488	2,446
4	2,092	1,536	1,616	1,318	1,358	1,482	1,483	1,357
5	2,169	1,536	1,611	1,298	1,333	1,472	1,464	1,338
6	1,118	0,577	0,711	0,452	0,586	0,563	0,726	0,437
7	1,180	0,583	0,697	0,416	0,519	0,558	0,629	0,415
8	0,775	0,144	0,208	0,070	0,163	0,126	0,239	**0,067**
9	0,848	0,147	0,187	0,071	0,146	0,117	0,206	**0,068**
10	0,925	0,154	0,189	0,072	0,134	0,120	0,192	**0,068**
11	1,006	0,160	0,191	0,073	0,125	0,122	0,183	**0,069**
12	1,093	0,167	0,194	0,075	0,121	0,124	0,180	**0,069**
13	1,187	0,174	0,196	0,077	0,120	0,126	0,182	**0,069**
14	1,301	0,181	0,198	0,077	0,116	0,129	0,179	**0,069**
15	1,413	0,190	0,204	0,080	0,114	0,132	0,178	**0,070**
16	1,547	0,198	0,208	0,082	0,112	0,135	0,178	**0,070**
17	1,674	0,206	0,213	0,084	0,111	0,138	0,179	**0,071**
18	1,821	0,214	0,218	0,086	0,112	0,142	0,182	**0,071**
19	1,952	0,221	0,222	0,088	0,111	0,145	0,183	**0,071**
20	2,095	0,231	0,227	0,091	0,111	0,148	0,185	**0,072**

n/N	λ_1	λ_2	λ_3	π_1	π_2	π_3	π_4	Adaptive
1	12,027	11,986	11,956	12,002	11,992	11,959	12,049	12,010
2	8,673	8,623	8,577	8,679	8,685	8,587	8,789	8,681
3	6,504	6,386	6,339	6,418	6,444	6,339	6,615	6,484
4	3,015	2,897	2,839	3,011	3,291	2,829	3,729	2,984
5	2,628	2,460	2,566	2,315	2,517	2,514	2,720	2,402
6	1,106	1,026	1,282	0,593	0,863	1,156	1,113	0,815
7	1,083	0,547	0,675	0,723	2,412	0,596	2,970	0,478
8	1,102	0,348	0,441	0,369	1,522	0,352	2,125	**0,236**
9	1,392	0,425	0,523	0,389	1,284	0,436	1,718	**0,252**
10	1,688	0,485	0,552	0,444	1,347	0,464	1,749	**0,266**
11	2,014	0,551	0,578	0,497	1,449	0,486	1,852	**0,278**
12	2,340	0,628	0,618	0,560	1,680	0,523	2,135	**0,294**
13	2,613	0,690	0,646	0,549	1,560	0,546	2,067	**0,298**
14	2,947	0,761	0,682	0,557	1,377	0,580	1,910	**0,305**
15	3,197	0,827	0,716	0,579	1,275	0,613	1,801	**0,312**
16	3,614	0,904	0,756	0,615	1,245	0,646	1,778	**0,322**
17	3,887	0,962	0,784	0,653	1,306	0,672	1,855	**0,328**
18	4,184	1,026	0,816	0,673	1,302	0,702	1,868	**0,333**
19	4,475	1,085	0,847	0,691	1,259	0,731	1,841	**0,339**
20	4,805	1,160	0,885	0,723	1,223	0,766	1,820	**0,345**

Abrupt parameter changes

n/N	λ_1	λ_2	λ_3	π_1	π_2	π_3	π_4	Adaptive
1	4.164	4.116	4.414	4.095	4.053	4.105	4.096	4.082
2	3.206	3.215	3.680	3.050	3.049	3.185	3.097	3.052
3	3.078	3.075	3.621	2.915	2.905	3.044	2.966	2.909
4	1.993	2.071	2.597	1.187	1.872	2.035	1.865	1.762
5	1.989	2.044	2.708	1.796	1.836	2.004	1.851	1.748
6	1.116	1.299	1.699	0.946	1.098	1.264	0.939	0.809
7	1.072	1.179	1.841	0.908	1.004	1.144	0.926	0.818
8	0.281	0.372	0.870	0.180	0.298	0.359	0.184	**0.082**
9	0.218	0.300	0.940	0.154	0.241	0.288	0.148	**0.076**
10	0.209	0.270	1.014	0.143	0.212	0.258	0.144	**0.074**
11	0.207	0.255	1.103	0.138	0.195	0.242	0.145	**0.074**
12	0.210	0.250	1.191	0.137	0.188	0.237	0.148	**0.075**
13	0.216	0.253	1.283	0.139	0.188	0.239	0.151	**0.075**
14	0.220	0.251	1.401	0.139	0.184	0.236	0.153	**0.075**
15	0.225	0.251	1.511	0.140	0.181	0.235	0.156	**0.076**
16	0.230	0.251	1.656	0.141	0.179	0.234	0.160	**0.076**
17	0.236	0.251	1.785	0.142	0.177	0.235	0.164	**0.077**
18	0.243	0.254	1.962	0.144	0.178	0.238	0.167	**0.077**
19	0.249	0.256	2.098	0.146	0.178	0.239	0.170	**0.077**
20	0.255	0.259	2.242	0.148	0.178	0.241	0.174	**0.078**

n/N	λ_1	λ_2	λ_3	π_1	π_2	π_3	π_4	Adaptive
1	13.808	13.828	13.806	13.764	13.754	13.819	13.718	13.771
2	10.619	10.684	10.571	10.542	10.546	10.667	10.446	10.554
3	8.346	8.549	8.246	8.189	8.271	8.512	8.011	8.203
4	4.934	5.732	4.306	4.589	5.084	5.649	3.959	4.205
5	3.729	4.371	3.420	3.497	3.969	4.283	3.320	4.205
6	1.620	2.658	1.082	1.248	1.966	2.537	0.872	0.620
7	3.175	5.830	1.718	2.739	4.927	5.693	1.110	0.772
8	2.175	4.048	1.428	1.732	3.247	3.984	0.514	**0.274**
9	1.785	3.114	1.677	1.385	2.505	3.013	0.629	**0.280**
10	1.829	3.125	2.005	1.411	2.501	3.013	0.713	**0.298**
11	1.973	3.285	2.402	1.505	2.646	3.164	0.768	**0.313**
12	2.259	3.971	2.779	1.745	3.126	3.692	0.836	**0.328**
13	2.327	3.792	3.110	1.770	3.081	3.698	0.866	**0.341**
14	2.244	3.453	3.477	1.645	2.720	3.365	0.884	**0.352**
15	2.191	3.215	3.716	1.561	2.469	3.099	0.947	**0.358**
16	2.219	3.120	4.175	1.547	2.370	2.984	0.937	**0.370**
17	2.303	3.218	4.451	1.602	2.456	3.089	0.947	**0.380**
18	2.379	3.260	4.781	1.640	2.475	3.121	1.009	**0.387**
19	2.419	3.221	5.112	1.644	2.417	3.082	1.039	**0.393**
20	2.467	3.185	5.459	1.654	2.359	3.044	1.070	**0.399**

$(\pi_1, \ldots, \pi_4)$ lattice algorithms (for different values of n), with the results yielded by the proposed adaptive scheme (for different values of N). Note that when the model order is not underestimated ($n, N \geq 8$) the algorithm with adaptive order and memory assignment provides results that are uniformly the best, irrespective of the choice of N.

Our second example shows the result of application of the proposed approach to analysis of a real signal. Fig. 3 shows the plots of five fragments of a speech signal (sampled at the rate of 22.05 kHz) and the corresponding estimates of the time-varying spectrum obtained using the parallel estimation scheme described above (with the same settings).

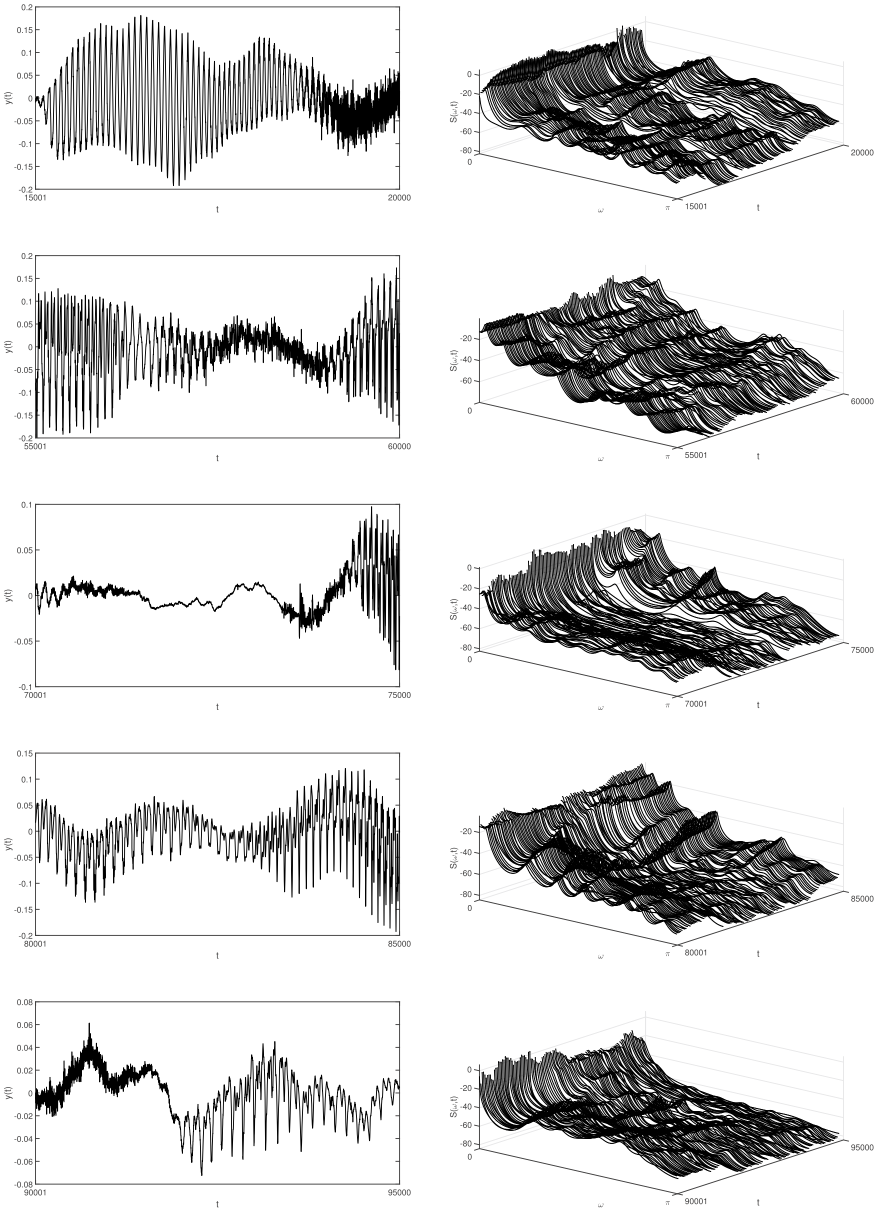


Fig. 3 Five fragments of a speech signal (left figures) and the estimated time-varying spectra (right figures)

Conclusion

A new noncausal (bidirectional) lattice filtering algorithm was designed for off-line identification of nonstationary autoregressive processes and an adaptive mechanism was proposed for dynamic selection of the number of estimated coefficients and the most appropriate estimation memory, matching the degree of process nonstationarity. It was shown that the proposed adaptive parallel estimation scheme outperforms the fixed-order fixed-memory algorithms it is made up of.

References

1. Wada, T., Jinnouchi, M., Matsumura, Y.: Application of autoregressive modelling for the analysis of clinical and other biological data. Ann. Inst. Statist. Math. **40**, 211–227 (1998)
2. Jirsa, V.K., McIntosh, A.R. (Eds.): Handbook of Brain Connectivity. Springer (2007)
3. Takalo, R., Hytti, H., Ihalainen, H., Sohlberg, A.: Adaptive autoregressive model for reduction of noise in SPECT. Comp. Math. Methods in Medicine **2015**(9) (2015)
4. Li, C., Nowack, R.L.: Application of autoregressive extrapolation to seismic tomography. Bull. Seism. Soc. Amer. **94**, 1456–1466 (2004)
5. Lesage, P., Glangeaud, F., Mars, J.: Applications of autoregressive models and time-frequency analysis to the study of volcanic tremor and long-period events. J. Volc. Geotherm. Res. **114**, 391–417 (2002)
6. Brillinger, D., Robinson, E.A., Schoenberg, F.P. (Eds.): Time Series Analysis and Applications to Geophysical Systems. Springer (2012)
7. Baddour, K.E., Beaulieu, N.C.: Autoregressive models for fading channel simulation. IEEE Trans. Wirel. Comm. **4**, 1650–1662 (2005)
8. Hayes, J.F., Ganesh Babu, T.V.J.: Modeling and Analysis of Telecommunication Networks. Wiley (2004)
9. Niedźwiecki, M.: Identification of Time-Varying Processes. Wiley (2000)
10. Niedźwiecki, M., Meller, M., Chojnacki, D.: Lattice filter based autoregressive spectrum estimation with joint model order and estimation bandwidth adaptation. In: Proceedings 56th IEEE Conference on Decision and Control, Melbourne, Australia, pp. 4618–4625 (2017)
11. Lee, D.T.L., Morf, M., Friedlander, B.: Recursive least squares ladder estimation algorithms. IEEE Trans. Acoust. Speech Signal Process **29**, 627–641 (1981)
12. Moulines, E., Priouret, P., Roueff, F.: On recursive estimation for time-varying autoregressive processes. Ann. Statist. **33**, 2610–2654 (2005)
13. Dahlhaus, R.: Locally stationary processes. Handbook Statist. **25**, 1–37 (2012)
14. Friedlander, B.: Lattice filters for adaptive processing. Proc. IEEE **70**, 829–867 (1982)

Normalized Entropy Aggregation for Inhomogeneous Large-Scale Data

Maria da Conceição Costa and Pedro Macedo

Abstract It was already in the fifties of the last century that the relationship between information theory, statistics and maximum entropy was established, following the works of Kullback, Leibler, Lindley and Jaynes. However, the applications were restricted to very specific domains and it was not until recently that the convergence between information processing, data analysis and inference demanded the foundation of a new scientific area, commonly referred to as Info-Metrics [1, 2]. As a huge amount of information and large-scale data have become available, the term "big data" has been used to refer to the many kinds of challenges presented in its analysis: many observations, many variables (or both), limited computational resources, different time regimes or multiple sources. In this work, we consider one particular aspect of big data analysis which is the presence of inhomogeneities, compromising the use of the classical framework in regression modelling. A new approach is proposed, based on the introduction of the concepts of info-metrics to the analysis of inhomogeneous large-scale data. The framework of information-theoretic estimation methods is presented, along with some information measures. In particular, the normalized entropy is tested in aggregation procedures and some simulation results are presented.

Keywords Big data · Info-Metrics · Maximum entropy · Normalized entropy

M. da Conceição Costa (✉) · P. Macedo
Department of Mathematics and CIDMA – Center for Research
and Development in Mathematics and Applications, University of Aveiro,
3810-193 Aveiro, Portugal
e-mail: lopescosta@ua.pt
URL: http://www.ua.pt

P. Macedo
e-mail: pmacedo@ua.pt

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_2

Introduction

Inference and processing of limited information are still one of the most fascinating universal problems. As stated by Golan [2], a very recent publication, "[...] the available information is most often insufficient to provide a unique answer or solution for most interesting decisions or inferences we wish to make. In fact, insufficient information—including limited, incomplete, complex, noisy and uncertain information—is the norm for most problems across all disciplines." Also, regardless of the system or question studied, any researcher observes only a certain amount of information or evidence and optimal inference must take into account the relationship between the observable and the unobservable, [3].

Info-Metrics is a constrained optimization framework for information processing, modelling and inference with finite, noisy or incomplete information. It is at the intersection of information theory, statistical methods of inference, applied mathematics, computer science, econometrics, complexity theory, decision analysis, modelling and the philosophy of science, [2].

As Info-Metrics generalizes the Maximum Entropy (ME) principle by Jaynes [4, 5], which in turn relies on the maximization of Shannon's entropy, the notions of information, uncertainty and entropy are fundamental to the understanding of the methodologies involved. Each scientist and discipline have their own interpretation and definition of information within the context of their research and understanding but, in the context of Info-Metrics, it refers to the meaningful content of data, it's context and interpretation and how to transfer data from one entity to another. As for uncertainty, it arises from a proposition or a set of possible outcomes where none of the choices or outcomes is known with certainty (a proposition is uncertain if it is consistent with knowledge but not implied by knowledge). Therefore, these outcomes are represented by a certain probability distribution. The more uniform the distribution, the higher the uncertainty that is associated with this set of propositions or outcomes. Finally, the concept of entropy reflects what, on average, we expect to learn from observations and it depends on how we measure information. Technically, entropy is a measure of uncertainty of a single random variable. As such, entropy can be viewed as a measure of uniformity.

For a brief discussion of entropy, let us consider the set $\boldsymbol{A} = \{a_1, a_2, \ldots, a_K\}$ to be a finite set and $\boldsymbol{p}$ a proper probability mass function on $\boldsymbol{A}$. The amount of information needed to fully characterize all of the elements of this set consisting of K discrete elements is defined by the Hartley's formula, $I(\boldsymbol{A}_K) = \log_2 K$. Shannon's information content of an outcome a_k is $h(a_k) = h(p_k) \equiv \log_2 \frac{1}{p_k}$. Shannon's entropy reflects the expected information content of an outcome and is defined as

$$H(\boldsymbol{p}) \equiv \sum_{k=1}^{K} p_k \log_2 \frac{1}{p_k} = -\sum_{k=1}^{K} p_k \log_2 p_k = E\left[\log_2\left(\frac{1}{p(X)}\right)\right], \tag{1}$$

for the random variable X. This information criterion, expressed in bits, measures the uncertainty of X that is implied by $\boldsymbol{p}$. The entropy measure $H(\boldsymbol{p})$ reaches a maximum when $p_1 = p_2 = \cdots = p_K = \frac{1}{K}$ and a minimum with a point mass function. The entropy $H(\boldsymbol{p})$ is a function of the probability distribution $\boldsymbol{p}$ and not a function of the actual values taken by the random variable.

The remainder of the paper is laid out as follows: in section "Generalized Maximum Entropy Estimator", maximum entropy and generalized maximum entropy estimation are briefly discussed. Section "Large-Scale Data and Aggregation" illustrates some traditional aggregation procedures and a new proposal based on normalized entropy. Section "Simulation Study" presents simulation results. Some conclusions and topics for future research are given in section "Concluding Remarks".

Generalized Maximum Entropy Estimator

The ME principle was discussed by Golan et al. [6], in order to develop analytical and empirical methods for recovering the unobservable parameters of a pure linear inverse problem. Considering then

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{p}, \tag{2}$$

where $\boldsymbol{y}$ is the vector $(N \times 1)$ of observations, $\boldsymbol{X}$ is a non-invertible matrix $(N \times K)$ with $N < K$, and $\boldsymbol{p}$ is the vector $(K \times 1)$ of unknown probabilities, the ME principle consists in choosing $\boldsymbol{p}$ that maximizes Shannon's entropy

$$H(\boldsymbol{p}) = -\sum_{k=1}^{K} p_k \ln p_k = -\boldsymbol{p}' \ln \boldsymbol{p}, \tag{3}$$

subject to the data consistency restriction, $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{p}$, and the additivity restriction, $\boldsymbol{p}'\mathbf{1} = 1$. Formally, the ME estimator is given by

$$\underset{\boldsymbol{p}}{\operatorname{argmax}} \left\{-\boldsymbol{p}' \ln \boldsymbol{p}\right\}, \tag{4}$$

subject to the model consistency and additivity constraints,

$$\begin{cases} \boldsymbol{y} = \boldsymbol{X}\boldsymbol{p} \\ \mathbf{1}'\boldsymbol{p} = 1 \end{cases}. \tag{5}$$

There is no closed-form analytical solution, but a numerical approximation can be obtained using the Lagrange multipliers. It can be said that the Jaynes maximum entropy formalism has enabled us to solve the pure inverse problem with this optimization (maximization) procedure, regarding it as an inference problem. The ME principle is the basis for transforming the information in the data into a probabilistic distribution that reflects our uncertainty about individual outcomes.

To extend the ME estimator to the linear regression model represented by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{e}, \tag{6}$$

where, as usually, $\boldsymbol{y}$ denotes a $(N \times 1)$ vector of noisy observations, $\boldsymbol{\beta}$ is a $(K \times 1)$ vector of unknown parameters, $\boldsymbol{X}$ is a known $(N \times K)$ matrix of explanatory variables, and $\boldsymbol{e}$ is the $(N \times 1)$ vector of random disturbances (errors), Golan et al. [6], considered each β_k as a discrete random variable with a compact support and $M \geq 2$ possible outcomes and each e_n as a finite and discrete random variable with $J \geq 2$ possible outcomes. The error vector is considered here as another vector of unknown parameters to be estimated simultaneously with the vector $\boldsymbol{\beta}$. In this context, the linear regression model is represented as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{Z}\boldsymbol{p} + \boldsymbol{V}w, \tag{7}$$

where

$$\boldsymbol{\beta} = \boldsymbol{Z}\boldsymbol{p} = \begin{bmatrix} z_1' & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & z_2' & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & z_K' \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_1 \\ \boldsymbol{p}_2 \\ \vdots \\ \boldsymbol{p}_K \end{bmatrix}, \tag{8}$$

and

$$\boldsymbol{e} = \boldsymbol{V}w = \begin{bmatrix} \boldsymbol{v}_1' & \boldsymbol{0} & \dots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{v}_2' & \dots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \dots & \boldsymbol{v}_N' \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix}. \tag{9}$$

Matrices $\boldsymbol{Z}$ $(K \times KM)$ and $\boldsymbol{V}$ $(N \times NJ)$ are the matrices of support values and vectors $\boldsymbol{p}$ $(KM \times 1)$ and w $(NJ \times 1)$ are the vectors of unknown probabilities to be estimated. Note that each β_k, $k = 1, 2, \dots, K$, and each e_n, $n = 1, 2, \dots, N$, are viewed as expected values of discrete random variables $\boldsymbol{z}_k$ and $\boldsymbol{v}_n$, respectively, with $M \geq 2$ and $J \geq 2$ possible outcomes, within the lower and upper bounds of the corresponding support spaces. Thus, the generalized maximum entropy (GME) estimator is given by

$$\underset{\boldsymbol{p},w}{\operatorname{argmax}} \left\{ -\boldsymbol{p}' \ln \boldsymbol{p} - w' \ln w \right\}, \tag{10}$$

subject to the consistency (with the model) and additivity (for $\boldsymbol{p}$ and w) constraints,

$$\begin{cases} \boldsymbol{y} = \boldsymbol{X}\boldsymbol{Z}\boldsymbol{p} + \boldsymbol{V}w, \\ \boldsymbol{1}_K = (\boldsymbol{I}_K \otimes \boldsymbol{1'}_M)\boldsymbol{p}, \\ \boldsymbol{1}_N = (\boldsymbol{I}_N \otimes \boldsymbol{1'}_J)w, \end{cases} \tag{11}$$

where $\otimes$ represents the Kronecker product. The optimal probability vectors, $\widehat{\boldsymbol{p}}$ and $\widehat{w}$, are used to obtain point estimates of the unknown parameters and the unknown errors with $\widehat{\boldsymbol{\beta}} = \boldsymbol{Z}\widehat{\boldsymbol{p}}$ and $\widehat{\boldsymbol{e}} = \boldsymbol{V}\widehat{w}$. Some properties of the GME estimator, such as consistency and asymptotic normality, are discussed in detail, for example, in Mittelhammer et al. [7].

Large-Scale Data and Aggregation

Large-scale data or big data usually refers to datasets that are large in different ways: many observations, many variables (or both); observations are recorded in different time regimes or are taken from multiple sources. Some difficult issues arise in dealing with this kind of data like, for instance, retaining optimal (or, at least, reasonably good) statistical properties with a computationally efficient analysis; or dealing with inhomogeneous data that does not fit in the classical framework: data is neither i.i.d. (exhibiting outliers or not belonging to same distribution) nor stationary (time-varying effects may be present).

Standard statistical models (linear or generalized linear models for regression or classification) fail to capture inhomogeneity structure in data, compromising estimation and interpretation of model parameters, and, of course, prediction. On the other hand, statistical approaches for dealing with inhomogeneous data (such as varying-coefficient models, mixed effects models, mixture models or clusterwise regression models) are typically very computationally cumbersome.

Ignoring heterogeneity in data, computational burden can be addressed with the following procedure, [8]: firstly, construct g groups from the large-scale data (groups may be overlapping and may not cover all observations in the sample); then, for each group compute an estimator, $\widehat{\boldsymbol{\beta}}_g$, through standard techniques (e.g., OLS, ridge or LASSO); finally, considering the ensemble of estimators, aggregate them into a single estimator, $\widehat{\boldsymbol{\beta}}$.

Traditional Aggregation Procedures

Several aggregation procedures have been already proposed in literature. Three of them are presented next.

1. Bagging: this procedure results in less computational complexity and even allows for parallel computing. It simply averages the ensemble estimators with equal weight to obtain the aggregated estimator, [8, 9]:

$$\widehat{\boldsymbol{\beta}} := \sum_{g=1}^{G} w_g \widehat{\boldsymbol{\beta}}_g, \tag{12}$$

where $w_g = \frac{1}{G}$ for all $g = 1, 2, \ldots, G$. The estimates $\widehat{\boldsymbol{\beta}}_g$ are obtained from bootstrap samples, where the groups are sampled with replacement from the whole data. It is a simple procedure and the weights do not depend on the response $\boldsymbol{y}$, but it is not suitable for inhomogeneous data.

2. Stacking: instead of assigning a uniform weight to each estimator, [10, 11] proposed the aggregated estimator

$$\widehat{\boldsymbol{\beta}} := \sum_{g=1}^{G} w_g \widehat{\boldsymbol{\beta}}_g, \tag{13}$$

where

$$w := \underset{w \in \boldsymbol{W}}{\operatorname{argmin}} \left\| \boldsymbol{y} - \sum_{g=1}^{G} w_g \widehat{\boldsymbol{y}}_g \right\|_2, \tag{14}$$

and, using a ridge constraint, $\boldsymbol{W} = \{w : \|w\| \leq s\}$, for some $s > 0$, or using a sign constraint, $\boldsymbol{W} = \{w : \min_g w_g \geq 0\}$, or using a convex constraint, $\boldsymbol{W} = \{w : \min_g w_g \geq 0 \text{ and } \sum_{g=1}^{G} w_g = 1\}$. The idea is to find the optimal linear or convex combination of all ensemble estimators, but it is also not suitable for inhomogeneous data.

3. Magging: corresponds to maximizing the minimally "explained variance" among all data groups, [8], such that

$$\widehat{\boldsymbol{\beta}} := \sum_{g=1}^{G} w_g \widehat{\boldsymbol{\beta}}_g, \tag{15}$$

where

$$w := \underset{w \in \boldsymbol{W}}{\operatorname{argmin}} \left\| \sum_{g=1}^{G} w_g \widehat{\boldsymbol{y}}_g \right\|_2, \tag{16}$$

and $\boldsymbol{W} = \{w : \min_g w_g \geq 0 \text{ and } \sum_{g=1}^{G} w_g = 1\}$. The idea is to choose the weights as a convex combination to minimize the $\|\cdot\|_2$ of the fitted values, $\widehat{\boldsymbol{y}}$. If the solution is not unique, it is considered the solution with lowest $\|\cdot\|_2$ of the weight vector among all solutions. This procedure was the first that we are aware of that was proposed for heterogeneous data. The main idea is that if an effect is common across all groups, then it cannot be "averaged away" by searching for a specific combination of the weights. The common effects will be present in all groups and will be retained even after the minimization of the aggregation scheme.

We believe the question as to whether the effects are really common across all groups may not be answered straightforwardly. If the groups carry information about the whole dataset and there are inhomogeneities, why should we consider that, with random sub-sampling, all groups are equally informative?

These considerations led us to the idea of choosing the groups according to their "information content".

Proposed Aggregation Procedure

To measure the information content in a system and to measure the importance of the contribution of each piece of data or constraint in reducing uncertainty, Golan et al. [6], stated that, in the ME formulation, the maximum level of entropy-uncertainty results when the information-moment constraints are not enforced and the distribution of probabilities over the K states is uniform. As each piece of effective data is added, there is a departure from the uniform distribution, which implies a reduction of uncertainty. The proportion of the remaining total uncertainty is measured by the normalized entropy (NE),

$$S(\widehat{\boldsymbol{p}}) = -\frac{\sum_k \widehat{p}_k \ln \widehat{p}_k}{\ln(K)}, \tag{17}$$

where $S(\widehat{\boldsymbol{p}}) \in [0, 1]$ and $\ln(K)$ represents maximum uncertainty (the entropy level of the uniform distribution with K outcomes). A value $S(\widehat{\boldsymbol{p}}) = 0$ implies no uncertainty and a value $S(\widehat{\boldsymbol{p}}) = 1$ implies perfect uncertainty. Related to the normalized entropy, the information index (II) is defined as $1 - S(\widehat{\boldsymbol{p}})$ and measures the reduction in uncertainty.

In this work, we propose a new aggregation scheme that is based on identifying the information content of a given group through the calculation of the normalized entropy. The proposed NE aggregated estimator is then

$$\widehat{\boldsymbol{\beta}} := \sum_{g=1}^{G} w_g \widehat{\boldsymbol{\beta}}_g, \tag{18}$$

where w_g is defined by normalized entropy using GME,

$$S(\widehat{\boldsymbol{p}})_g = \frac{-\widehat{\boldsymbol{p}}' \ln \widehat{\boldsymbol{p}}}{K \ln M}, \tag{19}$$

for the signal, $\boldsymbol{X\beta}$, such that $\sum_{g=1}^{G} w_g = 1$. This aggregation procedure is a weighted average of the collection of regression coefficient estimates as in Bagging, Stacking and Magging. The idea is almost as simple as Bagging and it is expected to provide similar results if the data is homogeneous. However, since the weights in (18) will depend on the information content of each group according to (19), or some function of it, the weights will be, in general, non-uniform (as in Stacking and Magging) if the data is inhomogeneous.

Following section reports some simulated situations for which the NE aggregated estimator was calculated and compared to the aggregated estimator based on Bagging.

Simulation Study

A linear regression model was considered, where $\boldsymbol{X}$ is the simulated matrix of explanatory variables, drawn randomly from normal distributions; $\boldsymbol{\beta}$ is a vector of parameters, $\boldsymbol{e}$ is the vector of random disturbances, drawn randomly from normal distributions and $\boldsymbol{y}$ is the constructed vector of noisy observations. For this simulation, $\boldsymbol{\beta}$ was considered as

$$\boldsymbol{\beta} = [1.8,\quad 1.2,\quad -1.4,\quad 1.6,\quad -1.8,\quad 2.0,\quad -2.0,\quad 0.2,\quad -0.4,\quad 0.6,\quad 0.8]. \tag{20}$$

Necessary reparameterizations were done considering $M = 5$ and $J = 3$ and different matrices $\boldsymbol{Z}$ containing the supports for the parameters. The support matrix $\boldsymbol{V}$ containing the supports for the errors was set considering symmetric and zero-centred supports using the three-sigma rule with the empirical standard deviation of the noisy observations.

Simulations were done considering $\boldsymbol{X}$ a (20000×11) matrix; $\boldsymbol{\beta}$ a (11×1) vector; $\boldsymbol{e}$ a (20000×1) vector and $\boldsymbol{y}$ a (20000×1) vector. The error distribution was considered to be normal, with mean value zero and standard deviation five. Several matrices $\boldsymbol{X}$ of explanatory variables were simulated, corresponding to different condition numbers (c.n.).[1] Random sub-sampling with replacement was done considering different number of groups and 50 observations per group. The Euclidean norm of the difference between the aggregated estimator $\widehat{\boldsymbol{\beta}}$ and the true parameter $\boldsymbol{\beta}$, $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$, is calculated for each simulated case and the results are given in Tables 1, 2, 3, 4 and 5. For each case, three different solutions are presented, namely,

1. NE1: the chosen $\widehat{\boldsymbol{\beta}}$ corresponds to the GME estimate for the group with lower normalized entropy, (NE). This solution does not correspond, in fact, to an aggregated estimator; it corresponds to a chosen estimate amongst all groups;
2. NE2: the chosen $\widehat{\boldsymbol{\beta}}$ corresponds to the weighted average of the GME estimates of all groups, weighted by the information index, II, where II = 1−NE;
3. Bgg: the $\widehat{\boldsymbol{\beta}}$ chosen corresponds to Bagging (average of the OLS estimates of all groups).[2]

The present results are intended to highlight the overall tendencies we encountered in the simulation study. Many other situations were simulated, with many different matrices of explanatory variables, $\boldsymbol{X}$, corresponding to a wide range of variation regarding the matrix condition number, which, as is well known, is related to the

[1] Ratio of the largest singular value of $\boldsymbol{X}$, with the smallest singular value.

[2] It is not considered here the case of a single learning set, as in [9], and the need to take repeated bootstrap samples from it.

Table 1 Euclidean norm of the difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, with $z_k = [-10, 10]$

n.g.	Solution	c.n. = 1337
5	NE1	4.26
	NE2	4.18
	Bgg	181.23

Table 2 Euclidean norm of the difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, with $z_k = [-10, 10]$

n.g.	Solution	c.n. = 43030
5	NE1	4.22
	NE2	4.25
	Bgg	1432.59

Table 3 Euclidean norm of the difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, with $z_k = [-10, 10]$

n.g.	Solution	c.n. = 1337
5	NE1	4.26
	NE2	4.18
	Bgg	181.23
10	NE1	4.47
	NE2	4.31
	Bgg	171.22
50	NE1	4.45
	NE2	4.30
	Bgg	49.36
100	NE1	5.48
	NE2	4.34
	Bgg	38.74

Table 4 Euclidean norm of the difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, with $z_k = [-100, 100]$

n.g.	Solution	c.n. = 1337
5	NE1	32.31
	NE2	10.17
	Bgg	214.56

Table 5 Euclidean norm of the difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}$, with $z_k = [-100, 100]$

n.g.	Solution	c.n. = 1337	c.n. = 43030
5	NE1	32.31	35.54
	NE2	10.17	15.59
	Bgg	214.56	5440.47

presence of collinearity[3] in the explanatory variables. In this paper, only two extreme cases were chosen to be presented, the first one corresponding to a relatively small condition number (c.n. around 1300) and the second one corresponding to a much higher condition number (c.n. around 43000).

It can be concluded that, for both cases, $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$ is much lower for any of the normalized entropy methodologies, when compared to Bagging, as can be seen from any of the Tables 1, 2, 3, 4 and 5.

Comparing Tables 1 and 2, same number of groups (n.g.=5) and same support vectors for the parameters ($z_k = [-10, 10]$) were considered. The higher condition number in Table 2 results in a much higher $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$ for the Bagging procedure, whereas the normalized entropy methodologies behave in the same way as with the much lower condition number, revealing that the presence of collinearity does not seem to compromise the results provided by the normalized entropy aggregation procedures. Since the GME estimator is appropriate in the estimation of ill-posed models, including models with ill-conditioned design matrices, these results are not surprising.

Considering Table 3, the analysis was done changing the number of groups in the aggregation. The Bagging procedure tends to provide better results in terms of lower $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$, as the number of groups rises. This observation does not come as a surprise due to sampling and inferential statistics theory. The normalized entropy methodologies do not seem to follow this behaviour, as the $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$ remains approximately constant as the number of groups gets higher. This may be considered an advantage of this aggregation procedure, since there is no need for bigger data sets (and consequent higher computational burden) in order to have comparable results in terms of precision.

Finally, Tables 4 and 5 refer to the effect of changing the amplitude of the support vectors, z_k. It can be seen that as the support vector z_k changes from $[-10, 10]$, in Table 1, to $[-100, 100]$, in Table 4, all aggregation procedures provide worse results in terms of $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$. Widening the amplitude of the support vectors results in a less informative probability distribution for the parameters, which should lead to a smaller departure from total uncertainty as compared to the situation where the support vectors are less wide. It is expected, then, that the normalized entropy methodologies provide better results when the amplitude of the support vectors are smaller. The results of the simulation study are in agreement with this interpretation. Nevertheless, when the same analysis is done considering a matrix of explanatory variables $\boldsymbol{X}$, with higher condition number, as presented in Table 5, even though the normalized entropy methodologies provide worse results, as already discussed, the Bagging procedure provides even worse results: while $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$ changes from 4.25 to 15.59 for the information index weighted average of the GME estimates (solution NE2), the corresponding change for the Bagging procedure is from 1432.59 (which is already a very poor value concerning the precision of the estimates) to 5440.47.

[3] The concept is not used here in a literal sense. A discussion about similar notions of this concept is available in Belsley et al. [12, pp. 85–98].

Concluding Remarks

The idea of an aggregation procedure based on normalized entropy is promising as it is clear from the simulation study that this approach provides very satisfactory solutions. The normalized entropy methodologies, in particular, the aggregation procedure based on the weighting of the groups by the information index, always results in a $\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|_2$ much lower than the one obtained with Bagging. This discrepancy tends to aggravate in the presence of high collinearity, as that is the case when the explanatory variables matrices, $\boldsymbol{X}$, have high condition numbers. On the other hand, the use of more groups in the aggregation scheme does not seem to improve the overall quality of the estimates obtained through the normalized entropy methodologies, what turns out to be an advantage towards this procedure. These observations suggest that a further and thorough simulation analysis with different error structures or severe inhomogeneities may reveal substantial differences between normalized entropy aggregation schemes and Bagging, eventually penalizing the second. These analysis will be conducted in future work, along with investigation of other scenarios, such as the detection of zero coefficients, non-normal regressors and other violations of the classical framework. Also, the comparison with Magging is a very important analysis that remains to be explored.

Acknowledgements This research was supported by the Portuguese national funding agency for science, research and technology (FCT), within the Center for Research and Development in Mathematics and Applications (CIDMA), project UID/MAT/04106/2019.

References

1. Golan, A.: On the state of art of Info-Metrics. In: Huynh, V.N., Kreinovich, V., Sriboonchitta, S., Suriya, K. (Eds.) Uncertainty Analysis in Econometrics with Applications, pp. 3–15. Springer, Berlin (2013)
2. Golan, A.: Foundations of Info-Metrics: Modeling, Inference, and Imperfect Information. Oxford University Press, New York (2018)
3. Golan, A.: On the foundations and philosophy of Info-Metrics. In: Cooper, S.B., Dawar, A., Lowe, B.L. (Eds.) CiE2012. LNCS, vol. 7318, pp. 238–245. Springer, Heidelberg (2012)
4. Jaynes, E.T.: Information theory and statistical mechanics. Phys. Rev. **106**, 620–630 (1957)
5. Jaynes, E.T.: Information theory and statistical mechanics II. Phys. Rev. **108**, 171–190 (1957)
6. Golan, A., Judge, G., Miller, D.: Maximum Entropy Econometrics—Robust Estimation with Limited Data. Wiley, Chichester (1996)
7. Mittelhammer, R., Cardell, N.S., Marsh, T.L.: The Data-constrained generalized maximum entropy estimator of the GLM: asymptotic theory and inference. Entropy **15**, 1756–1775 (2013)
8. Bühlmann, P., Meinshausen, N.: Magging: maximin aggregation for inhomogeneous large-scale data. In: Proceedings of the IEEE 104 (1): Big Data: Theoretical Aspects, pp. 126–135. IEEE Press, New York (2016)
9. Breiman, L.: Bagging predictors. Mach. Learn. **24**, 123–140 (1996)
10. Wolpert, D.: Stacked generalization. Neural Netw. **5**, 241–259 (1992)
11. Breiman, L.: Stacked regressions. Mach. Learn. **24**, 49–64 (1996b)
12. Belsley, D.A., Kuh, E., Welsch, R.E.: Regression Diagnostics—Identifying Influential Data and Sources of Collinearity. Wiley, Hoboken, New Jersey (2004)

Modified Granger Causality in Selected Neighborhoods

Martina Chvosteková

Abstract Although Granger causality is a widely used technique to detect the causal relationship between time series, its direct application for nonlinearly modeled data is not appropriate. There have been proposed several extensions to nonlinear cases, but there is no method appropriate for detecting relations between time series in general. We present a new measure for evaluation of a causal effect between two time series, which is calculated on the selected local approximations of time-delay embedding reconstruction of state space by a linear regression model. The novel causal measure, called the modified Granger causality in selected neighborhoods (MGCiSN), reflects the proportion of the explained variation of the modeled variable by the past of the second variable only. The proposed procedure for evaluating the direct causal link between two nonlinearly modeled time series is applied to four data sets with different known nonlinear causal structures. Our experimental results support that the MGCiSN correctly detects underlying causal relationship in many cases and does not detect false causality, regardless of the number of samples.

Keywords Granger causality · Time-delay embedding reconstruction · Linear regression model · Prediction error

Introduction

Assessing the presence of directional interactions between simultaneously recorded variables is an essential issue in diverse areas including finance, neuroscience, sociology, and others. The Granger causality [7] has become the most popular tool to identify relationship between two time series corresponding to the recorded variables due to its computational simplicity. A variable X is said to Granger-cause another variable Y if the prediction error of Y in a linear regression model including its own past values and the past values of X as predictors is less (in some suitable sense)

M. Chvosteková (✉)
Institute of Measurement Science of Slovak Academy of Sciences, 9 Dúbravská cesta, 84104 Bratislava, Slovakia
e-mail: martina.chvostekova@savba.sk

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*, Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_3

than the prediction error of Y in a linear regression model including only its own past values. X Granger-causes Y means that the variable X was found to be helpful for forecasting variable Y. So, the notation of Granger causality implies predictability and precedence, but it does not imply true causality. Note that the detection approach can be theoretically used also for time series actually not generated by a linear regression model [1]. The most important issue for using the Granger causality analysis is verification that the time series data can be modeled by a stochastic linear regressive scheme. If the model assumption is not satisfied, the result of the commonly used F-test (or Wald test) for Granger causality is not approved.

Extensions of the Granger causality to nonlinear cases have been explored, e.g., in [2, 12, 13]. Also, the causality detection methods that do not directly follow the traditional Granger causality methodology have been proposed, e.g., [10, 15, 16]. The results from recently published rigorous comparison study of various causality detection methods indicate that the most methods have extremely low specificity (they produce false detection of causality) and there is no rule how to choose the appropriate method for particular data. Furthermore, extensive computations and no straightforward interpretation of the numeric results decrease the usefulness of some of these procedures. It can be concluded that identifying a causal relationship among simultaneously acquired processes is still not a satisfactorily closed issue, even for the bivariate case.

In this work, we present a new method for quantifying the causal structure for nonlinearly modeled bivariate time series. The proposed procedure, called the modified Granger causality in selected neighborhoods (MGCiSN), is based on a local approximation of the reconstructed dynamics by a linear regression model. The causality detection methods developed on a similar approach have appeared in [2, 6]. The presented procedure differs from the mentioned methods in the way how the local neighborhoods are selected and the most important difference is in using variable for exploring the relationship between time series. In our procedure, the direct causal influence of X on Y is evaluated by determining the proportion of predicted variation of the variable Y by the past of the variable X only. These proportions are calculated only on those local neighborhoods, where the linear regression model fits the reconstructed joint dynamics. The goodness of fit in neighborhoods is assessed by the coefficient of determination, R-squared (R^2). Finally, the MGCiSN is obtained as the average of these proportions from suitable neighborhoods over the attractor.

We examined the suggested measure on numerical nonlinear time series with known nonlinear asymmetric dependencies. Four artificially generated data sets were analyzed: unidirectionally coupled nonidentical two Hénon maps, bidirectionally coupled nonidentical two Hénon maps, and two systems composed of unidirectionally different nonlinear coupled chaotic maps.

Methodology

Consider two variables X and Y, represented by the stationary time series x and y, respectively. The classical Granger causality starts with fitting the time series by a bivariate autoregressive model ($VAR(p)$), where the Akaike information criterion (AIC) or the Schwartz Bayesian information criterion (BIC) is usually used to determine the order p. In this study, the number of predictors, which are included in the modeling, are chosen by employing Takens' time-delay embedding [17]. The state spaces $\widetilde{X}$ corresponding to the time series x is reconstructed by the following time-delayed embedding vector

$$x^{m_x,\tau_x}(t) = (x(t), x(t-\tau_x), \ldots, x(t-(m_x-1)\tau_x))^T, \tag{1}$$

where m_x is the embedding dimension and τ_x is the time delay. Similarly, the time-delayed embedding vector reconstructing the state space $\widetilde{Y}$ of time series y is defined as

$$y^{m_y,\tau_y}(t) = (y(t), y(t-\tau_y), \ldots, y(t-(m_y-1)\tau_y))^T. \tag{2}$$

The most common practice to determine the reconstruction parameters is to take the delay as the first minimum of the mutual information between the delayed components [5] and the embedding dimension is estimated by the false nearest neighbor technique [8]. For investigating the Granger causality, the time delays must be equal, therefore we will use $\tau_x = \tau_y = \tau$ in the following text. Now, the embedding dimensions m_x and m_y will determine the number of lagged observations of X and Y in a linear regression model, respectively.

Let us express the delay vector of the joint state space of $\widetilde{X}$ and $\widetilde{Y}$ as

$$z^{m_x,m_y,\tau}(t) = (x^{m_x,\tau}(t)^T, y^{m_y,\tau}(t)^T)^T, \tag{3}$$

where $z^{m_x,m_y,\tau}(t)$ is a point in the (m_x+m_y)-dimensional reconstructed state space $\widetilde{Z}$. It is supposed that the joint dynamics can be locally approximated by a linear map, written as $z(t+\tau) = Az(t) + \epsilon(t)$, where A is $(m_x+m_y) \times (m_x+m_y)$ coefficient matrix and $\epsilon(t)$ is the error vector. In the next step of our procedure, the neighborhoods in $\widetilde{Z}$ suitable for a local linear approximation are selected.

Selection of Local Neighborhoods

Let $z^{m_x,m_y,\tau}(t_{[0]})$ be a point in the joint state space $\widetilde{Z}$ and $z^{m_x,m_y,\tau}(t_{[1]})$, $z^{m_x,m_y,\tau}(t_{[2]})$, $\ldots$, $z^{m_x,m_y,\tau}(t_{[k]})$ be its k-nearest neighbors. For all $k+1$ points, fit the full linear regression models of the following form

$$x(t_{[i]} + \tau) = c_{xy} + \sum_{j=0}^{m_x-1} a_{xx}^{(j)} x(t_{[i]} - j\tau) + \sum_{j=0}^{m_y-1} a_{xy}^{(j)} y(t_{[i]} - j\tau) + \epsilon_{xy}(t_{[i]}), \tag{4}$$

$$y(t_{[i]} + \tau) = c_{yx} + \sum_{j=0}^{m_x-1} a_{yx}^{(j)} x(t_{[i]} - j\tau) + \sum_{j=0}^{m_y-1} a_{yy}^{(j)} y(t_{[i]} - j\tau) + \epsilon_{yx}(t_{[i]}), \tag{5}$$

where $\epsilon_{xy}(t)$, $\epsilon_{yx}(t)$ are the prediction error terms and their magnitudes can be evaluated by their variances, i.e., $var(\epsilon_{xy})$, $var(\epsilon_{yx})$. The unknown intercepts c_{xy}, c_{yx}, and the unknown coefficients $a_{xx}^{(j)}$, $a_{xy}^{(j)}$, $a_{yx}^{(j)}$, $a_{yy}^{(j)}$ of the models (4, 5) can be determined by the least squares technique. Then, for the same $k+1$ points, perform fitting process of the intercept-only models of the following form

$$x(t_{[i]} + \tau) = \bar{x} + \epsilon_x(t_{[i]}), \tag{6}$$

$$y(t_{[i]} + \tau) = \bar{y} + \epsilon_y(t_{[i]}), \tag{7}$$

where $\epsilon_x(t_{[i]})$, $\epsilon_y(t_{[i]})$ are the prediction error terms, $\bar{x}$ and $\bar{y}$ is mean of $x(t_{[i]})$'s and $y(t_{[i]})$'s, respectively. Now, we can define the sum of squares

$$s_{xy}^2 = \sum_{i=0}^{k} \hat{\epsilon}_{xy}(t_{[i]})^2, \; s_x^2 = \sum_{i=0}^{k} \hat{\epsilon}_x(t_{[i]})^2, \tag{8}$$

$$s_{yx}^2 = \sum_{i=0}^{k} \hat{\epsilon}_{yx}(t_{[i]})^2, \; s_y^2 = \sum_{i=0}^{k} \hat{\epsilon}_y(t_{[i]})^2, \tag{9}$$

where $\hat{\epsilon}_{xy}$, $\hat{\epsilon}_{yx}$, $\hat{\epsilon}_{xx}$, $\hat{\epsilon}_{yy}$, are determined estimates of error ϵ_{xy}, ϵ_{yx}, ϵ_x, ϵ_y based on the fitted models (4, 5, 6, 7).

In classical linear regression analysis, the coefficient of determination, denoted R^2 (R - squared), is often used for evaluating the model fit. The quality of the fitted full linear regression model (5) in the neighborhood corresponding to $z^{m_x,m_y,\tau}(t_{[0]})$ is evaluated via

$$R^2_{y/\{x,y\}} = 1 - s_{yx}^2/s_y^2. \tag{10}$$

If $R^2_{y/\{x,y\}}$ value is greater than a prescribed value, denoted R^*, then the $(k+1)$-th nearest point to $z^{m_x,m_y,\tau}(t_{[0]})$, denoted $z^{m_x,m_y,\tau}(t_{[k+1]})$, is added to the neighborhood and $R^2_{y/\{x,y\}}$ is calculated again. If the new value of $R^2_{y/\{x,y\}}$ is not smaller than one from the previous step, then the next nearest point $z^{m_x,m_y,\tau}(t_{[k+2]})$ is added to the neighborhood and the procedure is repeated. The number of points in a suitable neighborhood for detecting the causal link of X on Y increases as long as $R^2_{y/\{x,y\}}$ does not decrease. Analogously, the appropriateness of a neighborhood in $\tilde{Z}$ for exploring the causal link Y to X is evaluated through the value $R^2_{x/\{x,y\}} = 1 - s^2_{xy}/s^2_x$. Here we suggest to use $R^* = 0.95$ and $k = 30$.

Modified Granger Causality Index

The mathematical formulation of the classical Granger causality is based on a linear regression modeling of stochastic processes. The Granger causality does not imply true causality, it reflects variable's prediction ability. The idea behind the Granger causality is well comprehended. X Granger-causes Y, if the prediction error of y from a linear regression model including only own past values of y as predictors (restricted linear regression model) is reduced, in a statistically suitable sense, by incorporating past values of x as predictors in the linear regression model. The magnitude of the Granger causality of X on Y can be measured by the log ratio $F_{X\to Y} = \ln(var(\epsilon_{yy})/var(\epsilon_{yx}))$, see, e.g., [1], or by the Granger causality index $\delta_{X\to Y} = 1 - var(\epsilon_{yx})/var(\epsilon_{yy})$ see, e.g., [2], where $var(\epsilon_{yy})$ denotes the prediction error variance for the restricted model , and $var(\epsilon_{yx})$ denotes the prediction error variance for the full linear regression model. Both measures should be null if the past of x is ineffective to the prediction improvement of y. It is important to note that it is meaningless to compare nonzero values of $F_{X\to Y}$ from another couple of series since $F_{X\to Y}$ is not scaled on a range. The Granger causality index is a more pragmatic quantity from this point of view, $\delta_{X\to Y}$ indicates a proportion of the variance of y in the restricted model, which is not explained by past of y itself and can be explained by added past of x as predictors to a linear representation of the observed processes. In general, the Granger causality is focused on improving the prediction error of modeled variable in a restricted model, but the value of the prediction error alone is out of interest. It means that if $(var(\epsilon_{yy}), var(\epsilon_{yx})) = (10, 1)$ or $(var(\epsilon_{yy}), var(\epsilon_{yx})) = (10^{-4}, 10^{-5})$, then $\delta_{X\to Y} = 0.9$ in both cases, i.e., 90% of unexplained variation of Y by its own past is expressed by implementing the past of X to a representation of Y and the causal link X to Y is indicated. Consequently, the methods based on the Granger causality concept fail to detect asymmetric causal dependencies, even linear, between bivariate time series, the false causality is often detected, see, e.g., [9]. The suggested novel causality measure is based on specifying the proportion of the variation of the modeled variable explained actually by past values of the second variable in full linear regression model.

The proportion of the explained sum of squares of y by a fitted full linear regression model is known through the value $R^2_{y/\{x,y\}}$ for a selected suitable neighborhood. In the next step of our procedure, on the same specified neighborhood, the restricted linear regression model of the form

$$y(t_{[i]} + \tau) = c_{yy} + \sum_{j=0}^{m_y - 1} a_y^{(j)} y(t_{[i]} - j\tau) + \epsilon_{yy}(t_{[i]}) \qquad (11)$$

is fitted. The unknown parameters of the model, the intercept c_{yy} and the coefficients $a_y^{(j)}$ can be estimated by the least squares procedure. The estimates $\hat{\epsilon}_{yy}(t_{[i]})$ of the prediction error $\epsilon_{yy}(t_{[i]})$ are used to determine the following sum of squares

$$s_{yy}^2 = \sum_{i=0}^{\tilde{k}} \hat{\epsilon}_{yy}(t_{[i]})^2, \tag{12}$$

where $\tilde{k}+1$ denotes the number of points in the suitable neighborhood. Then, the proposed measure for quantifying the causal effect of X on Y in a selected suitable neighborhood, denoted $\lambda_{X\rightarrow Y}$, is defined as

$$\lambda_{X\rightarrow Y} = \frac{\lambda'_{X\rightarrow Y}}{R^2_{y/\{x,y\}}} = \frac{s_{yy}^2 - s_{yx}^2}{s_y^2 - s_{yx}^2}, \text{ where } \lambda'_{X\rightarrow Y} = \frac{s_{yy}^2 - s_{yx}^2}{s_y^2}. \tag{13}$$

A value $\lambda'_{X\rightarrow Y}$ is a ratio of the explained sum of squares of y from the restricted model (11) by the full model (5) and the total sum of squares of variable y. Thus, the value $\lambda_{X\rightarrow Y}$ indicates the proportion of the variation of the response variable y, explained by the linear regression model including past of x and y as predictors, actually predicted by x values only.

The magnitude of the suggested modified Granger causality of X on Y in selected neighborhoods, denoted $\Lambda_{X\rightarrow Y}$, is the average of the indices $\lambda_{X\rightarrow Y}$ from all suitable neighborhoods on the attractor. Here, as the significant presence of the causal influence of X on Y, a $\Lambda_{X\rightarrow Y}$ value greater than 0.01 is considered. $\Lambda_{X\rightarrow Y} \geq 0.01$ means that more than 1% of the explained variation of y by the full linear regression model is actually expressed by past of x in the linear representation. In a similar way, the magnitude of the MGCiSN of Y on X denoted $\Lambda_{Y\rightarrow X}$ can be defined.

Numerical Experiments

In order to investigate the behavior of the proposed causality measure, we artificially generated time series from four model systems of nonlinear dynamics with known nonlinear causal links: three systems composed of unidirectionally coupled chaotic maps with different types of nonlinear causal effect and one system of bidirectionally coupled chaotic maps with non-linear causal effect. In all numerical experiments, the strength of the coupling c varied on a specified range and the MGCiSN was evaluated on the data sets of $N = 10000$ and $N = 1000$ samples for each considered c value. The time delay for the studied examples was set to $\tau = 1$ and the embedding dimensions in analyzed systems were set up to $m_x = m_y = 2$. Note that none of the data sets is appropriate for the classical Granger causality analysis.

Unidirectional Nonlinear Coupling I

In the first example, we studied two unidirectionally coupled nonidentical Hénon maps (see, e.g., [4]):

$$x(t) = 1.4 - x(t-1)^2 + 0.3x(t-2), \tag{14}$$

$$y(t) = 1.4 - [cx(t-1)y(t-1) + (1-c)y(t-1)^2] + 0.1y(t-2) \tag{15}$$

where the strength of the coupling c was varied from 0 to 0.78 with increments of 0.06. By construction, the variable X represented by the time series x has causal influence on the variable Y represented by the time series y for $c \neq 0$, and the variables X and Y are not causally connected for $c = 0$.

Figures 1 and 2 show results of our first experiment. We observe that the MGCiSN successfully indicates the absence of the causal influence of Y on X , $\Lambda_{Y\to X}$ is zero at all values of c for $N = 10000$ and $N = 1000$. The MGCiSN correctly detected the causal relationship of X on Y at $c > 0.06$ for both sample sizes, but $\Lambda_{X\to Y}$ is nonzero at almost all $c > 0$.

Fig. 1 Unidirectional coupling I (true causality $X \to Y$) at coupling strengths $c = \{0, 0.06, \ldots, 0.72, 0.78\}$. MGCiSN ($\Lambda$) for $N = 10000$

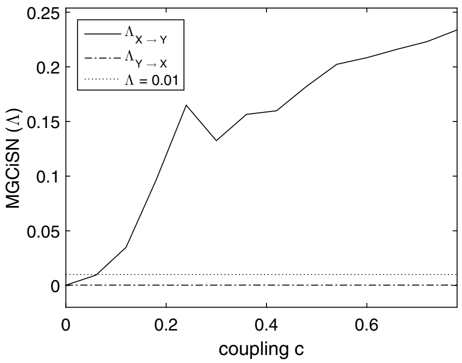
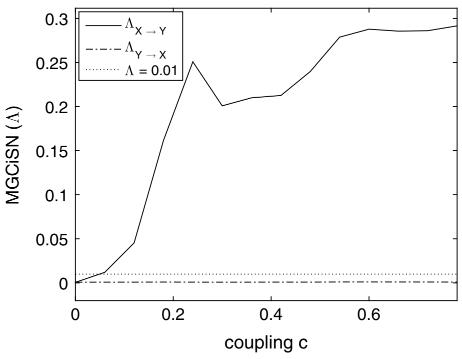


Fig. 2 Unidirectional coupling I (true causality $X \to Y$) at coupling strengths $c = \{0, 0.06, \ldots, 0.72, 0.78\}$. MGCiSN ($\Lambda$) for $N = 1000$



Unidirectional Nonlinear Coupling II

The next model represents two interacting nonidentical nonlinear time series with a nonlinear causal effect:

$$x(t) = 3.4x(t-1)[1 - x(t-1)^2]e^{-x(t-1)^2} + 0.6x(t-1), \tag{16}$$

$$y(t) = 3.4y(t-1)[1 - y(t-1)^2]e^{-y(t-1)^2} + 0.3y(t-2) + cx^2(t-2). \tag{17}$$

The coupling strength c varied from 0 to 2 with a step of 0.2. By construction, X has causal influence on Y for $c \neq 0$, and there is no causal connection between X and Y for $c = 0$.

Figures 3 and 4 show results of our second experiment. We observe that the MGCiSN successfully indicates the absence of causal influence of Y on X at any c for both sample sizes, but $\Lambda_{Y \to X}$ is close to chosen significance level 0.01 at c close to 2 for $N = 1000$. The MGCiSN correctly detected causal relationship from X to Y at $c > 0.04$ for $N = 10000$ and $N = 1000$.

Fig. 3 Unidirectional coupling II (true causality $X \to Y$) at coupling strengths $c = \{0, 0.2, \ldots, 1.8, 2\}$. MGCiSN ($\Lambda$) for $N = 10000$

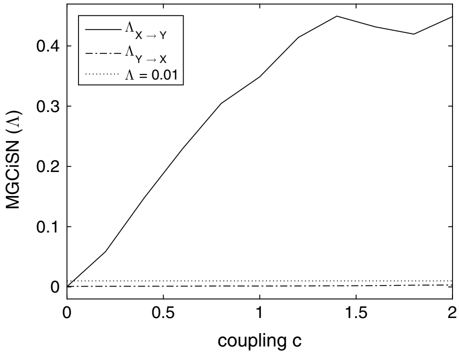
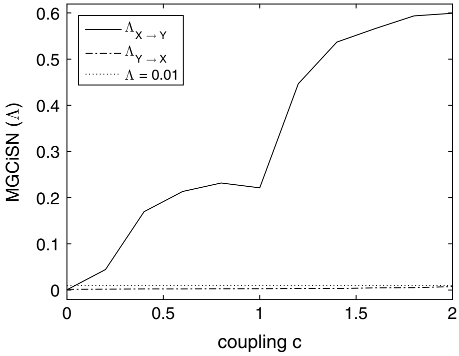


Fig. 4 Unidirectional coupling II (true causality $X \to Y$) at coupling strengths $c = \{0, 0.2, \ldots, 1.8, 2\}$. MGCiSN ($\Lambda$) for $N = 1000$



Nonlinear Causal Coupling III

The next model represents another example of two unidirectionally interacting non-identical nonlinear time series with a nonlinear causal effect:

$$x(t) = 3.4x(t-1)[1 - x(t-1)^2]e^{-x(t-1)^2} + 0.6x(t-1), \tag{18}$$

$$y(t) = 3.4y(t-1)[1 - y(t-1)^2]e^{-y(t-1)^2} + cy(t-2)x(t-2). \tag{19}$$

The coupling strength c varied from 0 to 1 with a step of 0.1. By construction, the variable X has causal influence on the variable Y for $c \neq 0$, and there is no causal connection between X and Y for $c = 0$.

Figures 5 and 6 show results of our third experiment. We observe that the MGCiSN successfully indicates the absence of the causal influence of Y on X at any c for both samples $N = 10000$ and $N = 1000$, $\Lambda_{Y\to X}$ is almost zero except the case $N = 1000$ at coupling strength close to 1. The causal relationship from X to Y is detected by the MGCiSN at $c > 0.22$ for $N = 10000$ and at $c > 0.16$ for $N = 1000$.

Fig. 5 Unidirectional coupling III (true causality $X \to Y$) at coupling strengths $c = \{0, 0.1, \ldots, 0.9, 1\}$. MGCiSN ($\Lambda$) for $N = 10000$

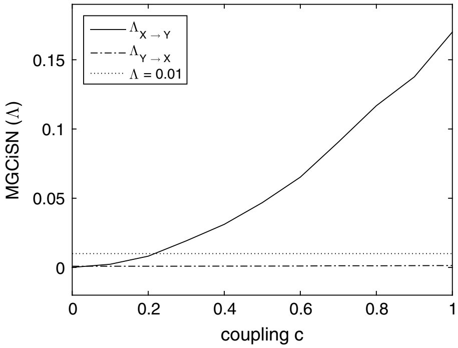
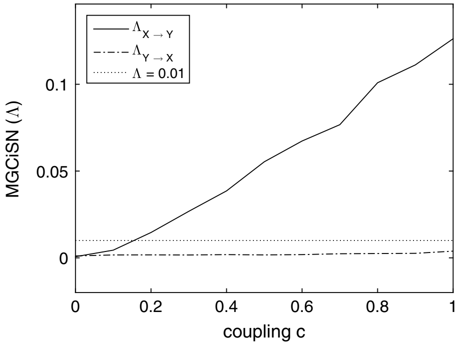


Fig. 6 Unidirectional coupling III (true causality $X \to Y$) at coupling strengths $c = \{0, 0.1, \ldots, 0.9, 1\}$. MGCiSN ($\Lambda$) for $N = 1000$



Bidirectional Nonlinear Coupling

For the forth example, we used bidirectionally coupled nonidentical Hénon maps (see, e.g., [11]):

$$x(t) = 1.4 - x(t-1)^2 + 0.3x(t-2) + c(x(t-1)^2 - y(t-1)^2), \qquad (20)$$

$$y(t) = 1.4 - y(t-1)^2 + 0.1y(t-2) + c(y(t-1)^2 - x(t-1)^2). \qquad (21)$$

The coupling strength c was varied from 0 to 0.2 with a step of 0.02. By construction, the variable X and the variable$Y$ are bidirectionally causally connected for $c \neq 0$, and they are independent for $c = 0$.

Figures 7 and 8 show results of our fourth experiment. We observe that the MGCiSN successfully detected correct causal relationship $X \to Y$ at all values of c for both samples $N = 10000$ and $N = 10000$. The MGCiSN detected the causal link $Y \to X$ only at $c > 0.07$ for considered sample sizes, but $\Lambda_{Y \to X}$ is nonzero at all values of c for both sample sizes.

For more experimental results, see [3].

Fig. 7 Bidirectional coupling (true causality $X \leftrightarrow Y$) at coupling strengths $c = \{0, 0.02, \ldots, 0.18, 0.2\}$. MGCiSN ($\Lambda$) for $N = 10000$

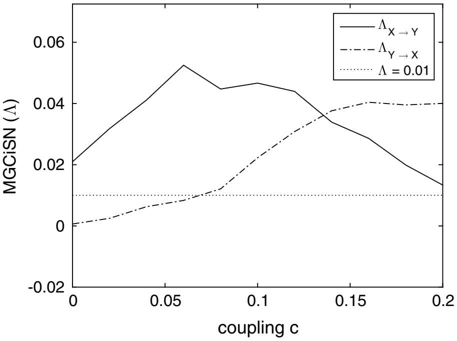
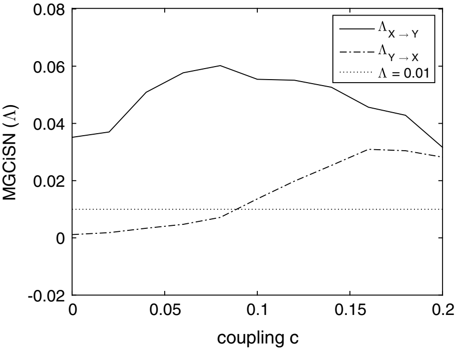


Fig. 8 Bidirectional coupling (true causality $X \leftrightarrow Y$) at coupling strengths $c = \{0, 0.02, \ldots, 0.18, 0.2\}$. MGCiSN ($\Lambda$) for $N = 1000$



Discussion

Detection of asymmetric and possibly nonlinear dependence between time series has application in a wide range of different sciences (e.g., climatology, neurophysiology, economy, etc.) and to avoid the false detection of causality is a major problem [14]. The suggested MGCiSN does not produce false-positive results and false-negative results are produced for very small coupling strengths. The MGCiSN has ability to capture nonlinear relationships without sample size limits. Although five parameters (τ, m_x, m_y, k, R^*) controlled the procedure, based on our experience, we reported that the choice $R^* = 0.95$ is the most influencing part of the method. If there is no neighborhood in the joint reconstructed phase space where $R^2 \geq R^*$, we cannot recommend the proposed MGCiSN for detecting causal dependencies between variables.

Conclusion

We present a novel causality estimator for two simultaneously recorded variables, the modified Granger causality in selected neighborhoods. MGCiSN is based on fitting linear regression models in local neighborhoods of the reconstructed joint dynamics and reflects the proportion of the predicted variation of the modeled variable by the past of the second variable only. The effectiveness of the suggested causal measure was studied on artificially generated nonlinear coupled nonlinear time series. The results showed that the MGCiSN works well for quantifying the asymmetric causal dependence of two nonlinear time series data, the sample size has a negligible influence on results. The computational properties and the experimental results indicate that the MGCiSN is potentially the first-choice technique in many cases to reveal causal relationship between two time series not appropriated for the Granger causality analysis.

Acknowledgements The work was supported by the Slovak Research and Development Agency, project APVV-15-0295, and by the Scientific Grant Agency VEGA of the Ministry of Education of the Slovak Republic and the Slovak Academy of Sciences, by the projects VEGA 2/0081/19 and VEGA 2/0054/18.

References

1. Barnett, L., Seth, A.-K.: The MVGC multivariate Granger causality toolbox: a new approach to Granger-causal inference. J. Neurosci. Methods **223**, 50–68 (2014)
2. Chen, Y., Rangarajan, G., Feng, J., Ding, M.: Analyzing multiple nonlinear time series with extended Granger causality. Phys. Lett. A **324**, 26–35 (2004)
3. Chvosteková, M.: Modified Granger causality in selected neighborhoods. In: Valenzuela, O., Rojas, F., Pomares, H., Rojas, I. (Eds.) ITISE 2018, International Conference on Time Series

and Forecasting, Proceedings of Papers, vol. 2, pp. 614–624. Godel Impresiones Digitales S.L. (2018). ISBN: 978-84-17293-57-4

4. Faes, L., Nollo, G., Chon, K.: Assessment of Granger causality by nonlinear model identification: application to short-term cardiovascular variability. Ann. Biomed. Eng. **36**, 381–395 (2008)
5. Fraser, A.-M., Swinney, H.-L.: Independent coordinates for strange attractors from mutual information. Phys. Rev. A **33**, 1134–1140 (1986)
6. Freiwald, W.-A., Valdes, P., Bosch, J., et al.: Testing non-linearity and directedness of interactions between neural groups in the macaque inferotemporal cortex. J. Neurosci. Methods **94**, 105–119 (1999)
7. Granger, C.-W.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica **37**, 424–438 (1969)
8. Kennel, M.-B., Brown, R., Abarbanel, H.-D.-I.: Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A **45**, 3403–3411 (1992)
9. Krakovská, A., Jakubík, J., Chvosteková, M., et al.: Comparison of six methods for the detection of causality in a bivariate time series. Phys. Rev. E **97**, 042207 (2018)
10. Krakovská, A., Hanzely, F.: Testing for causality in reconstructed state spaces by an optimized mixed prediction method. Phys. Rev. E **94**, 052203 (2016)
11. Lungarella, M., Ishiguro, K., Kuniyoshi, Y., Otsu, N.: Methods for quantifying the causal structure of bivariate time series. Int. J. Bifurc. Chaos **17**(3), 903–921 (2007)
12. Marinazzo, D., Pellicoro, M., Stramaglia, S.: Kernel method for nonlinear Granger causality. Phys. Rev. Lett. **100**(14), 144103 (2008)
13. Nicolaou, N., Constandinou, T.-G.: A Nonlinear causality estimator based on non-parametric multiplicative regression. Front. Neuroinformatics **10**, 19 (2016)
14. Paluš, M., Vejmelka, M.: Directionality of coupling from bivariate time series: How to avoid false causalities and missed connections. Phys. Rev. E **75**, 056211 (2007)
15. Shannon, C.-E.: A Mathematical theory of communication. Bell Syst. Tech. J. **27**, 379–423 (1948)
16. Sugihara, G., May, R., Ye, H., Hsieh, C.-H., Deyle, E., Fogarty, M., Munch, S.: Detecting causality in complex ecosystems. Science **338**, 496–500 (2012)
17. Takens, F.: Detecting strange attractors in turbulence. In: Rand, D.A., Young, L.-S. (eds.) Dynamical Systems and Turbulence. Lecture notes in Mathematics, vol. 898, pp. 366–381. Springer (1981)

Computing Environment for Forecasting Based on System Dynamics Models

Radosław Pytlak, Damian Suski, Tomasz Tarnawski, Zbigniew Wawrzyniak, Tomasz Zawadzki and Paweł Cichosz

Abstract The paper proposes the computing environment which allows building forecasts based on System Dynamics models. The environment is equipped with GUI for building dynamic models according to Forrester's methodology. The unique feature of the application is that the created models can be calibrated with the help of optimization procedures tailored for solving nonlinear least squares problems with differential–algebraic equations. Furthermore, the application enables verification of decision rules inherited in System Dynamics models by solving problems associated with the models dynamic optimization—then the model with optimal decision rules can be simulated to build forecasts of interests. To illustrate the functionalities of the environment, the example of the model of drug prevalence is discussed in some detail.

Keywords System Dynamics · Trajectories forecasting · Decision rules · Optimal control

R. Pytlak (✉)
Faculty of Mathematics and Information Science, Warsaw University of Technology,
Koszykowa 75, 00-662 Warsaw, Poland
e-mail: r.pytlak@mini.pw.edu.pl

D. Suski
Faculty of Mechatronics, Warsaw University of Technology, Warsaw, Poland
e-mail: suski@mchtr.pw.edu.pl

T. Tarnawski
Kozminski University, 03-301 Warsaw, Poland
e-mail: ttarnawski@kozminski.edu.pl

Z. Wawrzyniak · P. Cichosz
Faculty of Electronics and Information Technology, Warsaw University of Technology,
Warsaw, Poland
e-mail: magic2k@wp.pl

T. Zawadzki
War Studies University, 00-910 Warsaw, Poland
e-mail: tzawa@o2.pl

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_4

Introduction

System Dynamics offers an approach to forecast that is essentially different from the one based on econometric models, as it allows to include decision rules (which are dynamical in nature) within the model. Building predictions boils down to running new simulations of model equations with appropriately changed parameters describing the decision rules. Also, nonlinear functions are a common thing in multi-equation System Dynamics models while multi-equation nonlinear econometric models are a rarity due to extreme difficulties in their building.

The quality of quantitative predictions drawn from System Dynamics models—in the form of generated trajectories of selected variables—depends heavily on the specification of values of the model's parameters. If our predictions are to possess any merit, we must be certain that the simulated trajectories of interest are close to these which have been observed. This can be achieved in the process of model calibration, provided that the structure of model's equations are not far from the actual dynamics of the real process. Model calibration can refer to statistical methods to measure quality of parameters estimation, but the final model and its use in predicting future values of model's variables is deterministic. The model calibration is based on optimization tools; however, there are other applications of optimization within the process of building dynamical models, such as finding adequate structures of model's decision rules or choosing proper (parameters') values for these decision rules.

The paper describes a newly implemented computing environment that brings together System Dynamics (SD) graphical modeling methodology, simulation of the model's behavior, defining optimal control problems (by supplementing the SD model with objective function), and advanced numerical algorithms for solving various classes of dynamic optimization problems. It was designed to make it possible—define and solve a number of related problems defined around the same model. This way one can start with specifying a dedicated problem of calibrating the model's parameters (solved by selecting parameters' values for which the distance between empirical and generated trajectories is minimized). Once the model is calibrated, it can then be used to build another optimization problem dealing with choosing control strategies that would optimize (some measures of) certain aspects of the system's behavior. The environment makes it, therefore, possible to produce forecasts for the values of the system's variables under different decision rules and helps the user select such decisions for which the foreseen behavior is most preferable.

The proposed software package can be compared to a number of existing solutions, such as Vensim ([1] or any of similar SD simulation applications), OpenModelica [2], or JModelica.org [3]. However, we believe that our solution is unique as it exceeds the capabilities of these competitors. In comparison with Vensim, it not only offers simulation (and does that with a significantly wider selection of numerical integration engines) but also provides support for dynamic optimization, which is completely lacking in the Vensim class of System Dynamics simulation packages. In that respect, it is much closer in kind to the Modelica-related applications with built-in Optimica language support, i.e., OpenModelica or JModelica.org (Optimica is a close relative to Modelica with slight syntax extensions for defining optimal control problems; it was initially proposed in [4]). What sets us apart from the two is the designed-in support for a wide range of various optimization solvers (as described, e.g., in [5])—while both implementations, of OpenModelica and JModelca.org, are based solely on the collocation algorithm—which makes them less appropriate for certain classes of problems, calibration in particular.

In our environment, the models can either be built by means of System Dynamics standard graphical notation or through text files written in a dedicated language—DOML. The Dynamic Optimization Modeling Language (DOML) was initially proposed in [5] as a Modelica-based programming language-independent communication format while developing Interactive Dynamic Optimization Server (IDOS, described therein). The DOML format provides a mean for defining dynamic systems (i.e., described by systems of Ordinary Differential Equations or Differential–Algebraic Equations) together with user controls and objective functions that together compose optimal control problems.

The paper is organized as follows. The following section "System Description" provides a general description of the graphical and computational environment. Section "Problem Formulation" proceeds toward formulating an exemplary dynamical model that illustrates its capacities and can be used for forecasting the behavior of a system of interest. The example model deals with forecasting drug use, at a national level, under different strategies undertaken by law enforcement authorities to combat drugs. Additional attention is devoted to the specification of the system by means of the DOML language. Next, in section "Numerical Procedures for Optimal Control Problems", the numerical algorithms implemented within the environment and used in this particular case are briefly described, together with discussion of the related aspects of calibrating the model and specifying decision rules. Finally, section "Discussion of Results" provides a discussion of simulation and optimization results yielding trajectories showing the foreseen behavior of the system's variables.

System Description

The system that has been developed to solve optimization problems consists of three main components:

- modeling application
- DOML compiler
- simulation and optimization libraries.

and their interrelationship is schematically shown in Fig. 1.

The modeling component called JOptisim allows users to create graphical models according to System Dynamics methodology which introduces three fundamental elements of SD models, namely stocks, flows, and auxiliary variables (or parameters)—in that respect it is comparable to similar graphical tools such as Vensim and Stella as all model's elements can be graphically placed, removed, renamed,

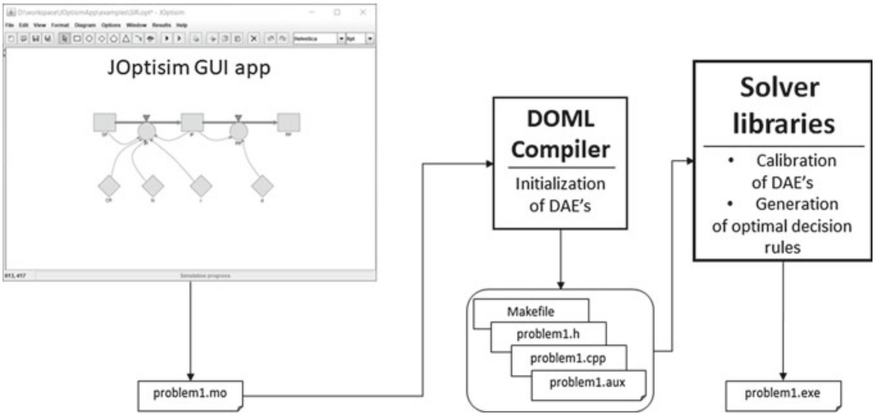


Fig. 1 Composing elements of the computing environment

or copied at any time and then saved to and restored from external file. JOptsim also runs preliminary simulations to check the correctness of the model. The implemented simulation procedure is based on advanced RADAU5 numerical code which enables to track trajectories with steep changes, especially observed in so-called stiff equations.

Besides creating a standard System Dynamics models, the application allows to define optimization problem by filling additional values in a dedicated window. The window supports a definition of horizon used in optimization problem, objective function, and constraints of optimization problem. Such created model can be saved in an external file for further editing or exported to the DOML language file either as ordinary differential or differential–algebraic equations. The resulting DOML files are sent to Modelica-based simulation and optimization engine composed of DOML compiler (as the first stage) and a number of dedicated numerical libraries.

The compiler was fitted with a number of additional features needed to implement the postulated extensions to the standard Modelica language. The extensions and their implementations were discussed in detail in a number of articles (see, e.g., [5, 6]) but in short, they boil down to:

– provide a mechanism for choosing among a number of different solver packages applicable to (different classes of) dynamic optimization problems;
– allow for solver chaining—i.e., using a sequence of solvers on the same problem, where the solution from one (usually more robust but less accurate) can be used in the next (that, e.g., provides higher accuracy but requires a relatively good starting point);
– introduce labeling of equations (and constraints) together with a way for defining so-called adjoint variables, which are necessary when using certain class of optimization algorithms.

To our mind, these elements make the language significantly better fit for handling the vast variety of optimal control problems out there.

The compiler transcribes the DOML definition of a problem into a regular programming language (in most cases being C++, but in some cases, it is C, Fortran of R). The generated code includes appropriately woven calls to external numerical libraries needed to solve a particular problem. The environment setup is prepared to handle various cases as it is equipped with several sophisticated solvers for dynamic optimization problems: based on "a priori discretization" of differential equations; solvers using adjoint equations; shooting procedures. Second, the environment includes state-of-the-art numerical libraries dealing, e.g., with numerical integration or automatic differentiation–optimization problems can have many differential equations and providing Jacobians of the right-hand side of these equations is impractical (and these Jacobians are needed to solve optimal controls problems efficiently).

Problem Formulation

The exemplary problem we consider is the determination of the optimal time distribution of police forces intended for fighting against drugs. The calculations are based on the drug prevalence model built with the help of System Dynamics methodology. The model is based on the drug prevalence model presented in [7] and additional contribution related to drug market in Poland depicted in [8].

The appropriate model is presented in Fig. 2 and we will now briefly describe the structure of the model. The core of the model is the flow of people between two groups of people: people who have used drugs during the previous year (*PastYearUsers*); people who have not used drugs during the previous year (*NotPastYearUsers*). The

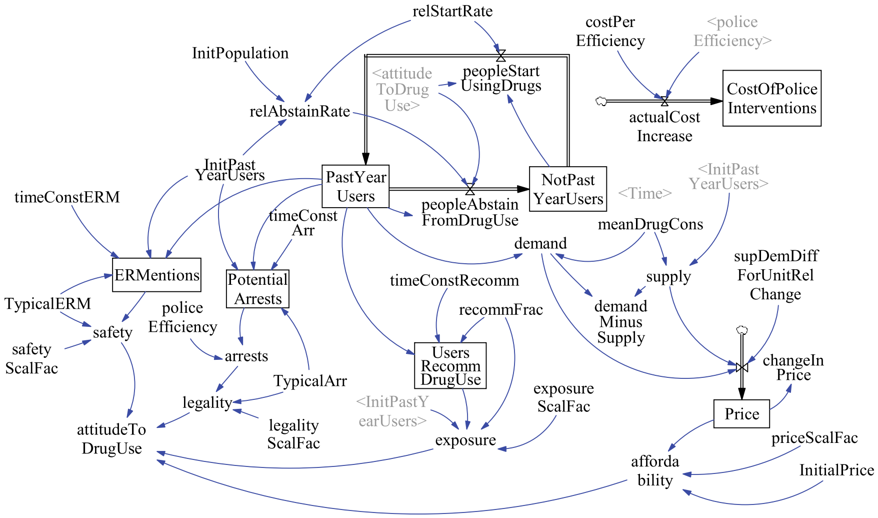


Fig. 2 System Dynamics model of drug prevalence

flow of people between these two groups consists of two subflows: flow of people who have started using drugs during the previous year (*peopleStartUsingDrugs*); flow of people who have abstained from drug-using during the previous year (*peopleAbstainFromDrugUse*). The sizes of these flows depend on the sizes of populations and the positive attitude to drug use among the society, which is modeled by one aggregated variable *attitudeToDrugUse*. In our model, the attitude to drug use depends on the following factors: exposure to drugs through social contacts (*exposure*); perception of health consequences of drug use (*safety*); perception of consequences of breaking the drug law(*legality*); affordability of drugs (*affordability*).

The exposure to drugs through social contacts depends on the prevalence of drugs in the population and it increases the positive attitude to drug use. In our model, we assume that the perception of health consequences of drug use depends on the number of medical interventions associated with using drugs (*ERMentions*). With the increasing number of medical interventions, the positive attitude to drug use decreases. The number of medical interventions depends directly on the prevalence of drugs in the population. The perception of law consequences of drug possession and distribution depends on the number of arrests (*arrests*). With the increasing number of arrests, the positive attitude to drug use decreases. The number of arrests depends on the prevalence of drugs in population as well as on the efficiency of police (*policeEfficiency*). The police efficiency models a general ability of police to fight against drugs and it is assumed to be the only control variable in our exemplary problem.

Drugs affordability depends on the drug unit price (*Price*). With the increasing price, the positive attitude to drug use decreases. To describe the dependence of drug unit price on the drug prevalence, a simple economic modeled is used. It is assumed that the changes in drug unit price depend on the actual price and the difference between drug demand and supply. Drug demand depends on the drug prevalence and drug supply is modeled as an external function of time. It is possible to utilize more complicated economic models, but for our needs, the proposed model gives satisfactory results.

Our goal is to determine the optimal distribution of police forces over the assumed time interval to achieve the best results in fighting against drugs. In our model, the optimized control variable is *policeEfficiency*, which represents the efficiency of the police in arresting people breaking the drug law. *policeEfficiency* variable has been introduced to our model arbitrary and does not possess a simple and direct interpretation. Nevertheless, it is possible to associate it with other measurable quantities such as a number of police patrols per week or the drug fight budget per year. In any case, the methodology remains the same.

To test the computational environment, we solved the exemplary optimal control problem. The parameters of the drug prevalence model have been adjusted according to data regarding the cocaine prevalence in Poland provided by European Monitoring Centre for Drugs and Drug Addictions. In the considered optimal control problem, we assume that at each time moment the *policeEfficiency* stays within interval [0, 2]. This constraint models, e.g., the limited number of policemen available at each moment. Next, we assume that police interventions generate costs that are integrated over time. The variable *CostOfPoliceInterventions* represents the summary cost of all police interventions up to a moment t_1

$$CostOfPoliceInterventions(t_1) = \int_0^{t_1} costPerEfficiency \cdot policeEfficiency(t)dt. \tag{1}$$

where $costPerEfficiency = 1$. We impose the additional constraint on the total cost of police interventions over the whole time interval $[0, t_1] = [0, 20]$ years

$$CostOfPoliceInterventions(t_1) \leq 20 \tag{2}$$

The above inequality represents the budget constraints as the *policeEfficiency* cannot always assume the maximum value 2, for which the variable *CostOfPoliceInterventions* would reach a value 40. It is therefore required to distribute the police forces somehow over the whole time interval. The proposed cost function, which is supposed to be minimized, is the number of drug users at the final time $PastYearUsers(t_1)$.

The model and the optimization tasks have been implemented as a DOML script [5]. The header of the DOML script defines the cost function and the time interval.

```
optimization drugs_opt ( objective = PastYearUsers(finalTime),
startTime = 0, finalTime = 20.0 )
```

Next all the variables in the model are declared.

```
parameter Real InitialPrice= 45;
...
parameter Real TypicalERM= 100;
Real affordability;
...
Real arrests;
input Real policeEfficiency(initialGuess = 1, min = 0, max = 2);
Real PotentialArrests(start = 133.33333333333334);
...
Real Price(start = 22.5);
```

There are four types of variables: constant parameters, casual variables, input variables, and state variables, which correspond to stock variables in System Dynamics convention [9]. Besides the declaration, for input variables, the initial guess and bounds are defined and for state variables, the initial value is defined.

In the `equation` section of DOML script system equations are defined. Differential equations define derivatives of state variables, whereas algebraic equations define the remaining variables.

```
equation
  der(Price) = (changeInPrice) - 0;
  ...
  0 = - safety + (TypicalERM*safetyScalFac)/ERMentions;
```

In the `constraint` section, the remaining constraints are defined

```
constraint
  CostOfPoliceInterventions(finalTime) <= 20;
```

Numerical Procedures for Optimal Control Problems

There are three types of optimization problems linked with the methodology our computing environment realizes. The first two are associated with models calibration. We assume that the calibration of a System Dynamics model should be performed in two stages. In the first one, one looks for functions which describe the relationship in a given node of a CLD diagram. This calibration reduces, in fact, to solving a standard nonlinear (in general) least squares problem and can be accomplished by a variant of the Gauss–Newton method. In that case, we use data which represent variables linked to a particular node.

The model calibration in the second stage is needed to construct an adequate model for forecasting with respect to a particular variable which is a solution to DAEs representing System Dynamics model. In that case, the calibration procedure is much more elaborate since it takes into account the entire model dynamics and so it requires a tailored dynamic optimization procedure based on a least squares objective function.

Then, we apply optimal control to verify (construct) decision rules which are inherently present in a System Dynamics model. In that case, we solve a standard optimal control problem having in mind that a procedure for solving that problem must satisfy the following conditions: (1) it should tackle control problems described by large-scale differential–algebraic equations; (2) it must have an initialization procedure for algebraic variables which is called at initial time and at times at which controls exhibit jumps.

The first requirement is a direct consequence of the System Dynamics methodology. The dynamic model in the form of differential–algebraic equations is of special form. Algebraic variables are functions of other algebraic variables and differential variables (those variables which are solutions to differential equations). In the literature on differential–algebraic equations of this form are called semi-explicit index one differential–algebraic equations [10].

The index one means in this context that algebraic equations can be uniquely expressed as functions of differential variables. And this is really the case of System Dynamics models which can be transformed into first-order ordinary differential equations. However, in general, we do not want to do that since we would like to have the optimal control problem defined in terms of algebraic and differential equations in order, for example, to express some constraints in the problem as functions of algebraic variables which have their meanings in the model.

The solver we use to solve optimal control problems with System Dynamics models is described in detail in [11]. The solver has several features which make it suitable for control problems with System Dynamics models. It is based on a continuous-time approach to optimal control algorithms and thus it is able to solve accurately control problems with state constraints [11]. On the other hand, it uses adjoint equations to evaluate gradients, so it can be used to solve optimal control problems with many differential–algebraic equations which often occur in the context of System Dynamics modeling.

Nonlinear Least Squares Calibration of Dynamical Systems

In order to explain our approach to the calibration of dynamical models, consider first the static nonlinear least squares problem:

$$\min_{p \in \mathcal{R}^{n_p}} \left[f(x) = \frac{1}{2} \|g(p)\|^2 = \frac{1}{2} \sum_{i=1}^{m} ((g_i(p))^2 \right] \tag{3}$$

where $g(p) = [g_1(p), g_2(p), \ldots, g_m(p)]^T$.

Important feature of the Gauss–Newton method for the nonlinear least squares problem is that the Hessian matrix of the objective function (3) can be effectively approximated by the matrix based on the Jacobian matrix related to the transformation g—see [12] for details:

$$\nabla^2 f(p) \approx J(p)^T J(p). \tag{4}$$

Here, $J(p)$ is the Jacobian matrix of the transformation g.

Let us adopt the Gauss–Newton procedure for the one which is suitable for calibrating dynamical systems with continuous-time dynamics described by DAEs.

Suppose that a system is represented by the equations

$$F(\dot{x}(t), x(t), u(t), p) = 0, \quad t \in [0, t_f], \tag{5}$$

where vector p represents system parameters. The solution to system (5) is dependent on p and thus we denote it by x^p.

We approximate the parameters p by solving the nonlinear least squares problem

$$\min_{p} \left[MSE_l(p) = \frac{1}{N} \sum_{j=1}^{N} \left(x_l^p(t_j) - \hat{x}_l(t_j) \right)^2 = \frac{1}{N} \sum_{j=1}^{N} MSE_l^j(p) \right] \tag{6}$$

subject to the constraints (5). Here, we recall, that $\hat{x}_l$ is the empirical (measured) trajectory.

That problem can be solved by nonlinear programming techniques provided that we can evaluate gradients of $MSE_l(p)$.

On the other hand, the gradient of the functional $MAE_l(p)$ may be evaluated with the help of adjoint equations if we observe that $MSE_l(p)$ is composed of N functions $MSE_l^j(p)$.

The gradient of $MSE_l^j(p)$ can be calculated by using the adjoint equations for the system (5) and the function $MSE_l^j(p)$—it means that these adjoint equations will be integrated from time t_j (see, for example, [11] on the use of adjoint equations in dynamic optimization and [6] on their application to this particular function). Then, $\nabla MSE_l(p) = \sum_{j=1}^{N} \nabla MSE_l^j(p)$.

The adjoint equations for the considered objective function we present for the case when system equations are in the form of ordinary differential equations

$$\dot{x}(t) = f(x(t), t, p),\ t \in [0, t_f],\ x(0) = x_0. \tag{7}$$

Then

$$\nabla MSE_l^j(p) = \int_0^{t_j} f_p(x^p(t), t, p)^T q_j(t) dt, \tag{8}$$

where q_j is the solution to the adjoint equations

$$q_j(t) = 0,\ t \in (t_j, t_f] \tag{9}$$

$$q_j(t_j) = \Delta_j \tag{10}$$

$$\dot{q}_j(t) = -f_x(x^p(t), t, p)^T q_j(t),\ t \in [0, t_j). \tag{11}$$

Here,

$$\Delta_j = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 2(x_l^p(t_j) - \hat{x}_l(t_j)) \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Having objective function values and its gradients, we can build an optimization procedure for model calibration. A general scheme for such a procedure may look like stated below.

General Calibration Procedure (GCP)

1. Set initial values of parameters: p_1 and set $k = 1$.
2. For parameters p_k, calculate system trajectories x^{p_k} by numerically integrating system equations. On that basis, determine objective function value $MSE_l(p_k)$ through values $MSE_l^j(p_k)$, $j = 1, \ldots, N$.

3. Having trajectories x^{p_k} and values $MSE_l^j(p_k), j = 1, \ldots, N$ solve adjoint equations and determine $\nabla MSE_l^j(p_k)$, $j = 1, \ldots, N$, $\nabla MSE_l(p_k)$.
4. Determine the direction of descent p_k using some optimization procedure.
5. Perform directional minimization with the help of the optimization procedure to evaluate the step size α_k. Substitute $p_k + \alpha_k d_k$ for p_{k+1}. increase k by one and go to Step 2).

That general calibration scheme covers also a Gauss–Newton approach to a model calibration. However, in this case, we need to specify how the direction of descent d_k is determined on the basis of vectors $\nabla MSE_l^j(p_k)$, $j = 1, \ldots, N$ and $\nabla MSE_l(p_k)$. In this case, we use the relations:

$$J_l(p_k) = \begin{bmatrix} \nabla MSE_l^1(p_k)^T \\ \nabla MSE_l^2(p_k)^T \\ \vdots \\ \nabla MSE_l^N(p_k)^T \end{bmatrix}, \; H_l(p_k) = J_l(p_k)^T J_l(p_k)$$

$$p_{k+1} = p_k - \alpha_k \left[H_l(p_k)\right]^{-1} \nabla MSE_l(p_k). \tag{12}$$

Algorithm for Optimal Control Problems with SD Models

In general, System Dynamics models are described by differential–algebraic equations (DAEs). Therefore, we should consider the following optimal control problem $\mathbf{P_{DAE}}$:

$$\min_u \phi(x(t_f)),$$

subject to the constraints:

$$F(\dot{x}(t), x(t), y(t), u(t), t) = 0 \text{ a.e. on } T, \; x(0) = x_0 \tag{13}$$

$$q(t, x(t)) \leq 0 \;\; \forall t \in T \tag{14}$$

$$h_i^1(x(t_f)) = 0 \;\; \forall i \in E \tag{15}$$

$$h_j^2(x(t_f)) \leq 0 \;\; \forall j \in I \tag{16}$$

$$u \in \mathcal{U} = \{u : \; u(t) \in \Omega \text{ a.e. on } T\}. \tag{17}$$

Here, $x(t) \in \mathcal{R}^{n_d}$, $y(t) \in \mathcal{R}^{n_a}$, $u(t) \in \mathcal{R}^m$, $n = n_d + n_a$, and Ω is a convex compact set. We assume that for any x_0, $u \in \mathcal{U}$ there exists a unique solution to (13): (x^u, y^u). We call x a *differential* state and y an *algebraic* state.

The problem we have to consider has both terminal constraints (15)–(16) (equality and inequality), state constraints (14) (constraints which must be satisfied at every point of time horizon) and hard constraints on controls (17). The System Dynamics equations (13) contain controls u which are introduced to System Dynamics to verify the optimality of decision rules used in the System Dynamics model. Once the problem $\mathbf{P_{DAE}}$ is solved, we build once again (possibly new, but almost surely with new parameters) decision rules and new forecasts on the basis of these rules.

Essentially the above optimal control problem can be numerically tackled in the very similar way as the problem described by ODEs, in particular, due to the uniqueness assumption stated above, it can be stated in *reduced space* (see [11] for details). In fact, our computing environment applies a numerical procedure stated in [11]. It is the procedure which uses a RADAU IIA procedure (it is an implicit Runge–Kutta method) for the integration of the system equations (13)—the justification for the choice is given in [11] (Chap. 6). Having trajections of system equations obtained by numerical integration gradients of all functionals defining the problem $\mathbf{P_{DAE}}$ are evaluated in the reduced space with the help of adjoint equations associated with system equations (Chap. 6 of [11]).

Since our system equations are in the form of DAEs, we need a procedure for their initialization—usually we have initial values of differential states x but algebraic variables y have to be found so algebraic equations are satisfied at initial time. Due to the way System Dynamics equations are constructed (with the help of the GUI), the functions $y_i(t) = init(x^u(t), u(t), i)$, $i = 1, \ldots, n_a$ can be built—these functions are created while processing model equations by the DOML compiler.

Discussion of Results

Several optimal control problems based on System Dynamics have been defined and solved. In [13], an optimal control problem related to sanitary teams activities during an epidemics of foodborne disease is discussed. In the problem, decision variables (controls) are linked with the number of sanitary teams which try to contain the epidemics by isolating infected people and by eliminating contaminated food. The optimal control problem is further discussed in [14] where decision rules are derived for sanitary teams activities on the basis of the optimal solution to the problem.

The optimization results presented in the section are related to the System Dynamics model of drug prevalence stated in section "Problem Formulation". They should be treated as preliminary results since additional experiments with the optimization model (and its variants) are needed to propose new decision rules for the distribution of police forces which fight against drugs use. On the basis of these decision rules, we could then forecast the spread of drugs use as described by the model variable $PastYearUsers$.

The numerical procedure presented in section "Numerical Procedures for Optimal Control Problems" has found the optimal solution shown in Fig. 3 (one optimal state trajectory is given in Fig. 4). The analysis of the obtained solution suggests the

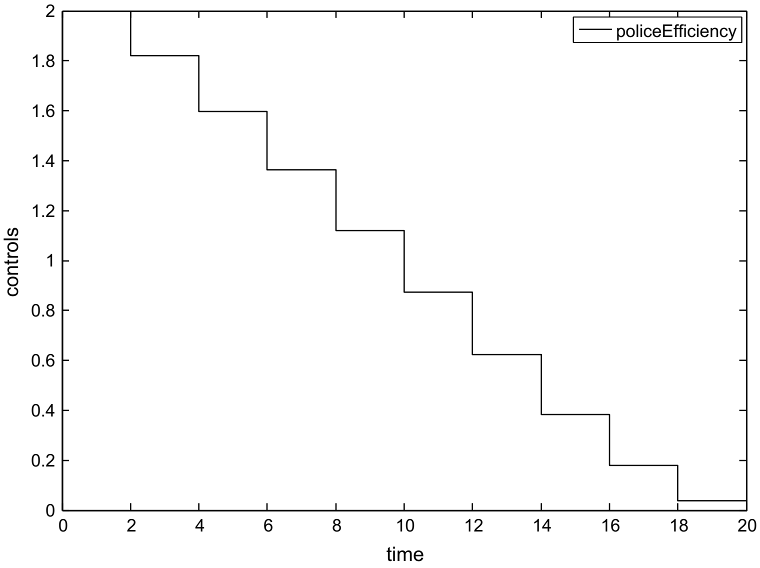


Fig. 3 Optimal trajectory for the control *policeEfficiency*

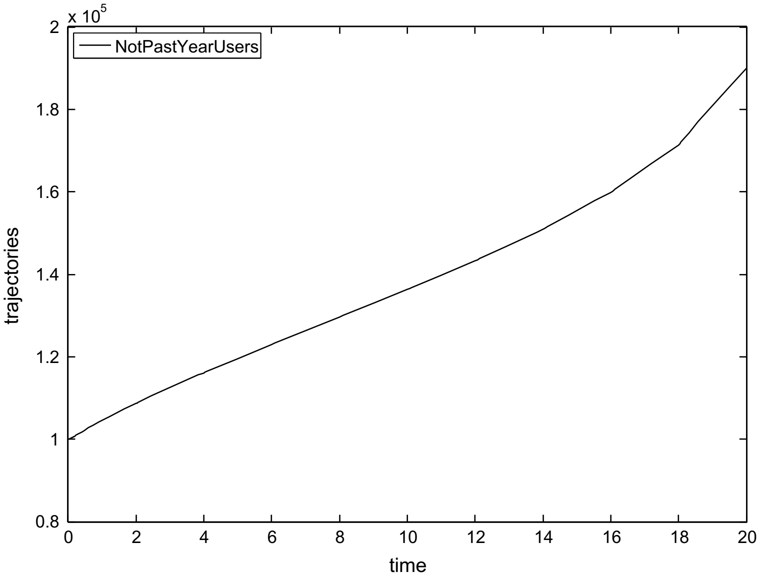


Fig. 4 Optimal trajectory for the control *NotPastYearUsers*

following strategy for the use of police forces: use the highest possible efficiency in the beginning and then reduce it linearly to its lowest possible level at the end of the planning horizon—the strategy has, in fact, one parameter which could be determined by a simple sweeping procedure.

Acknowledgements The work was partially funded by the grant DOB-BIO7/05 /02/2015 of Polish National Office for Research and Development.

References

1. Vensim site, Ventana Systems. http://vensim.com/, June 2018
2. OpenModelica site, Open Source Modelica Consortium (OSMC). http://www.openmodelica.org/, June 2017
3. Open Source Project: JModelica.org. http://www.jmodelica.org/, April 2017
4. Åkesson, J.: Tools and languages for optimization of large-scale systems. Ph.D. thesis, Department of Automatic Control, Lund University, Lund, Sweden (2007)
5. Pytlak, R., Tarnawski, T., Fajdek, B., Stachura, M.: Interactive dynamic optimization server-connecting one modeling language with many solvers. Optim. Methods Softw. **29**, 118–1138 (2014)
6. Pytlak, R., Tarnawski, T.: IDOS—(also) a web based tool for calibrating modelica models. In: Proceedings of the 10th International Modelica Conference, 10–12 March , Lund, Sweden, pp. 1095–1104 (2014)
7. Homer, J.B.: A system dynamics model of national cocaine prevalence. Syst. Dyn. Rev. **9**(1), 49–78 (1993)
8. Hołyst, B., Wawrzyniak, Z.M., Wiciak, K.: Prognozowanie kryminologiczne w wymiarze społecznym, Tom I–Metodologia, Analiza, Tendencje rozwojowe, vol. 1. Wydawnictwo Naukowe PWN SA (2017) (in Polish)
9. Sterman, J.D.: Business Dynamics, Systems Thinking and Modeling for a Complex World. The McGraw-Hill Companies, Inc (2000)
10. Brenan, K.E., Campbell, S.L., Petzold, L.R.: Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. SIAM (1996)
11. Pytlak, R.: Numerical Methods for Optimal Control Problem With State Constraints. Lecture Notes in Mathematics, vol. 1707. Springer, Berlin (1999)
12. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (1999)
13. Pytlak, R., Suski, D., Zawadzki, T., Stecz, W.: Decision support system for sanitary teams activities. Int. J. Decis. Support Syst. Technol. **3**, 23–45 (2014)
14. Pytlak, R., Zawadzki, T.: A New Approach to Optimization of a System Dynamics Model. Submitted for publication
15. Bryson, A.E., Ho, Y.: Applied Optimal Control: Optimization, Estimation and Control. Taylor & Francis, New York (1975)
16. Pytlak, R., Zawadzki, T.: On solving optimal control problems with higher index DAEs. Optim. Methods Softw. **29**, 1139–1162 (2014)

Novel Order Patterns Recurrence Plot-Based Quantification Measures to Unveil Deterministic Dynamics from Stochastic Processes

Shuixiu Lu, Sebastian Oberst, Guoqiang Zhang and Zongwei Luo

Abstract Forbidden ordinal patterns are known to be useful to discriminate between chaotic and stochastic systems. However, while uncorrelated noise can be separated from deterministic signals using forbidden ordinal patterns, correlated noise exhibits apparently forbidden ordinal patterns, which can impede distinguishing noise from chaos. Here, we introduce order patterns recurrence plots to visualise the difference among deterministic chaotic systems, and stochastic systems of uncorrelated and correlated noise. In an order pattern plot of a chaotic system with an optimal embedding dimension, the diagonal lines remain preserved, while uncorrelated noise shows up as thinly isolated dots and correlated noise forms clusters. We propose two measures, the mean and the median of relative frequencies of order patterns that appear in a time series to distinguish those dynamics. The effectiveness of the two measures is analysed using bifurcation diagrams of the logistic map, the tent map, the delayed logistic map and the Hénon map. Our results show, that both, the mean and the median, distinguish chaos from quasiperiodicity in the delayed logistic map. The mean of relative frequencies of order pattern is reciprocal to the number of order patterns that occur in a given time series and thus can be a measure of forbidden structures—which becomes unbounded. While the mean is robust to the change of parameters in the bifurcation diagrams, the median exhibits sensitive changes, which is significant to characterise chaotic signals.

S. Lu (✉) · S. Oberst (✉) · G. Zhang
Centre for Audio, Acoustics and Vibration (CAAV), University of Technology Sydney, Ultimo, NSW 2007, Australia
e-mail: shuixiu.lu@student.uts.edu.au

S. Oberst
e-mail: sebastian.oberst@uts.edu.au

G. Zhang
e-mail: guoqiang.zhang@uts.edu.au

S. Lu · Z. Luo
Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen, China
e-mail: luozw@sustc.edu.cn

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_5

Keywords Forbidden ordinal patterns · Permutations · Noise

Introduction

Given a measured time series, irregularity often dominates. The irregularities can emerge in economics [18], finance [31] and natural sciences (physics [10], biology [27], chemistry [41], geology [25], medicine [44] or engineering [28, 40, 43]). For real data, it is often necessary to distinguish between stochastic dynamics (random noise, high dimensional) and deterministic dynamics which can also be chaotic, of either low or high dimensions.

However, both a stochastic system and a chaotic system generate a broadband power spectrum in the frequency domain, which often hinders their disentanglement [34]. The situation complicates when it comes to real-life data or natural information (life sciences) since measurements here are inextricably contaminated by omnipresent dynamical or observational noise [17, 26, 29, 34].

To discriminate chaotic systems with observational noise from stochastic systems, Bandt and Pompe [6] developed a symbolisation scheme, the Bandt–Pompe(BP) methodology, which encodes a time series into order patterns. An order pattern with embedding dimension m represents a permutation of the set $\{0, 1, ..., m-1\}$, which is an invariant under the process of monotonous transformations and known to be robust with regards to observational Gaussian noise [6]. Owing to this robustness, order patterns have attracted a growing interest [7, 16, 32, 33, 47].

Two streams of studies have been devoted to quantify order patterns and the degree of determinism on the basis of the BP method. The first stream is that of *forbidden ordinal patterns* [3, 4, 36, 37, 45, 46, 48], which are non-occurring permutations in a time series [2, 4, 9, 17, 35].

The occurrence of forbidden ordinal patterns are related to structures of deterministic processes. For a deterministic map $x_{i+1} = f(x_i)$, the number of intersections of $f^0(x) = x$, $f^1(x)$, ..., $f^{m-1}(x)$ determines the number of order patterns. As a result, when the number of intersections is smaller than $m!$ (and exponentially growing with m), the map exhibits a *forbidden pattern* [4, 5, 36].

However, there is a minimal embedding dimension to detect the outgrowth of the forbidden ordinal patterns [36]. For the logistic map $(x_{i+1} = 4x_i(1-x_i))$, as $m = 3$, the number of intersections of f^0, f^1, f^2 is 6, hence the number of order patterns equals to 5 and the permutation (2, 1, 0) is forbidden, irrespective of the time series' length [4]. Also, the number of forbidden ordinal patterns is robust to low degrees of sampling irregularities as the time series is generated from irregular time intervals [16].

To visualise forbidden ordinal patterns, Kulp and Smith [15], Kulp and Zunino [17] introduce a permutation spectrum test, which plots the counts of frequencies of each permutation of $\{0, 1, ..., m-1\}$. A permutation that counts 0 is a forbidden ordinal pattern and indicates determinism. However, this plot fails to discriminate determinism from correlated noise, which is stochastic but shows missing order patterns. Both forbidden and missing order patterns are pattern structures that do not show up in a time series. However, while the forbidden order patterns is unrelated to the length of the time series, the missing order patterns have a small probability to show up as long as the time series is long enough. Permutation entropy is further needed to facilitate the identification of determinism based on this plot [30, 35, 36].

On the other hand, the second stream of measures that applied the BP method is to visualise and estimate the recurrent behaviour of order patterns, using order patterns recurrence plot (OPRP). OPRP was developed by Groth [13] to visualise the dependencies between two time series. While a conventional recurrence plot (RP) is based on the (phase) spatial and temporal closeness to define recurrent and laminar behaviour [21], a recurrence in an OPRP is that the same order pattern recurs [8, 13].

However, the literature is scarce on interpreting recurrence quantification analysis (RQA) measures extracted from an OPRP [8]. Schinkel et al. [38] and Marwan et al. [20] employ the RQA to detect determinism of event-related potentials of brain responses using electroencephalography (EEG). Donner et al. [11] use RQA to investigate short-term dynamics of discrete-valued data. McCullough et al. [22] regenerate a time series using an ordinal network to compare the dynamics of the original time series and a surrogate time series designed from ordinal network data using the RQA and OPRP.

While the aforementioned OPRPs are based on ordering the time series, Caballero-Pintado et al. [8] define a new OPRP using a symbolic correlation integral and by ordering the time series relative to the original.

Motivated by the study of Lu et al. [19], our goal here is to connect these two streams of studies and show that OPRP can visualise the effect of an exponential increase of forbidden ordinal patterns through the increase of embedding dimension m. To unveil the change of dynamics as the parameters of the dynamical system change, we introduce and validate two new measures, the mean and the median of the relative frequencies of order patterns that occur in a time series. To examine the evolution of those two measures, as deterministic test models, we use the logistic map, the delayed logistic map, the tent map and the Hénon map; as stochastic test models, we use Gaussian and Brown noise.

Methodology

Underlying a one-dimensional time series ($\{x_t\}_{t=1}^{N}$) with length N, order patterns of $\{x_t\}_{t=1}^{N}$ depend on the time delay τ and the embedding dimension m [6]. We encode x_t to its order pattern through ranked elements in the vector $\overrightarrow{x(t)} = (x_t, x_{t+\tau}, ..., x_{t+(m-1)\tau})$ [13]. Here, we study order patterns in the case of $\tau = 1$, and then rank $\overrightarrow{x(t)} = (x_t, x_{t+1}, ..., x_{t+(m-1)})$, where 0 corresponds to the smallest element in $\overrightarrow{x(t)}$. To guarantee the uniqueness of order patterns, when $x_{t+i} = x_{t+j}$, $i < j$, we assume $s_i < s_j$, where s_i is the rank of x_{t+i} in $\overrightarrow{x(t)}$. As a result, order patterns of $\{x_t\}_{t=1}^{N}$ are permutations of $\{0, 1, ..., m-1\}$, and the length of order patterns equals the embedding dimension m with the number of well defined order patterns in $\{x_t\}_{t=1}^{N}$ equalling $N - m + 1$. An order pattern of $\{x_1 = 4, x_2 = 1, x_3 = 7, x_4 = 5\}$, with embedding dimension $m = 4$ corresponds to the permutation (ordered sequence) $(1, 0, 3, 2)$ since 7 is the largest number corresponding to the symbol 3 and 1 being the smallest element corresponding to the symbol 0.

Order Patterns Recurrence Plot

Given a time series, an OPRP [13, 21] visualises its order patterns (i.e. ordered sequences) and analyses recurrent order structures using the following matrix:

$$R_{i,j}(m) = \begin{cases} 1, \pi_i = \pi_j \\ 0, \pi_i \neq \pi_j \end{cases} \quad i, j = 1, 2, ..., N - m + 1 \tag{1}$$

where π_i and π_j are the order patterns of the ith and jth element of $\{x_t\}_{t=1}^{N}$, respectively.

Forbidden Ordinal Patterns

Let $\Pi_m = \{\pi_{\xi_1}, ..., \pi_{\xi_n}\}$ be the set of order patterns that occur in $\{x_t\}_{t=1}^{N}$ with embedding dimension m and C_i be the frequency of $\pi_{\xi_i} \in \Pi_m$, then

$$\sum_{i=1}^{n} C_i = N - m + 1 \tag{2}$$

where n is the number of order patterns that appear in $\{x_t\}_{t=1}^{N}$.

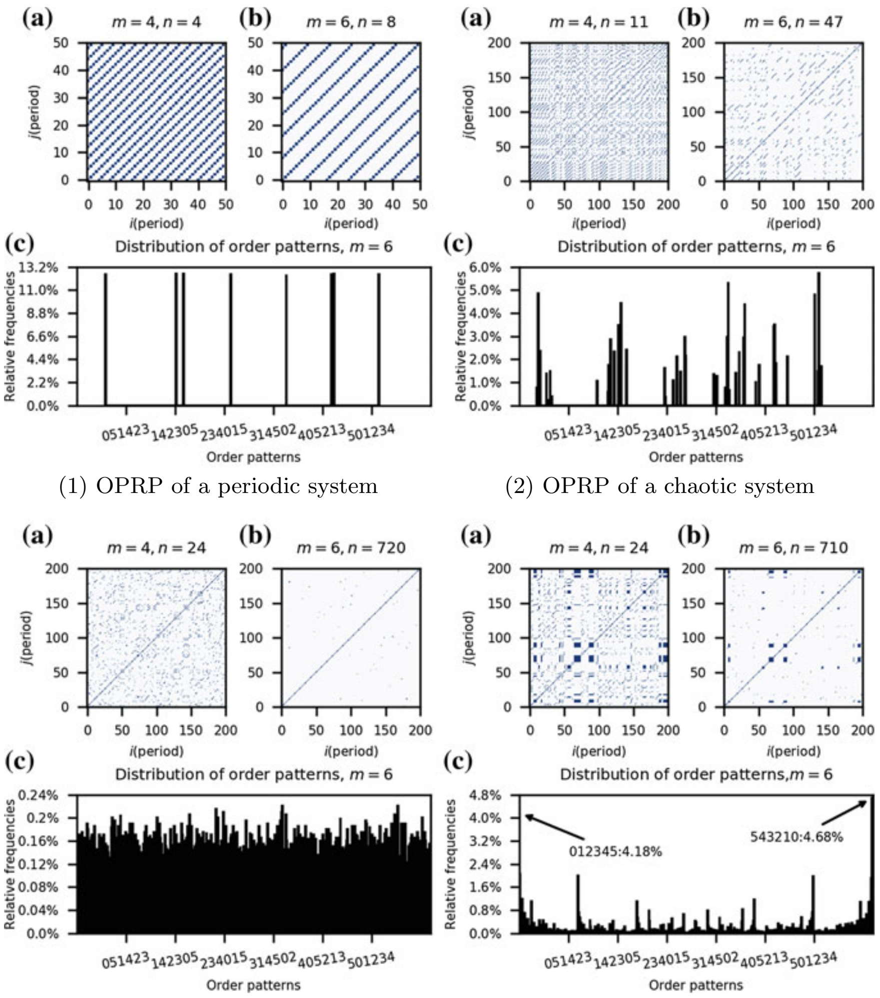
The maximal value of n is $m!$ when $N \geq m! + m - 1$. This leads to all possible order patterns showing up and the number of forbidden ordinal patterns becoming zero. If the time series is long enough to allow every possible order pattern ($N \gg m!$) to show up, then the number of forbidden ordinal patterns (FOP) becomes

$$\text{FOP} = m! - n \tag{3}$$

with $n < m!$. The relative frequency (RF) of π_{ξ_i} is then provided by

$$\text{RF}_{\pi_{\xi_i}} = \frac{C_i}{N - m + 1} \tag{4}$$

The mean of relative frequencies (MRF) is then given by



(1) OPRP of a periodic system

(2) OPRP of a chaotic system

(3) OPRP of Gaussian noise

(4) OPRP of Brown noise

Fig. 1 OPRPs as m increases. The embedding dimension m is 4 (**a**) and 6 (**b**). **c** is the distribution of order patterns. (1) and (2) logistic map $x_i = \alpha x_{i-1}(1 - x_{i-1})$ for $\alpha = 3.55$ (periodic) and 3.91 (chaotic), respectively; (3) Gaussian noise (20,000 samples chosen from $\mathcal{N}(0,\ 1)$); (4) Brown noise (20,000 samples chosen from power spectrum $S(f) = (1/f)^2$, with $f \sim \mathcal{N}(0,\ 1)$ [42])

$$\mathrm{MRF} = \sum_{i=1}^{n} \mathrm{RF}_{\pi_{\xi_i}}/n = \sum_{i=1}^{n} \frac{C_i}{n(N - m + 1)} = \frac{1}{n} \tag{5}$$

Kulp and Zunino [17] observe that a few order patterns of the fractional Brown noise have high frequencies. This leads to a skewed frequency distribution. The median is a better indicator to look at the statistic property of the frequency distribution. The reason is in that the median can reduce the influence of outliers [12]. The median of relative frequencies (MDRF) can be expressed via

$$\mathrm{MDRF} = \mathrm{median}\{\mathrm{RF}_{\pi_{\xi_1}}, ..., \mathrm{RF}_{\pi_{\xi_n}}\} = \frac{\mathrm{median}\{C_1, ..., C_n\}}{N - m + 1} \tag{6}$$

However, the MRF and the MDRF may variate in a small interval. This hinders visualising a small variation in the MRF and the MDRF. We therefore take the negative value of the logarithm of the mean (mean^l) and the median (median^l) of all relative frequencies. Taking the logarithm is consistent with the definition of permutation entropy of Bandt and Pompe [6]. Here, the mean^l and the median^l are expressed using

$$\mathrm{mean}^l = -\log \mathrm{MRF} = \log n \tag{7}$$

$$\mathrm{median}^l = -\log \mathrm{MDRF} = \log(N - m + 1) - \log(\mathrm{median}\{C_1, ..., C_n\}). \tag{8}$$

Plugging Eq. 3 into Eq. 7, we obtain a relationship between the mean^l and the FOP via

$$\mathrm{mean}^l = \log(m! - \mathrm{FOP}). \tag{9}$$

A pseudo algorithm to calculate the mean^l and the median^l is provided below.

Algorithm 1 Calculation of the mean^l and the mean^l

```
procedure TIME SERIES TO SYMBOLIC SEQUENCES THROUGH THE BP METHOD
    if $i < N - m + 1$, where $N = 20,000$, $m = 6$.
    Order patter $\pi_j = (j_0, \cdots, j_5) \leftarrow \{x_i, \cdots, x_{i+5}\}$
    Record the frequency: $\pi_j : C_j$, where $j \leq n$.
procedure RELATIVE FREQUENCY OF EACH $\pi_j$
    Calculate the MRF and MDRF of each $\pi_j$ according to Eqs. 5 and 6.
procedure CALCULATE THE $\mathrm{mean}^l$ AND THE $\mathrm{median}^l$
    Calculate the $\mathrm{mean}^l$ and the $\mathrm{mean}^l$ according to Eqs. 7 and 8.
```

Bifurcation diagrams are plotted using the following procedures:

Algorithm 2 Make the bifurcation diagram

```
procedure CHANGE THE BIFURCATION PARAMETER:
    for $r_i = r_{\min} + i(r_{\max} - r_{\min})/1{,}000$, where $i \leq 1{,}000$ do
        Record the steady-state solutions to form a time series (x_list)
        Implement Algorithm 1 through x_list
```

where $r_{\min}$ and $r_{\max}$ are the minimal and maximal value of bifurcation parameters to plot the bifurcation diagram.

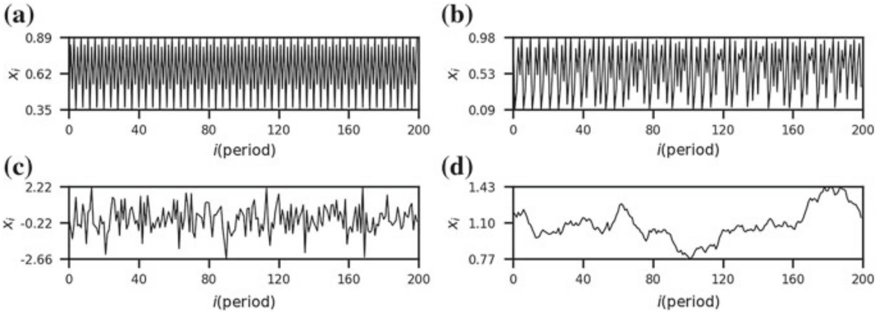


Fig. 2 Time series. (1) and (2) logistic map (periodic, $r = 3.55$), and (chaotic, $r = 3.91$); (3) Gaussian noise of Fig. 1(3); (4) Brown noise of Fig. 1(4); parameters of noise as provided previously

Results

Order Patterns Recurrence Plot

For Gaussian noise (Fig. 1(3)), the increase of m makes the OPRP sparse. In Fig. 1(3a, b), we use Gaussian noise and see that the OPRP is denser for $m = 4$ than for $m = 6$. However, the OPRP of Brown noise (Fig. 1(4)) preserves many points indicating that the system has the order pattern of which the relative frequency is high. For the periodic or the chaotic system under study (Fig. 1(1, 2)), the OPRPs show distinctly more diagonal lines than the cases of Gaussian or Brown noise. As highlighted by Rosso et al. [36], a minimal embedding dimension allows the occurrence of forbidden ordinal patterns and their exponential growth. An OPRP can then have an optimal embedding dimension that rules out Gaussian noise. Figure 2 shows the time series of the corresponding signals in Fig. 1.

The reasons for a sparse OPRP in case of Gaussian noise are that no forbidden ordinal patterns exist [4] and that the relative frequencies of every order pattern are close [1]. Therefore as m increases (Fig. 1(3c)), the number of order patterns that occurs is up to $m!$, and thus the probability of an order pattern that recurs in a given time window is $\frac{1}{m!}$, which is low for large m, leading to a spare distribution of points in a given time window.

However, for Brown noise, in a given time window, the time series shows an upward (downward) tendency (Fig. 2d). As the upward (downward) tendency lasts for six time intervals, it forms the order pattern 012345 (543210). If the tendency lasts for over six time intervals, the order pattern does not change and forms a cluster in the OPRP (Fig. 1(4c)).

Due to the tendencies in the time series, Brown noise shows high relative frequencies (over 4%) of order patterns (012345 and 543210) (Fig. 1(4c)), compared with Gaussian noise (below 0.24%) (Fig. 1(3b)). As consistent with Rosso et al. [35], Brown noise shows missing ordinal patterns ($n = 710 < 6! = 720$) (Fig. 1(4b, c)).

Bifurcation Diagrams

Caballero-Pintado et al. [8] showed that RQA in the OPRP can detect changing dynamics. Now, we show how the mean^l and median^l unveil a change in dynamics and are able to uncover details in bifurcation diagrams. The embedding dimension is $m = 6$ in the following.

Figure 3 shows that the change of the mean^l is like a staircase as bifurcation parameters change. That is, the number of forbidden ordinal patterns is robust to the change in dynamics. This result is complementary to the observation that forbidden ordinal patterns are robust to irregular sampling time [16].

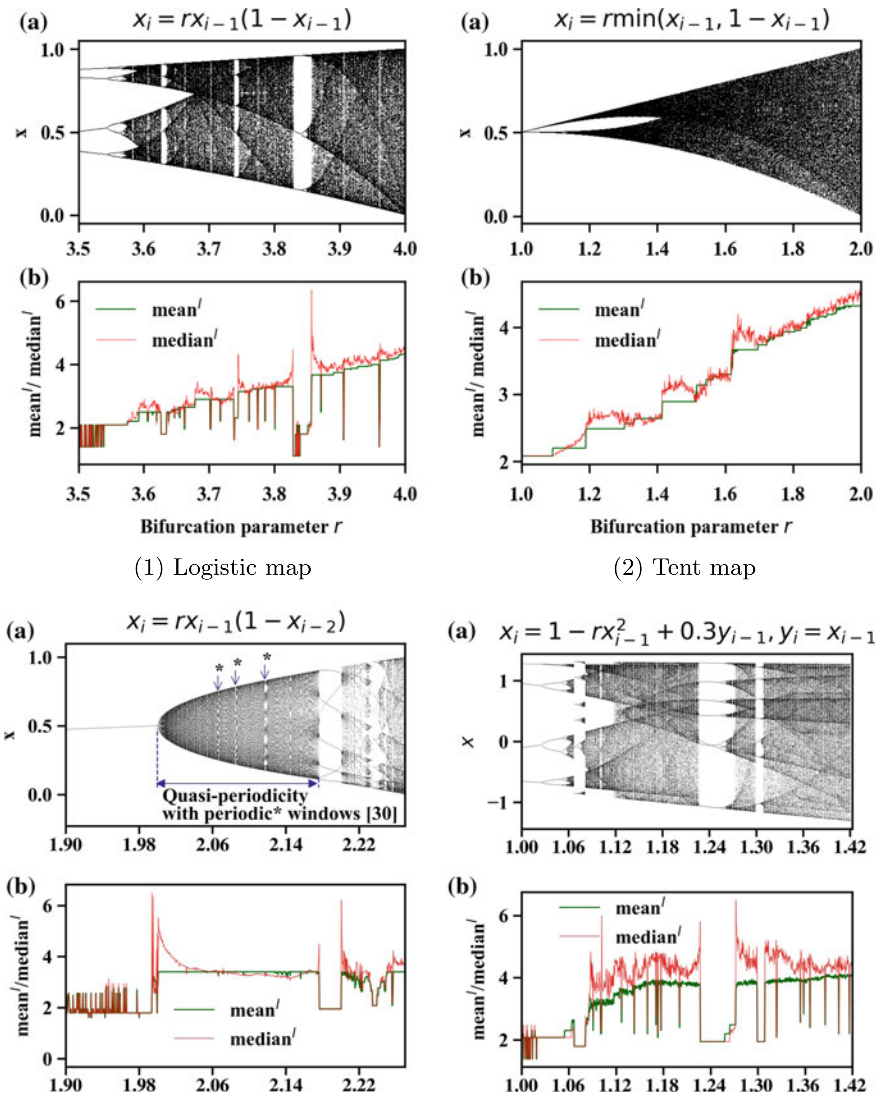
From Fig. 3(1b, 2b, 3b and 4b), we find that the median^l quickly responds to a small change in the bifurcation parameter. As a result, the median^l is non-robust to a change in dynamics. In a periodic window, the median^l is close or equal to the mean^l. However, in a chaotic window, the median^l and the mean^l become more visible. Also, the median^l abruptly decreases to fit with the mean^l in a periodic point or a periodic window. In contrast, a sudden increase of the mean^l is found at points where the system switches between chaotic and periodic dynamics.

In a periodic window, the length of a time series may cause the number of some order patterns to become slightly higher than the order patterns of the neighbouring regimes.[1] For the same length of the time series, as the periodicity increases, this difference becomes smaller. However, the difference may be visualised in a low periodic window such as the period-3 window. This is possible since a low periodicity leads to a high relative frequency for each order pattern. The unit difference of low periodicity induces a higher difference between the median^l and the median^l than that of a high periodicity (Fig. 3(1b)).

In a periodic window, the order patterns are evenly distributed without consideration of the unit difference, whereas in a chaotic window, the order patterns are not necessarily evenly distributed [17]. An uneven distribution of order patterns can result in a difference of median^l and mean^l. Figure 3(1, 2) shows that the mean^l changes step by step even in a chaotic region, contributing to the number of order patterns which is robust against the change in dynamics. According to Eq. 3, when n is robust, FOP is robust as well. Therefore, our results support that the number of forbidden ordinal patterns show some degree of robustness to the changes of dynamics since the mean^l is related to the number of forbidden ordinal patterns.

As shown by Sprott [39], for the delayed logistic map, it is difficult to distinguish chaos from quasiperiodicity using a bifurcation diagram. However, Fig. 3(3b) shows different structures of the change of the median^l and mean^l under the two kinds of dynamics. In the window of almost quasi-periodicity (see Fig. 3(3a)), where the

[1] For example, in our numerical experiments, $N = 20{,}000$, $m = 6$, for the logistic map with $r = 3.55$, eight order patterns appear, so that the total amount of these eight order patterns is $N - m + 1 = 19{,}995$. However, 19,995 cannot be divided by 8, so the count of some order patterns is 2,500, that of the others is 2,499, resulting in the difference of unity between the mean and the median of $\{C_1, \ldots, C_n\}$. This further shows the difference between the mean^l and the median^l according to Eqs. 7 and 8.



(1) Logistic map

(2) Tent map

(3) Delayed logistic map

(4) x component of Hénon map

Fig. 3 Bifurcation diagrams (**a**) and the evolution of the mean^l and the median^l (**b**). (1)–(4) correspond to logistic map, tent map, delayed logistic map and Hénon map, respectively

maximal Lyapunov exponent remains at zero [24, 39], the mean^l shows no evident variations and the median^l has no abrupt changes. Also, different from periodic windows, the evolution of the median^l does not fit with that of the mean^l, showing a different quality in changes.

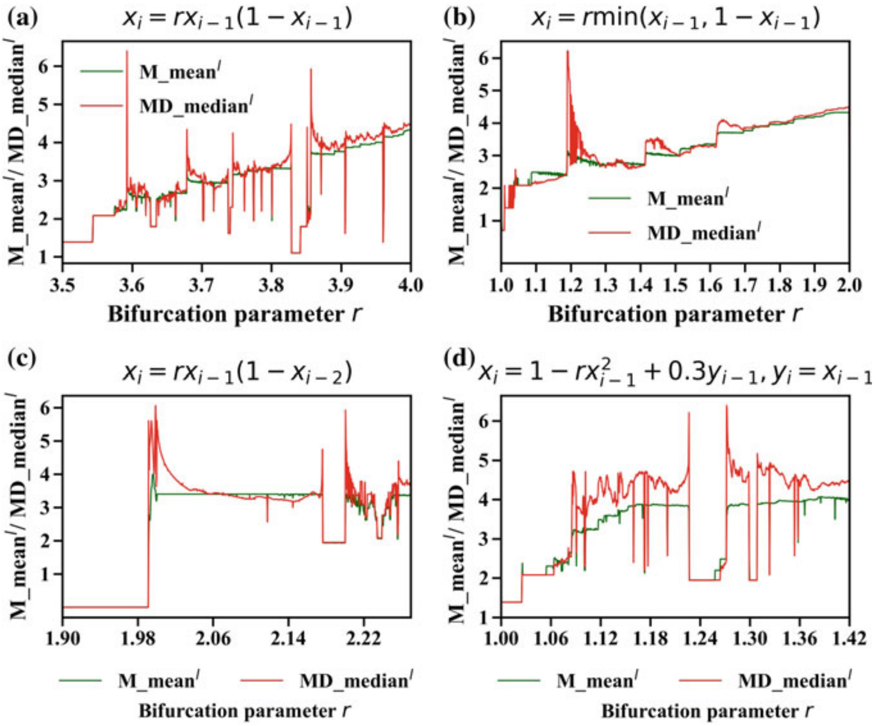


Fig. 4 Analysis of the mean^l and the median^l as more initial conditions are considered. The M_mean^l represents the mean of the mean^l; the MD_median^l is the median of the median^l. The bifurcation diagram is plotted on a 1,000 × 1,000 grid with each point in the x-axis having 1,000 samples of randomised conditions. **a** Logistic map; **b** tent map; **c** delayed logistic map; **d** Hénon map

Next, we study the influence of randomised initial conditions on the mean^l and the median^l as bifurcation parameters change. We implement the Algorithm 2 using 1,000 randomised initial conditions for each r_i. For logistic map and tent map, initial conditions are chosen from $\mathrm{U}(0, 1)$; for the delayed logistic map, both the x_0 and x_1 are chosen from $\mathrm{U}(0.4, 0.8)$; for the Hénon map, both the x_0 and y_0 are chosen from $\mathrm{U}(0, 1)$. The mean^l and median^l are calculated in each sample. Then the y-axis records the mean of the mean^l or the median of the median^l. Figure 4 shows the results. Figures 4c and 3(3b) have the same phenomena in the almost quasiperiodic window: the mean^l is almost fixed.

The influence of noise on the two measures studying a noise-contaminated regime of the logistic map is examined in Fig. 5. Variations in the mean^l is less than those of the median^l. This result is consistent with the observation that the FOP is robust against Gaussian noise [1]. Since the mean^l $= \log(m! - FOP)$, a robust FOP leads to a more robust mean^l against Gaussian noise. However, the median^l shows persistent variations, which reflects that the system is changing. Yet, whether this robustness

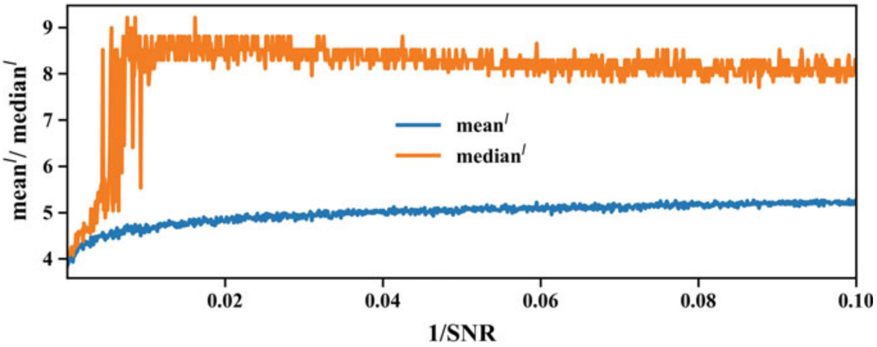


Fig. 5 Influence of noise on the mean^l and the median^l of the logistic map as $r = 3.91$. SNR represents the signal-to-noise ratio. Here $\text{SNR} = \delta^2_{\text{signal}}/\delta^2_{\text{noise}}$, where $\delta^2_{\text{signal}} = 0.0926$. The noise is additive and follows $\mathcal{N}(0, \delta^2_{\text{noise}})$

will lead to similar prediction quality over a wide range of bifurcation parameters as shown for purely deterministic regimes and different or even different and varying SNRs will need to be studied in the future in more detail.

Conclusion

We have used the mean (mean^l) and the median (median^l) of the relative frequency of order patterns to characterise chaotic signals and stochastic signals. Although the mean and the median are common measurements in statistics, our definitions provide new measurements for order pattern recurrence plots to distinguish between chaotic signals and stochastic signals and to detect the change in dynamics in purely chaotic signals. Here, the mean^l quantifies the number of forbidden order patterns (FOP, see Eq. 9) for the first time.

Also, monitoring the embedding dimension can be used to qualify chaotic versus stochastic dynamics in an OPRP. An optimally embedded OPRP preserves diagonal lines for a chaotic system and leads in the case of uncorrelated noise to thinly distributed isolated points and in the case of correlated noise to small clusters. However, whether this holds true for any system especially if noise-contaminated systems are considered needs to be verified in the future.

While the mean^l is robust to changing bifurcation parameters, the median^l is sensitive to the change in the dynamics. Using the difference between the mean^l and the median^l and their fluctuations, we can distinguish chaos from quasiperiodic dynamics for the delayed logistic map.

The calculation of the median^l and the mean^l is easily accomplished and computationally inexpensive, which makes the median^l and the mean^l a cost-effective analysis tool to tell deterministic signals from stochastic signals.

Further, the mean^l shows the robustness to Gaussian noise. However, the median^l is not robust to Gaussian noise. Since the relative frequencies of each order pattern of Gaussian noise are close in value to each other, the mean^l and the median^l of deterministic signals being buried in Gaussian noise, are likely to be different from that of pure noise.

From a practical viewpoint, especially low-dimensional deterministic signals are often contaminated by high-dimensional processes (noise); yet only the lower dimensional scales are often of interest. Cleaning these signals by employing dynamics-preserving non-linear (geometric) filters as conducted by Oberst et al. [25, 29] could expand utilising the measures, which are suggested here to real-life data problems. Especially studying border-collision/grazing bifurcations the measures could potentially assist in detecting of tipping points in bifurcation diagrams of maps and flows related to discontinuous dynamics, cf. [14, 23].

References

1. Amigó, J., Zambrano, S., Sanjuán, M.A.: Combinatorial detection of determinism in noisy time series. EPL (Eur. Lett.) **83**(6), 60005 (2008)
2. Amigó, J.: Permutation Complexity in Dynamical Systems. Springer, Berlin (2010)
3. Amigó, J.M., Kocarev, L., Szczepanski, J.: Order patterns and chaos. Phys. Lett. A **355**(1), 27–31 (2006)
4. Amigó, J.M., Zambrano, S., Sanjuán, M.A.: True and false forbidden patterns in deterministic and random dynamics. EPL (Eur. Lett.) **79**(5), 50001 (2007)
5. Amigó, J.M., Zambrano, S., Sanjuán, M.A.: Detecting determinism in time series with ordinal patterns: a comparative study. Int. J. Bifurc. Chaos **20**(09), 2915–2924 (2010)
6. Bandt, C., Pompe, B.: Permutation entropy: a natural complexity measure for time series. Phys. Rev. Lett. **88**(17), 174102 (2002)
7. Barreiro, M., Marti, A.C., Masoller, C.: Inferring long memory processes in the climate network via ordinal pattern analysis. Chaos **21**(1), 013101 (2011)
8. Caballero-Pintado, M.V., Matilla-García, M., Ruiz Marín, M.: Symbolic recurrence plots to analyze dynamical systems. Chaos **28**(6), 063112 (2018)
9. Carpi, L.C., Saco, P.M., Rosso, O.: Missing ordinal patterns in correlated noises. Phys. A Stat. Mech. Appl. **389**(10), 2020–2029 (2010)
10. Chen, B., Huang, J., Ji, J.: Control of flexible single-link manipulators having duffing oscillator dynamics. Mech. Syst. Signal Process. **121**, 44–57 (2019)
11. Donner, R., Hinrichs, U., Scholz-Reiter, B.: Symbolic recurrence plots: a new quantitative framework for performance analysis of manufacturing networks. Eur. Phys. J. Spec. Top. **164**(1), 85–104 (2008)
12. Gottwald, G.A., Melbourne, I.: Testing for chaos in deterministic systems with noise. Phys. D Nonlinear Phenom. **212**(1), 100–110 (2005)
13. Groth, A.: Visualization of coupling in time series by order recurrence plots. Phys. Rev. E **72**(4), 046220 (2005)
14. Hu, Z., Chen, X., Hu, P.: Dynamic pricing with gain-seeking reference price effects. Oper. Res. **64**(1), 150–157 (2016)
15. Kulp, C.W., Smith, S.: Characterization of noisy symbolic time series. Phys. Rev. E **83**(2), 026201 (2011)
16. Kulp, C., Chobot, J., Niskala, B., Needhammer, C.: Using forbidden ordinal patterns to detect determinism in irregularly sampled time series. Chaos **26**(2), 023107 (2016)

17. Kulp, C., Zunino, L.: Discriminating chaotic and stochastic dynamics through the permutation spectrum test. Chaos Interdiscip. J. Nonlinear Sci. **24**(3), 033116 (2014)
18. La Torre, D., Marsiglio, S., Privileggi, F.: Fractal attractors in economic growth models with random pollution externalities. Chaos **28**(5), 055916 (2018)
19. Lu, S., Luo, Z., Zhang, G., Oberst, S.: Order pattern recurrence plots: unveiling determinism buried in noise. In: FEIT Research Showcase. University of Technology Sydney, Sydney, NSW, Australia, 14 June 2018
20. Marwan, N., Groth, A., Kurths, J.: Quantification of order patterns recurrence plots of event related potentials. Chaos Complex. Lett. **2**, 301–314 (2007a)
21. Marwan, N., Romano, M.C., Thiel, M., Kurths, J.: Recurrence plots for the analysis of complex systems. Phys. Rep. **438**(5–6), 237–329 (2007b)
22. McCullough, M., Sakellariou, K., Stemler, T., Small, M.: Regenerating time series from ordinal networks. Chaos **27**(3), 035814 (2017)
23. Nazarimehr, F., Jafari, S., Hashemi Golpayegani, S.M.R., Perc, M., Sprott, J.C.: Predicting tipping points of dynamical systems during a period-doubling route to chaos. Chaos **28**(7), 073102 (2018)
24. Oberst, S., Lai, J.: A statistical approach to estimate the lyapunov spectrum in disc brake squeal. J. Sound Vib. **334**, 120–135 (2015)
25. Oberst, S., Niven, R., Lester, D., Ord, A., Hobbs, B., Hoffmann, N.: Detection of unstable periodic orbits in mineralising geological systems. Chaos **28**(8), 085711 (2018)
26. Oberst, S.: Nonlinear dynamics: towards a paradigm change via evidence-based complex dynamics modelling. In: NOVEM 2018, Ibiza, Spain, 7–9 May 2018
27. Oberst, S., Bann, G., Lai, J.C., Evans, T.A.: Cryptic termites avoid predatory ants by eavesdropping on vibrational cues from their footsteps. Ecol. Lett. **20**(2), 212–221 (2017a)
28. Oberst, S., Lai, J.: Chaos in brake squeal noise. J. Sound Vib. **330**(5), 955–975 (2011)
29. Oberst, S., Marburg, S., Hoffmann, N.: Determining periodic orbits via nonlinear filtering and recurrence spectra in the presence of noise. Procedia Eng. **199**, 772–777 (2017b)
30. Olivares, F., Plastino, A., Rosso, O.A.: Contrasting chaos with noise via local versus global information quantifiers. Phys. Lett. A **376**(19), 1577–1583 (2012)
31. Panchuk, A., Sushko, I., Westerhoff, F.: A financial market model with two discontinuities: bifurcation structures in the chaotic domain. Chaos Interdiscip. J. Nonlinear Sci. **28**(5), 055908 (2018)
32. Parlitz, U., Berg, S., Luther, S., Schirdewan, A., Kurths, J., Wessel, N.: Classifying cardiac biosignals using ordinal pattern statistics and symbolic dynamics. Comput. Biol. Med. **42**(3), 319–327 (2012)
33. Porfiri, M., Marín, M.R.: Symbolic dynamics of animal interaction. J. Theor. Biol. **435**, 145–156 (2017)
34. Rosso, O.A., Larrondo, H.A., Martin, M.T., Plastino, A., Fuentes, M.A.: Distinguishing noise from chaos. Phys. Rev. Lett. **99**(15), 154102 (2007)
35. Rosso, O.A., Carpi, L.C., Saco, P.M., Ravetti, M.G., Plastino, A., Larrondo, H.A.: Causality and the entropy complexity plane: robustness and missing ordinal patterns. Phys. A Stat. Mech. Appl. **391**(1), 42–55 (2012)
36. Rosso, O.A., Olivares, F., Zunino, L., De Micco, L., Aquino, A.L., Plastino, A., Larrondo, H.A.: Characterization of chaotic maps using the permutation bandt-pompe probability distribution. Eur. Phys. J. B **86**(4), 116 (2013)
37. Schindler, K., Gast, H., Stieglitz, L., Stibal, A., Hauf, M., Wiest, R., Mariani, L., Rummel, C.: Forbidden ordinal patterns of periictal intracranial EEG indicate deterministic dynamics in human epileptic seizures. Epilepsia **52**(10), 1771–1780 (2011)
38. Schinkel, S., Marwan, N., Kurths, J.: Order patterns recurrence plots in the analysis of ERP data. Cogn. Neurodynamics **1**(4), 317–325 (2007)
39. Sprott, J.C.: Chaos and Time-series Analysis, vol. 69. Oxford University Press, Oxford (2003)
40. Stender, M., Tiedemann, M., Hoffmann, N., Oberst, S.: Impact of an irregular friction formulation on dynamics of a minimal model for brake squeal. Mech. Syst. Signal Process. **107**, 439–451 (2018)

41. Strogatz, S.H.: Nonlinear Dynamics and Chaos: with Applications to Physics, Biology, Chemistry, and Engineering. CRC Press (2018)
42. Timmer, J., Koenig, M.: On generating power law noise. Astron. Astrophys. **300**, 707 (1995)
43. Wernitz, B., Hoffmann, N.: Recurrence analysis and phase space reconstruction of irregular vibration in friction brakes: signatures of chaos in steady sliding. J. Sound Vib. **331**(16), 3887–3896 (2012)
44. West, B.J.: Fractal Physiology and Chaos in Medicine, vol. 16. World Scientific (2012)
45. Zanin, M.: Forbidden patterns in financial time series. Chaos **18**(1), 013119 (2008)
46. Zanin, M., Zunino, L., Rosso, O.A., Papo, D.: Permutation entropy and its main biomedical and econophysics applications: a review. Entropy **14**(8), 1553–1577 (2012)
47. Zhang, J., Zhou, J., Tang, M., Guo, H., Small, M., Zou, Y.: Constructing ordinal partition transition networks from multivariate time series. Sci. Rep. **7**(1), 7795 (2017)
48. Zunino, L., Zanin, M., Tabak, B.M., Pérez, D.G., Rosso, O.A.: Forbidden patterns, permutation entropy and stock market inefficiency. Phys. A Stat. Mech. Appl. **388**(14), 2854–2864 (2009)

Time Series Modeling with MATLAB: The SSpace Toolbox

Diego J. Pedregal, Marco A. Villegas, Diego A. Villegas and Juan R. Trapero

Abstract SSpace is a MATLAB toolbox for State-Space modeling that provides the user with tools for linear Gaussian, nonlinear, and non-Gaussian systems with the most advanced and up-to-date features available in any State-Space framework. Great flexibility is achieved because each model is coded on a standard MATLAB function, thence having absolute control on particular parameterizations, parameter constraints, time variation of parameters or variances, arbitrary nonlinear relations with inputs, time aggregation, nested models, system concatenation, etc. The toolbox may be used by specifying State-Space systems from scratch or by using ready-to-use templates for standard methods (like VARMAX, exponential smoothing, unobserved components, Dynamic Linear Regression, etc.). The toolbox is freely available via a public code repository with full documentation and help system. This chapter demonstrates the toolbox's potential with several examples.

Keywords MATLAB · State-Space systems · Kalman filter · Smoother algorithm · Maximum likelihood

Introduction

SSpace is a MATLAB toolbox that implements linear, nonlinear and non-Gaussian State-Space (SS) systems in a very flexible and powerful way. It is mainly based on the work of Young and collaborators [14, 15] along many years seasoned with

This work was supported by the European Regional Development Fund and Spanish Government (MINECO/FEDER, UE) under the project with reference DPI2015-64133-R and by the Vicerrectorado de Investigación y Política Científica from UCLM by DOCM 31/07/2014 [2014/10340].

D. J. Pedregal (✉) · M. A. Villegas · D. A. Villegas
Industrial Engineering Politecnic, University of Castilla-La Mancha, 13071 Ciudad Real, Spain
e-mail: diego.pedregal@uclm.es

J. R. Trapero
Faculty of Chemical Sciences and Technologies, University of Castilla-La Mancha, 13071 Ciudad Real, Spain
e-mail: juanramon.trapero@uclm.es

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*, Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_6

many other elements, mainly found in the books of Harvey, Durbin, and Koopman [5, 7]. Though SS systems may be considered a “classical” tool nowadays (especially in engineering and economics), the approach is still remarkably alive as an area of active research, judging by the number of research articles and books on this topic.

There are also numerous packages available in the marketplace: some of them available for free, and many others available commercially. There are already several toolboxes written in MATLAB including some supplied with the core program (like Signal Processing, Control, etc.), but others also exist, such as CAPTAIN [14], SSM [12], SSMMATLAB [6], and E4 [2]. Two packages worth mentioning because of their relevance are STAMP [9] and SSfPack [10]. Some further examples are listed in volume 41, 2011 of the Journal of Statistical Software [3]. Some others are written either in R [13], RATS [4], gretl [11], etc. Among the commercial programs, the following incorporate SS routines with different degrees of complexity: Eviews, SAS, Stata, etc.

In a broad sense, SSpace provides the user with the most advanced and up-to-date features available in any State-Space framework, e.g., the capability of dealing with both univariate and multivariate models, exact Kalman filter initialization, univariate treatment of multivariate time series, nonlinear and non-Gaussian modeling, alternative objective functions in parameter optimization (not only maximum likelihood), straightforward modeling of nonlinear input–output relationships, etc.

The flexibility and easiness of use are reflected in the fact that SSpace was designed keeping in mind the final user and the usability of the library, by selecting easy-to-remember function names, and more importantly, by allowing a direct correspondence between the analytical expression of models and the corresponding definition in MATLAB code. In addition, users are also provided with a set of model templates for approaching many standard models with maximum simplicity. A full help system and documentation for each function are included in both HTML and MATLAB format, complemented with eight step-by-step demos to demonstrate the use of the toolbox with standard well-known examples and others much less standard.

A final advantage of SSpace is that it is freely available via the Internet at https://bitbucket.org/predilab/sspace-matlab/, where potential users are encouraged to push their own contributions and suggestions.

General State-Space Framework

SSpace supports multivariate linear and nonlinear Gaussian models and univariate non-Gaussian models. The linear Gaussian version is shown in Eq. (1).

$$\begin{aligned} \text{State Equations: } \alpha_{t+1} &= T_t \alpha_t + \Gamma_t + R_t \eta_t \\ \text{Observation Equations: } y_t &= Z_t \alpha_t + D_t + C_t \epsilon_t \end{aligned} \tag{1}$$

In these equations, α_t is the state vector of length n; y_t are the $m \times 1$ vectors of output data; $\eta_t \sim N(0, Q_t)$ and $\epsilon_t \sim N(0, H_t)$ are the state and observational vectors of Gaussian noises, with dimensions $r \times 1$ and $h \times 1$, respectively; both noises are allowed to be correlated by a system matrix $S_t = Cov(\eta_t, \epsilon_t)$ of dimension $r \times h$; Γ_t and D_t are two matrices included to deal with input–output models in a flexible way. The remaining elements in (1) are the rest of the system matrices with appropriate dimensions. The system is completed by making assumptions about the stochastic properties of the initial state vector, i.e., $\alpha_1 \sim N(a_1, P_1)$, where a_1 and P_1 are its mean and covariance matrix, respectively.

The nonlinear models in SSpace are shown in Eq. (2).

$$\begin{aligned} \alpha_{t+1} &= T_t(\alpha_t) + \Gamma_t + R_t(\alpha_t)\eta_t \\ y_t &= Z_t(\alpha_t) + D_t + C_t(\alpha_t)\epsilon_t \end{aligned} \tag{2}$$

Functions $T_t(\alpha_t)$ and $Z_t(\alpha_t)$ provide nonlinear transformations of the state vector into vectors of size $n \times 1$ and $m \times 1$, respectively. Matrices Q_t and H_t may also depend on the state vector, but $S_t = 0$.

Finally, the non-Gaussian SS setup is shown in Eq. (3):

$$\begin{aligned} \alpha_{t+1} &= T_t\alpha_t + \Gamma_t + R_t\eta_t \\ y_t &\sim p(y_t \mid \theta_t) + D_t \\ \theta_t &= Z_t\alpha_t \end{aligned} \tag{3}$$

Here θ_t is known as the *signal*. This representation allows stochastic volatility models (i.e., $y_t = exp(\frac{1}{2}\theta_t)\epsilon_t + D_t$); exponential family models (where $p(y_t \mid \theta_t) = exp[y_t'\theta_t - b_t(\theta_t) + c_t(y_t)]$, $-\infty < \theta_t < \infty$); and models in which the observations are generated by the relation $y_t = \theta_t + \epsilon_t$, $\epsilon_t \sim p(\epsilon_t)$ (with $p(\bullet)$ being a distribution of the exponential family).

Given any of the previous systems, the estimation problem consists of finding the first- and second-order moments (i.e., mean and covariance) of the state vector, conditional on all the data in a sample. The tools that allow this operation to be performed in linear Gaussian systems are the well-known Kalman filter, fixed interval, and disturbance smoothers. These algorithms may be adapted to deal with nonlinear systems by running them on a Taylor linear expansion of the original nonlinear system (extended Kalman filter and smoothers). Things become rather more complicated for non-Gaussian systems, which require simulation-based methods that imply running the recursive algorithms repeatedly with an extra computational burden. An excellent exposition of all these filtering and smoothing techniques may be found in [5], see also SSpace documentation.

The application of the recursive algorithms requires knowledge of all the system matrices. The normal situation is that part of the system matrices are known a priori and part are unknown. The unknowns are estimated in SSpace by time domain Exact Maximum Likelihood (ML) optimization, though less common procedures are also available in SSpace, for example, estimation by minimization of several-step-ahead forecast errors.

There are plenty of issues not commented in this chapter because of space constraints. For further reading, refer to SSpace documentation and [5, 7, 14, 15].

SSpace Overview

The feature that gives SSpace its real power is the possibility of specifying models in MATLAB coded functions. Such functions follow a fixed structure that is supplied with the toolbox in a set of templates that should be used to avoid coding bugs. The general template is a standard MATLAB function called `SampleSS` shown below. This function has an input argument `p`, that is a vector of unknown parameters and that will be estimated later on. The system matrix names are easily identifiable and the template will work properly as long as the user does not remove anything from it, and just adds meaningful MATLAB code.

```
function model = SampleSS(p)
  model.T = []; model.Gam = []; model.R = [];
  model.Z = []; model.D   = []; model.C = [];
  model.Q = []; model.H   = []; model.S = [];
```

Take as an example the AR(1) process in Eq. (4) with $var(\eta_t) = \sigma_\eta^2$.

$$y_t = \phi y_{t-1} + \eta_t \tag{4}$$

A straightforward SS representation of this model consists of Eq. (4) playing the role of the state equation with $y_t = \alpha_t$ as the observation equation. By comparing this particular case with the general linear Gaussian case in Eq. (1), the system SS matrices are inferred as $T_t = \phi$, $R_t = Z_t = 1$, $C_t = 0$, $Q_t = \sigma_\eta^2$, $H_t = 0$. Γ_t, and D_t do not exist because the model has no inputs. The filled-in version of `SampleSS` for this particular case is listed below, where the function is renamed as `ar1` to keep the original version of `SampleSS` intact for future use.

```
function model = ar1(p)
  model.T = p(1);      model.Gam = []; model.R = 1;
  model.Z = 1;         model.D   = []; model.C = 0;
  model.Q = 10.^p(2);  model.H   = 0;  model.S = [];
```

The input `p` is in this case a vector of two elements, namely ϕ and σ_η^2. Beware that the system matrix $Q = \sigma_\eta^2$, identified as the second element of the input argument `p`, should be positive or zero. Thence, Q is defined as any positive or negative power of 10.

Once the model is fully specified, the way it is handled should be told to SSpace by means of a number of fundamental functions shown in the following listing.

```
>> sys = SSmodel('y', data, 'model', @ar1);
>> sys = SSestim(sys);      % Estimation
>> sys = SSvalidate(sys);   % Validation
>> sys = SSsmooth(sys);     % Smoothing
```

The first command builds a new SSpace object, called `sys`, consisting of a model written in `ar1.m` that will be applied to the data stored in memory in a variable called `data`. The second command estimates the model by exact ML. The third shows the

Table 1 Main functions and templates included in SSpace

Main Functions	
`SSmodel`	Creates SSpace model object or adds properties to an existing one
`SSestim`	Estimation of an SSpace model
`SSvalidate`	Validation of an SSpace model
`SSfilter`	Optimal Kalman filtering of SSpace model
`SSsmooth`	Optimal fixed-interval smoothing of SSpace model
`SSdisturb`	Optimal disturbance smoother
`SSdemo`	Run SSpace demos 1–8
Templates	
Linear and Gaussian Models	
`SampleSS`:	General SS template
`SampleARIMA`:	ARIMA models with exogenous variables
`SampleVARMAX`:	VARMAX models
`SampleBSM`:	Basic structural model
`SampleDHR`:	Dynamic harmonic regression
`SampleDLR`:	Dynamic linear regression
`SampleES`:	Exponential smoothing with exogenous variables
Non-Gaussian Models	
`SampleNONGAUSS`:	General non-Gaussian models
`SampleEXP`:	Non-Gaussian exponential family models
`SampleSV`:	Sochastic volatility models
Nonlinear Models	
`SampleNL`:	General nonlinear models
Other Templates	
`SampleAGG`:	Models with time aggregation
`SampleCAT`:	Concatenation of State-Space systems
`SampleNEST`:	Nesting in inputs State-Space systems

results in tabular form with diagnostic statistics to check model validity. Finally, the last command provides the smoothed estimates of states and their covariance matrices. With each command, the system object `sys` is filled in with the relevant output information that may be used later on.

One caveat is that any model implemented has to be transformed into SS form as a previous step before it may be used in SSpace. However, this limitation is readily overcome, because the toolbox is provided with a number of predefined templates for a set of common methods. Table 1 lists all the available functions to deal with an existing SSpace system and the templates included, see details in the documentation.

For example, the same AR(1) model may be implemented with the aid of the `SampleARIMA` template. The advantage of using the `SampleARIMA` template is that the model is directly defined in terms of the ARIMA specification, instead of using its SS representation. Therefore, the user does not even need to know the SS representation of an AR model. The listing below shows the SampleARIMA template prepared to deal with a much more complicated ARIMA model, in which all the references to the system matrices of an SS model are replaced by alternative references to the backshift operator polynomials typical of an ARIMA model (as algebraic vectors), as is the norm in other MATLAB toolboxes.

```
function model= SampleARIMA(p)
Sigma = 10.^p(1);       % Noise variance
DIFFpoly= [1 -1]';      % Differences
ARpoly = conv([1 p(2)], [1 zeros(1, 11) p(3)])';
MApoly = [1 -p(2)]';    % AR and MA polynomials
D = p(4);               % Input variables (constant)
```

The model implemented is an ARIMA$(1, 1, 1) \times (1, 0, 0)_{12}$ with a constant (in matrix D) and a parameter constraint consisting of setting the AR(1) parameter as the negative of the MA(1). This is a constraint that will be ludicrous in many real situations but is introduced here solely as an example of how easy it is to implement parameter constraints in SSpace.

One last point worth mentioning is that Table 1 includes a template list for nonlinear and non-Gaussian templates. It also includes other templates to carry out useful operations with time series, namely time aggregation (SampleAGG), concatenation of SS systems (SampleCAT), and nesting SS systems in inputs (SampleNEST).

Examples

Example 1: Regression

Regression may be introduced in SSpace models in many different ways, and this worked example is included here as an illustration of SSpace flexibility when implementing this sort of models (see also demo number 5 of SSpace).

Consider 300 samples from a simulation of the model in Eq. (5), where B stands for the backshift operator such that $B^l y_t = y_{t-l}$; a_t is a Gaussian white noise serially independent with mean 0 and variance 0.25 and e_t is another Gaussian white noise with zero mean and variance 1. This case may be seen as a regression with three inputs, namely a constant, an AR(1) process, and a cosine wave. A simulated response of Eq. (5) is depicted in Fig. 1.

$$\begin{array}{c} y_t = 15 + 4u_t + 2\cos(2\pi t/50) + e_t \\ u_t = \frac{1}{(1-0.8B)} a_t \end{array} \tag{5}$$

Because of the simplicity of this model, one sees immediately that this regression with three inputs may be viewed as a simplified version of an SS system in which the state equation does not exist and the observation equation does not relate to the states

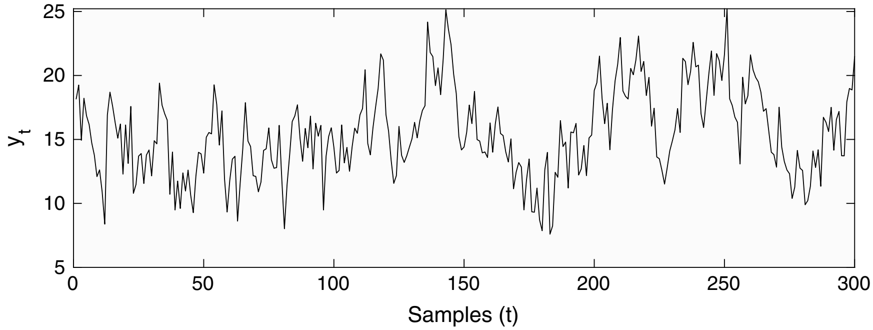


Fig. 1 A simulation sample of a model in Eq. (5)

in any way, i.e., $T_t = 0$, $R_t = Z_t = 0$, $C_t = 1$, $Q_t = 0$, $H_t = 1$, and Γ_t do not exist and $D_t = \beta$ is a vector of three time-invariant parameters affecting each input. The user model file based on the general `SampleSS` template is shown below, where matrix `D` plays the role of β parameters of the regression.

```
function model= regression1(p)
model.T = 0; model.Gam = []; model.R = 0; model.Z = 0;
model.D = [p(1) p(2) p(3)];  model.C = 1; model.Q = 0;
model.H = 1; model.S = 0;
```

Since the observation noise variance is not included in the model, it should be estimated by concentrated ML. The execution of the estimation function would explicitly require the use of the concentrated ML function. Assuming that the inputs have been included in a 3 × T MATLAB matrix called `u`; the command to estimate the model is

```
 >> sys=SSmodel('y',y, 'u', u, 'model',@regression1,
                 ...'OBJ_FUNCTION_NAME', @llikc);
```

In this listing, the output–input data are `y` and `u`, respectively; the model is in function `regression1` above; and the objective function is `llikc`, with the latter "c" indicating the concentrated ML optimization.

Estimation results truncated to save space for the time series in Fig. 1 are shown when using the `SSvalidate` function. Parameter estimates are highly significant and close to their theoretical values.

```
-----------------------------------------------------
           Param     S.E.     T-test   P-value |Gradient|
-----------------------------------------------------
 p(1)    15.0710   0.0623   241.9089   0.0000   0.000000
 p(2)     3.9847   0.0646    61.7117   0.0000   0.000000
 p(3)     1.7903   0.0835    21.4530   0.0000   0.000000
-----------------------------------------------------
```

A different way to implement this regression model consists of defining matrix D as a time-varying matrix and adding the model inputs as a second input in the user function. The next listing shows a variety of alternative definitions of matrix D.

```
function model= regression2(p, u)
...
model.D = [p(1) p(2) p(3)] * u;
...
model.D = filter(1, [1 -p(5)], u);
...
model.D = 1 ./ (exp(-p(5) * u));
...
model.D = p(5) * u;
ind = find(y > 0);
model.D(1, ind) = p(6) * u(1, ind);
```

The first case is just a redefinition of the linear model through a time-varying D matrix; the second one defines the input–output relation as a transfer function model that may be generalized to any order; the third case is a general nonlinear function; the fourth case is a linear piece-wise relation, depending on whether variable y is positive or negative (in this case, y should be supplied as an additional input to the user model function). It is important to note that, as all the previous specifications only affect the definition of matrix D for modeling input–output relationships, they may be introduced in any sort of model.

The call to estimate this model below is somewhat different from the previous calls because the additional input to function regression2 ought to be told explicitly.

```
 >> sys=SSmodel('y',y, 'model',@regression2,...
                    'user_inputs', u, 'OBJ', @llikc);
```

Any of the previous regression versions have an advantage that allows the use of SSpace in a completely novel and even "mischievous" way, consisting of interpolating missing values of input variables at the same time the model parameters are estimated (note that missing values in output variables are not a problem in SS systems in contrast to missing values in input variables). This is simply solved by including all the missing values as additional parameters to estimate. For example, assuming there are two missing consecutive values in the second input at observations 200 and 201, the following line of code should be introduced prior to the definition of matrix D.

```
...
u(2, 200)= p(4); u(2, 201)= p(5);
...
```

Another way to deal with regressions is by specifying them as Dynamic Linear Regressions in which the parameters are assumed to vary over time as either Random Walks or Integrated Random Walks. In the case of three inputs, the model may be written as in Eq. (6).

$$y_t = u_t\beta_t + \epsilon_t \tag{6}$$

This model may be fit into Eq. (1) easily if the state vector is just the time-varying parameter and it is assumed to follow an independent random walk process ($\beta_{t+1} = \beta_t + \eta_t$). Then $\beta_t = \alpha_t$, $T_t = R_t = I$ (an identity matrix), $Z_t = u_t$, $C_t = 1$, $Q_t = Q$ (diagonal), and $H_t = 1$. In this case, Z_t is a time-varying system matrix.

SampleDLR helps the user to specify this type of models correctly. In particular, the model in Eq. (5) is listed below.

```
function model = dlr(p, u)
D = [1 1 1];
Q = diag(10.^p(1:3));
H = 1;
```

Here, variable D indicates, with 1's, which of the inputs are affected by time-varying parameters; Q is the diagonal covariance matrix noises affecting the time-varying parameters in such a way that big diagonal values imply big time variations, while values close to zero imply constant parameters; finally, H is the variance of the observed noise, that in this specification is concentrated out from the likelihood function (i.e., H = 1). Such specification may be used in two different ways: (i) specifying zero variances in Q matrix is effectively telling the recursive algorithms that the model is a time constant regression, and the filtered states are their least squares recursive estimation; and (ii) estimating Q matrix, a time-varying regression is estimated.

The code in the next listing produces the results below, where SSpace automatically detects that the system is a time-varying regression and therefore shows the final states as the estimates of the time-varying parameters. Variances of parameters (10.^ [p(1) p(2) p(3)]) are clearly zero, implying that the regression parameters ([State(1) State(2) State(3)]) are constant and close to the simulated ones. Further regression discussions are included in demo number 5 of SSpace.

```
sys= SSmodel('y', y, 'model', @dlr, ...
             'user_input', u', 'OBJ', @llikc);
sys= SSestim(sys);
sys= SSvalidate(sys);
```

```
----------------------------------------------------
                Param      S.E.     T-test  P-val    |Grad|
----------------------------------------------------
    p(1)  -11.7780   62.4750     0.1885  0.8506  0.000000
    p(2)  -14.8145   38.7453     0.3824  0.7025  0.000000
    p(3)  -10.1355   45.8441     0.2211  0.8252  0.000000
State(1)   15.0710    0.0626   240.6985  0.0000         -
State(2)    3.9847    0.0649    61.4029  0.0000         -
State(3)    1.7903    0.0839    21.3454  0.0000         -
----------------------------------------------------
```

Example 2: Time Aggregation in a Basic Structural Model with Trigonometric Seasonality

The well-known air passengers data from [1], but with the first five years transformed to quarterly aggregated data is shown in Fig. 2.

An appropriate model for this time series is a Basic Structural Model [7] with trigonometric seasonality that may be implemented in SSpace easily with the help of `SampleBSM` template. A version of this model may be seen in the listing below (user function `airpasBsm`).

```
function model= airpasBsm(p)
% TREND MODEL (Local Linear Trend)
TT = [1 1;0 1];
ZT = [1 0];
RT = [1 0;0 1];
QT = diag(10.^(p(3:4)));
% TRIGONOMMETRIC SEASONAL MODEL with common variance
Periods = [12 6 4 3 2.4 2];
Rho     = [1 1 1 1 1 1];
Qs      = repmat(10.^(p(1)), 1, 6);
% IRREGULAR (observed noise)
H = 10.^(p(2));
```

However, in order to handle time aggregation, this function ought to be called inside another one based on `SampleAGG` template, which contains just the correct call to the user function. This extra function needs as an extra input, the output data to locate exactly where the time aggregation takes place, signaled in the output data as standard MATLAB NaN (Not-a-Number) values.

```
function model= airpasBsm_agg(p, y)
model1 = airpasBsm(p);
```

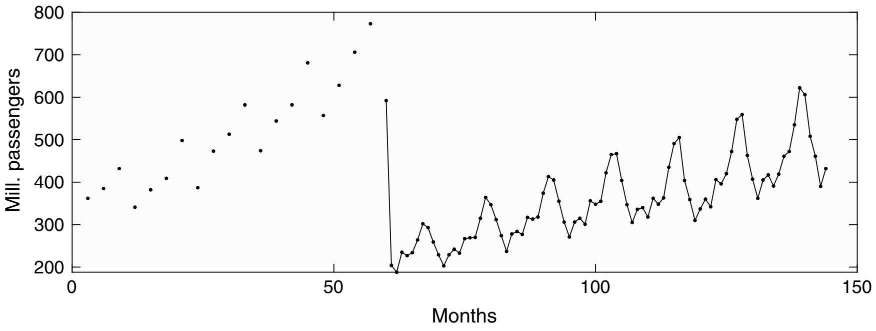


Fig. 2 Air passengers data of [1] with time aggregation

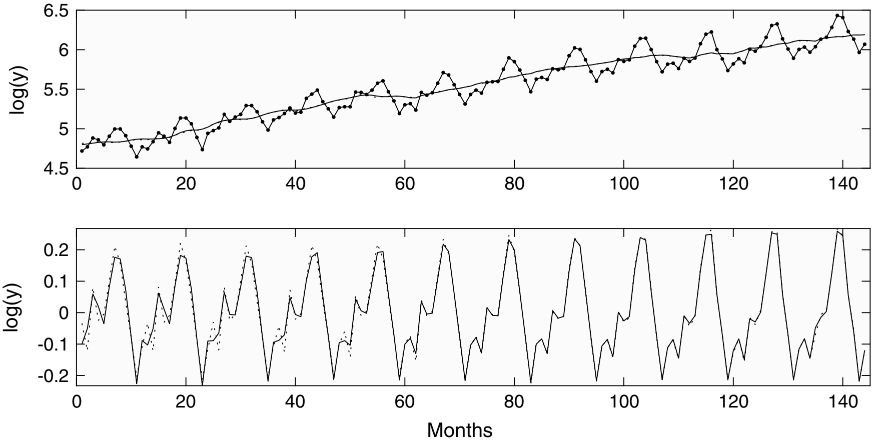


Fig. 3 Trend and log of air passengers (top panel) and seasonal component. Estimated components with time aggregation (solid lines) and without time aggregation (dotted) are shown

Smoothed trend and seasonal components obtained with this model with and without time aggregation are shown in Fig. 3. Differences are very small, meaning that the interpolation is rather appropriate.

Example 3: Demand Forecasting Comparisons

The robustness of SSpace is evaluated in this example, in which a thorough experiment is carried out. The data consists of 517 consecutive daily sales of 261 products from a Spanish franchise, specialized in selling dishes made from natural products. Figure 4 shows some typical examples of the time series in the dataset.

Although all time series are composed of integer values, the units are much larger in the top panel of Fig. 4. In fact, all observations in the bottom panel are below 12. It is well-known that in such cases, the Gaussian approximation is not appropriate and other discrete distributions (mainly Poisson) are superior in many respects (see, for example, [5]). This suggests that the sample should be split into two groups of series, i.e., those that may be treated as *continuous* with higher values per day, and the remainder, which hereafter will be referred to as *discrete* time series. There are 166 time series in the *continuous* group, i.e., 63.6% of the total and 95 (36.4%) in the *discrete* category.

The experimental setup of this example consists of the automatic identification and estimation of all forecasting methods for each time series using the initial 414 daily observations. 1–14 days ahead forecasts are then produced in a rolling experiment that advances the forecast origin one day at a time on the remaining out-of-sample observations. Thus, 90 rounds of daily forecasts were done for each product.

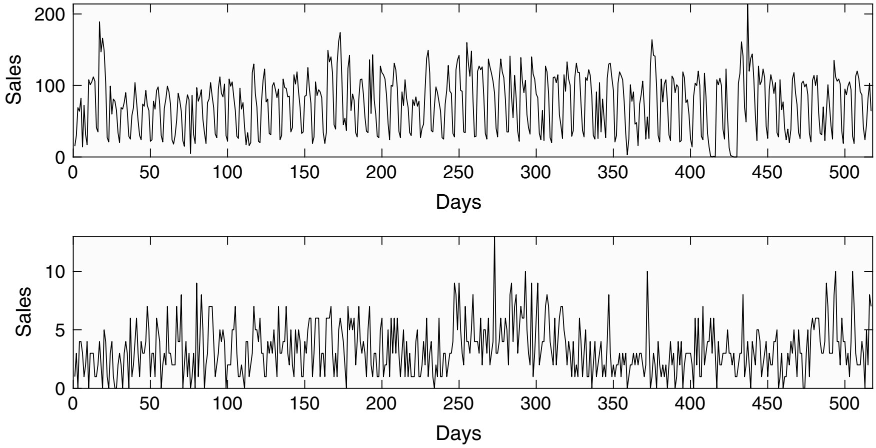


Fig. 4 Two examples of demand time series

The methods used are

- **NAIVE**: random walk.
- **AR**: pure autoregressive models with order identified with Bayesian Information Criterion (BIC).
- **ARIMA**: identified as in [8].
- **ETS**: ExponenTial Smoothing, identified as in [8].
- **UC**: Unobserved Components based on the identification of several trend, seasonal, and irregular components, based on the minimization of the BIC estimated with SSpace (using template `SampleBSM`).
- **UCp**: Poisson Unobserved Components used only for discrete time series based on the minimization of the BIC estimated with SSpace (using template `SampleEXP`).
- **MEAN**: mean combination of ARIMA, ETS, UC, and UCp (in the case of discrete time series).
- **MEDIAN**: median combination of ARIMA, ETS, UC and UCp (in the case of discrete time series).

Mean of Mean Absolute Errors across all time series and all methods are shown in Table 2 at different forecast horizons (from 1 to 14), with the best method for each forecasting horizon highlighted in boldface. The table shows clearly that UC is the method with fewer errors in continuous time series, while the UCp is the best in the case of discrete time series. MEAN and MEDIAN are often the second best. But, what is more important from the point of view of this chapter is that the results shown require repeated runs of SSpace subroutines, that worked robustly in this long experiment.

Table 2 Mean of Mean Absolute Errors for continuous and discrete time series. Best method for each forecasting horizon is highlighted in boldface

Continuous time series

	1	2	3	4	7	14
NAIVE	0.4695	1.0833	1.7437	2.4153	4.0597	8.3784
AR	0.3484	0.7198	1.0967	1.4816	2.6518	5.5116
ARIMA	0.3210	0.6634	1.0133	1.3695	2.4550	5.0598
ETS	0.3308	0.6849	1.0455	1.4117	2.5241	5.2015
UC	**0.3164**	**0.6539**	**0.9989**	**1.3500**	**2.4225**	**4.9901**
MEAN	0.3209	0.6635	1.0127	1.3691	2.4563	5.0740
MEDIAN	0.3216	0.6655	1.0163	1.3728	2.4590	5.0774

Discrete time series

	1	2	3	4	7	14
NAIVE	1.0128	2.0892	3.1674	4.2484	7.3912	14.9254
AR	0.8304	1.6736	2.5218	3.3707	5.9269	11.9996
ARIMA	0.8285	1.6634	2.5020	3.3425	5.8658	11.8557
ETS	0.8338	1.6762	2.5220	3.3694	5.9136	11.9533
UC	0.8147	1.6331	2.4542	3.2784	5.7589	11.6495
UCp	**0.8056**	**1.6161**	**2.4284**	**3.2456**	**5.7119**	**11.5940**
MEAN	0.8089	1.6242	2.4428	3.2642	5.7368	11.6101
MEDIAN	0.8104	1.6259	2.4445	3.2660	5.7378	11.6103

Conclusions

This chapter has presented SSpace, a new MATLAB toolbox for taking full advantage of the State-Space framework. SSpace is a toolbox for State-Space modeling that provides the user with the possibility to model linear Gaussian, nonlinear, and non-Gaussian systems with the most advanced and up-to-date features available in any State-Space framework, following mainly [5, 7, 14]. In addition, all system matrices are potentially time-varying and may be multivariate, several estimation methods are implemented, inputs to the system may be introduced explicitly, etc.

Further advantages are that a few functions are necessary to carry out a comprehensive analysis of time series, always used with a fixed pattern and with function names carefully chosen following mnemonic rules. However, what makes SSpace flexible, powerful, and transparent is that the user implements models directly by coding MATLAB functions. This feature makes extensions of models with non-standard properties possible, like time-varying parameters or variances, nonlinear input–output relations, etc.

The toolbox is supplied with a number of templates to carry out the time series analysis by some standard methods to avoid forcing the user to remember their respective SS forms. All these advantages, in addition to robustness, are illustrated with several worked examples taken from real data.

References

1. Box, G., Jenkins, G., Reinsel, G., Ljung, G.: Time Series Analysis: Forecasting and Control. Wiley (2015)
2. Casals, J., Garcia-Hiernaux, A., Jerez, M., Sotoca, S., Trindade, A.: State-Space Methods for Time Series Analysis: Theory, Applications and Software. Forthcoming by Chapman and Hall/CRC (2016)
3. Commandeur, J., Koopman, S., Ooms, M.: Statistical software for state space methods. J. Stat. Softw. **41**(1), 1–18 (2011). https://doi.org/10.18637/jss.v041.i01
4. Doan, T.: State space methods in rats. J. Stat. Softw. **41**(9), 1–16 (2011). https://doi.org/10.18637/jss.v041.i09
5. Durbin, J., Koopman, S.: Time Series Analysis by State Space Methods, no. 38. Oxford University Press (2012)
6. Gómez, V.: Ssmmatlab: a set of matlab programs for the statistical analysis of state space models. J. Stat. Softw. **66**(9), 1–37 (2015). https://doi.org/10.18637/jss.v066.i09
7. Harvey, A.: Forecasting, Structural Time Series Models and the Kalman Filter. Cambridge University Press (1989)
8. Hyndman, R.J., Khandakar, Y.: Automatic time series forecasting: the forecast package for R. J. Stat. Softw. **3**(27), 1–22 (2008)
9. Koopman, S., Harvey, A., Doornik, J., Shephard, N.: STAMP 8.2: Structural Time Series Analyser and Modeller and Predictor. Timberlake Consultants Limited (2009)
10. Koopman, S., Shephard, N., Doornik, J.: Statistical Algorithms for Models in State Space Form: SsfPack 3.0. Timberlake Consultants Press (2008)
11. Lucchetti, R.: State space methods in gretl. J. Stat. Softw. **41**(11), 1–22 (2011). https://doi.org/10.18637/jss.v041.i11
12. Peng, J., Aston, J.: The state space models toolbox for matlab. J. Stat. Softw. **41**(6), 1–26 (2011)
13. Petris, G., Petrone, S.: State space models in R. J. Stat. Softw. **41**(4), 1–25 (2011). https://doi.org/10.18637/jss.v041.i04
14. Taylor, C., Pedregal, D., Young, P., Tych, W.: Environmental time series analysis and forecasting with the captain toolbox. Environ. Model. Softw. **22**(6), 797–814 (2007)
15. Young, P., Pedregal, D., Tych, W.: Dynamic harmonic regression. J. Forecast. **18**(6), 369–394 (1999)

Advanced Computational Intelligence Methods for Time Series Analysis and Forecasting

Stacked LSTM Snapshot Ensembles for Time Series Forecasting

Sascha Krstanovic and Heiko Paulheim

Abstract Ensembles of machine learning models have proven to improve the performance of prediction tasks in various domains. The additional computational costs for the performance increase are usually high since multiple models must be trained. Recently, snapshot ensembles (Huang et al. in Snapshot ensembles: train 1 get M for free, (2017) [16]) provide a comparably computationally cheap way of ensemble learning for artificial neural networks (ANNs). We extend snapshot ensembles to the application of time series forecasting, which comprises two essential steps. First, we show that determining reasonable selections for sequence lengths can be used to efficiently escape local minima. Additionally, combining the forecasts of snapshot LSTMs with a stacking approach greatly boosts the performance compared to the mean of the forecasts as used in the original snapshot ensemble approach. We demonstrate the effectiveness of the algorithm on five real-world datasets and show that the forecasting performance of our approach is superior to conservative ensemble architectures as well as a single, highly optimized LSTM.

Keywords Time series · LSTM · ARIMA · Ensembles · Stacking · Meta-learning

Introduction

Estimating the future development of continuous data generated by one or more signals has been an ongoing research field of interest for various applications. For example, automated financial forecasting is vital to today's markets. Further, sensor-generated data driven by the Internet of Things requires robust methods for reliable forecasts of temporal data. Long Short-Term Memory (LSTM) [13] has proven to be an effective method for a variety of sequence learning tasks such as time series forecasting. Relying on a single LSTM, however, is prone to instability due to the dynamic

S. Krstanovic (✉) · H. Paulheim
Research Group Data and Web Science, University of Mannheim, Mannheim, Germany
e-mail: sascha@informatik.uni-mannheim.de

H. Paulheim
e-mail: heiko@informatik.uni-mannheim.de

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_7

behavior of time series data. Additionally, the optimization of LSTM parameters is a hard problem that requires time-intensive fine-tuning.

Another difficulty when dealing with time series problems lies in the slicing of the data, i.e., how many past values should be considered for training the model and generating forecasts. It is common practice to determine the top periodicity using a fast Fourier transformation and power spectra, and train one or more models based on that periodicity. This approach is prone to incompleteness because information may be encoded across patterns of varying periodicities in the series. It is also a time-consuming task as identifying the optimal sequence length is usually part of a manual preprocessing step. For these reasons, it is a challenge to create machine learning frameworks that are able to produce automated forecasts for a given series. Even a greatly tuned model fails to find important relationships in time series data if the selected time lags cannot represent these patterns. Therefore, a framework that can incorporate multiple sequence lengths is desirable.

We introduce a meta-learning approach based on snapshot ensembles that provide superior and robust forecast estimates across different datasets. In contrast to the original idea of snapshot ensembles, we do not adapt the parameters of the LSTM but leave them unchanged. Instead, we use different slices of the training data in order to escape local minima and to detect time-dependent patterns. Our proposed approach enables the automated generation of time series forecasts for a given series $y_1, ..., y_n$, including preprocessing steps like data standardization, periodicity detection, data slicing, and splitting. Hence, the amount of required manual work is greatly reduced by the proposed framework.

By sequentially training LSTMs with periodicities of decreasing strength, our algorithm is able to learn the different patterns of the respective seasonalities. This allows for higher generalization of the final model, thereby providing estimates that are robust with respect to the underlying data generation process.

The rest of this paper is structured as follows. Section "Related Work" provides an overview of existing approaches to time series forecasting and their application within ensemble frameworks. In section "Time Series Forecasting and Snapshot Ensembles", we introduce the concept of snapshot ensembles and explain our approach for their extension to the task of time series forecasting. We show that our method outperforms previous approaches on five datasets in section "Experiments". Eventually, we conclude and give an outlook on future research directions in section "Future Work and Conclusion".

Related Work

Time series forecasting is a highly common data modeling problem since temporal data is generated in many different contexts. Classical forecasting approaches are based on autoregressive models such as ARIMA, ARIMAX, and Vector Autoregression (VAR) [10, 25]. Here, a forecast estimate is dependent on a linear combination of past values and errors. Autoregressive models work well if the assumption of stationarity is true and the series is generated by a linear process [1]. On the other hand, these hard assumptions limit the effectiveness of autoregressive models if one deals with nonlinear series, as it is the case with the majority of practical time series problems.

LSTM, a particular variant of artificial recurrent neural networks (RNN), overcomes these shortcomings as it makes no assumptions about the prior distribution of the data. One can think of RNNs as regular feed-forward networks with loops in them. This enables RNNs to model data with interdependencies such as autoregression. It has been shown that artificial neural networks with one hidden layer can, in theory, approximate a continuous function arbitrarily well [14]. As the RNN gets deeper, vanishing or exploding gradients often leads to poor model performance [4, 22]. LSTMs solve this problem with a gating mechanism that controls the information flow in the neurons. LSTMs show superior performance in a variety of sequence learning tasks such as machine translation [11, 26].

Since autoregressive models perform well for linear series and neural networks for nonlinear data, there exist a number of hybrid approaches that make use of these characteristics. In those cases, the data is first split into a linear and a nonlinear component and each one is modeled independently. The individual results are then combined additively to determine the final estimate [2, 3, 28, 32].

The sequential nature of LSTMs has led to them being studied in the context of time series forecasting intensively. References [8, 19, 20] describe applications of LSTMs for forecasting tasks. References [1, 17] propose frameworks of LSTM ensembles with independently trained models. Finally, snapshot ensembles constitute a way to construct an ensemble of dependent ANNs at comparably low computational costs. A more detailed description is given in section "Introduction to Snapshot Ensembles". We extend this method to recurrent neural networks and sequential problems.

Time series analysis has also been investigated in the framework of convolutional neural networks (CNNs). Reference [5] uses an architecture inspired by the recent success of WaveNet for audio generation [27] which achieves competitive forecasting performance with relatively little training data available. A probabilistic approach that combines both RNNs and CNNs in a single framework is given in [30].

Finding periodicities in time series data is a key part in the preprocessing of time series data and proposes a major challenge for the automation of machine-generated forecasts. Reference [7] proposes a variation of the approximate string matching problem for automated periodicity detection. Reference [21] develops strategies on diversity generation and builds ensembles of the resulting models. In [6], a number of heterogeneous models are arbitrated by a meta-learner. Reference [9] applies Fourier transformations to the original data for feature generation and uses a feed-forward neural network for the modeling part based on these features. Reference [23] shifts CNN training entirely to the Fourier domain, thereby, achieving a significant speedup with practically no loss of effectiveness. Another approach that exploits Fourier transformations is given in [24]. We will use a similar methodology in the course of this paper.

Time Series Forecasting and Snapshot Ensembles

Time series data is subject to a number of properties due to interdependencies across observations:

1. Autoregression. In contrast to a machine learning setup where observations are independent of each other, sequence learning tasks are characterized by dependencies between observations. This has effects on data sampling and model evaluation as drawing completely random subsamples is not possible. Hence, a suitable sample strategy is indispensable when modeling temporal data.
2. Structural patterns and changes. Due to trend and seasonality effects, the behavior of a time series is subject to repetition and change at the same time. While similar patters may repeat over time, the frequency and intensity of those are usually not constant. This is one reason why ensemble methods are a powerful tool for time series data as each of the snapshot models incorporates information about different behavior.

Introduction to Snapshot Ensembles

Snapshot ensembles propose a novel technique to obtain an ensemble of ANNs at the same computational costs as fully training a single ANN. The central idea is that instead of training a number of independent ANNs, only one ANN must be optimized. In the process of optimization, the ANN converges to a number of different local minima. Every time the ANN reaches a local minimum, the model snapshot is stored along with its architecture and weights. The final weights of a snapshot serve as the weight initialization of the succeeding snapshot LSTM. Finally, each snapshot provides a prediction estimate and the ensemble predictor is calculated as the mean of the snapshot estimates. It was shown that this combination yields advantageous performance compared to the single best estimate [16].

Extending Snapshot Ensembles to Sequence Problems

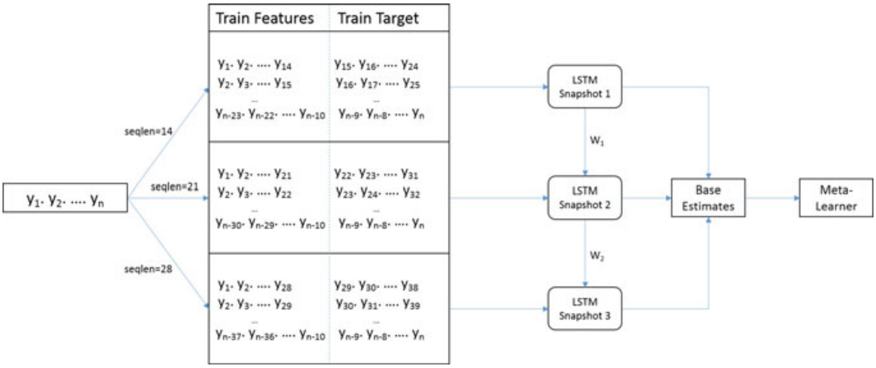
Time series forecasting can be interpreted as a sequence learning problem. Given an input sequence of scalars, the objective is to estimate the succeeding values of the sequence. An important task is to determine how many past values should be considered as the features under consideration, i.e., which slice dimension of the series allows for good model generalization. By nature, time series data is dynamic and subject to change over time, so an initial decision is not necessarily a sustainable solution. Designing ensembles of LSTM networks allows us to incorporate multiple sequence lengths into our prediction model. In the following, we explain how.

LSTMs with varying sequence lengths. By architecture, LSTMs are only capable to process sequences of equal lengths per epoch, due to the required matrix operations in the optimization process. In many applications, however, varying sequence lengths are inevitable. One example is machine translation where the length of an input sentence can be arbitrarily long [26]. Padding is usually used to overcome that problem [15]. This implicitly means that, although two models trained with even slightly different sequence lengths have a large intersection of training data, they learn different yet related patterns. This constitutes a promising setting for ensemble learning.

Locating candidate sequence lengths. In order to train a number of snapshot LSTMs with different sequence lengths, the first step is to identify the right choices of these. A naive approach is to select sequence lengths from a random distribution. To get sequence lengths that can catch effects of seasonality, we apply a fast Fourier transformation (FFT) to the training data and estimate the power spectra [29]. The motivation behind this is that the FFT is an efficient method to extract the right periodicities from a given time series. This allows the snapshots to encode different patterns, seasonalities, and other time-dependent effects in the series.

Generating a snapshot ensemble of LSTMs with varying sequences. Reference [16] conducts a variant of simulated annealing in order to adapt the learning rate and escape from local minima. In this case, a snapshot is a further optimization of its predecessor using the identical training data, which leads to a relatively low level of diversity across the snapshots. We propose another strategy in order to increase diversity: Instead of adapting the model parameters, we feed the LSTM with different slices of the data. This is possible because the dimensions of the training data must be identical within a single epoch but not for two separate epochs. Given a set $S = \{s_1, s_2, ..., s_n\}$ of different sequence lengths, we store in total n snapshots of the LSTM. After each snapshot based on s_i, the training process is continued with a different data slice through time according to s_{i+1}. The final holdout estimates of the individual snapshots are commonly combined by taking the mean of the base forecasts. This assumes that each snapshot is equally important with respect to the combination of forecasts. In order to allow for more flexibility, we extend the mean function by a meta-learner. Ridge regression has proven to be an effective choice here [31].

The process of the ensemble construction at training time is depicted in Fig. 1 for the example case $S = \{14, 21, 28\}$ and a forecasting horizon of 10. First, the training data $y_1, ..., y_n$ (75% of the total data) is split according to the most potent sequence lengths provided by the FFT (in decreasing order of FFT significance). In our experiments, we use the top 20 sequence lengths. Next, the first snapshot is trained with the respective data slices based on the first sequence length. We train each snapshot for five epochs and standardize the data by its z-transform prior to training. The base LSTM learners' architecture is set up of two LSTM layers with 64 and 128 neurons as well as 20% dropout. Adam is used as the optimizer with a learning rate of 0.001. The weight matrix of the first snapshot is then updated based on the data slices for the second sequence length, and so on. In total, training is done for $5 \cdot 20 = 100$



Train Features	Train Target
$y_1, y_2, \dots, y_{14}$ $y_2, y_3, \dots, y_{15}$ ... $y_{n-23}, y_{n-22}, \dots, y_{n-10}$	$y_{15}, y_{16}, \dots, y_{24}$ $y_{16}, y_{17}, \dots, y_{25}$ ... $y_{n-9}, y_{n-8}, \dots, y_n$
$y_1, y_2, \dots, y_{21}$ $y_2, y_3, \dots, y_{22}$ ... $y_{n-30}, y_{n-29}, \dots, y_{n-10}$	$y_{22}, y_{23}, \dots, y_{31}$ $y_{23}, y_{24}, \dots, y_{32}$ ... $y_{n-9}, y_{n-8}, \dots, y_n$
$y_1, y_2, \dots, y_{28}$ $y_2, y_3, \dots, y_{29}$ ... $y_{n-37}, y_{n-36}, \dots, y_{n-10}$	$y_{29}, y_{30}, \dots, y_{38}$ $y_{30}, y_{31}, \dots, y_{39}$ ... $y_{n-9}, y_{n-8}, \dots, y_n$

Fig. 1 LSTM snapshot training framework

epochs. After all the snapshots are trained, a ridge regression meta model learns how to combine the individual forecasts of the 20 snapshots. Analogously, during test time, all 20 base models provide their forecasts to the meta-learner, which then combines them to the final estimate for the 10-step ahead forecasts.

Experiments

We test the proposed methodology on five datasets of different kinds. We train a snapshot ensemble for each dataset where we start with the strongest periodicity according to the FFT. Subsequently, each LSTM snapshot is based on the next strongest periodicity. In total, 20 snapshots are trained. An overview of the datasets is given in Table 1 and Fig. 2. Furthermore, Fig. 3 displays the power spectrum for the sunspots series. This example shows that there exist a number of unequally well-suited periodicities. Each of these contains different patters which we aim to extract using snapshot ensembles. To show the effectiveness as well as the efficiency of our approach, the performance of the snapshot ensemble is measured against the following three baselines:

1. Independent LSTM ensemble. Instead of continuing the training process by escaping from a local minimum, the LSTM is reinitialized randomly and fed with the new data slices. Instead of n snapshots, we end up with n LSTMs whose training process was completely independent of one another. In contrast to this, a snapshot inherits its initial weights from its preceding snapshot.
2. Single optimized LSTM. The best sequence length according to the FFT is used for the optimization of a single LSTM over all epochs.
3. ARIMA with model selection based on the AIC.

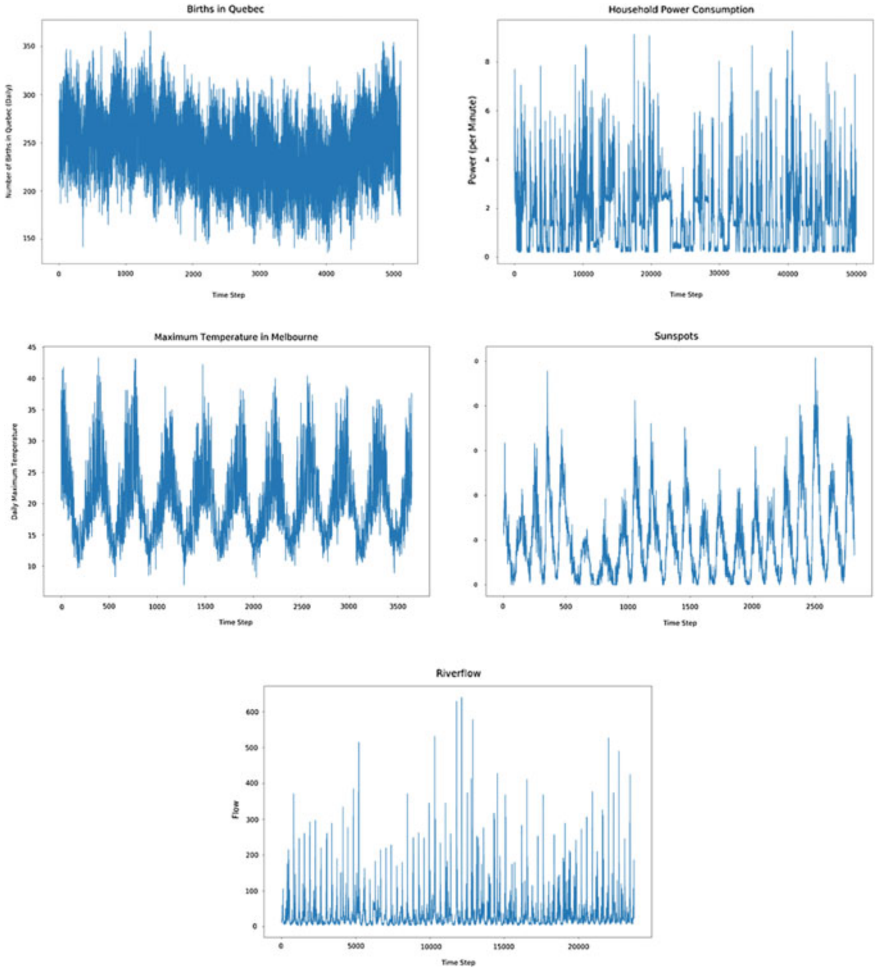


Fig. 2 Graphical data overview

Notably, the total number of epochs is identical for all the neural net approaches. Due to different slices of the training data, the total runtime of the latter approach can slightly differ from the ensemble methods in either direction.

Model Evaluation

We validate the performance of our approach on five different datasets listed in Table 1. Figure 2 illustrates the series on their original scale. Evidently, each of the datasets has its very own characteristics and dynamics. While the daily birth

Fig. 3 Power spectrum of the sunspots dataset

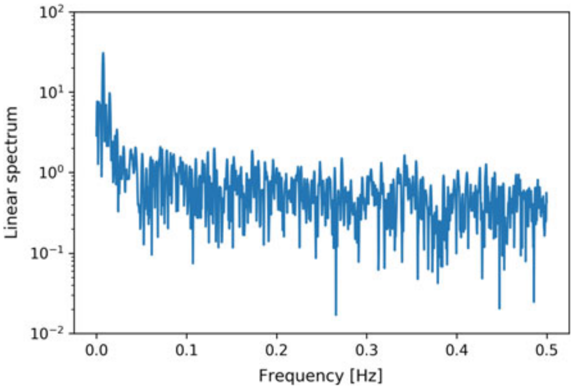


Table 1 Datasets of the experimental analysis

Data	Number of observations
Births in Quebec [12]	5,113
Household power consumption [18]	50,000
Maximum temperature in Melbourne	3,650
Number of sunspots	2,820
Riverflow	23,741

rates dataset shows signs of weak stationarity, the sensor-generated household power dataset depicts more chaotic behavior with random noises. The power dataset is sampled by the minute. River flow, a monthly sampled time series, is clearly nonstationary as well. The series of daily maximum temperatures repeats similar patterns over time as does the births data and shows clear signs of weak stationarity. Somewhere in between those cases fits the monthly sunspots data which shows seasonalities of varying strength and amplitude.

Figure 4 shows the root mean square error (RMSE)[1] on the holdout set of each dataset and method. Besides the performance of the stacked ensembles ("Snap Stack": stacked snapshot ensemble, "ClassEns Stack": stacked ensemble of independently trained LSTMs), metrics for mean ensemble forecasts ("Snap Mean", "ClassEns Mean"), and single model forecasts ("Single opt.") are shown. The key outcomes of the analysis are as follows:

- Snapshot ensembles with ridge regression as a meta-learner outperform conservative ensembles as well as the single, optimized model in all cases.The traditional ARIMA models show inferior forecasting accuracy.
- On average, the stacked snapshot ensemble performs 4.2% better than the next best baseline.

[1] https://www.qlik.com/us/products/qlik-data-market, accessed June 19, 2019.

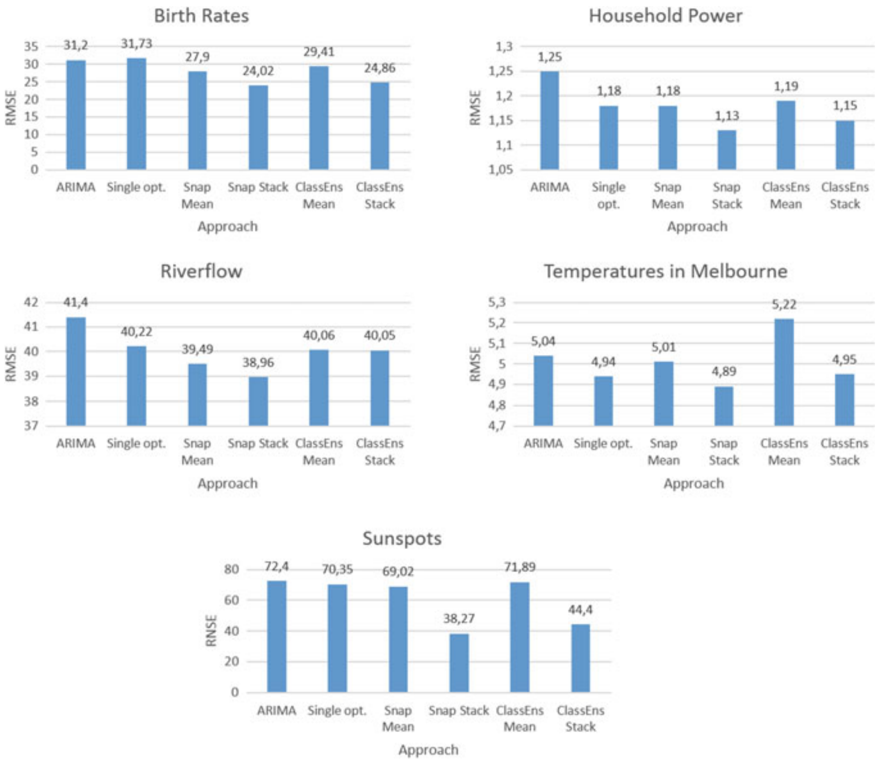


Fig. 4 Model performance

- The greatest performance gain obtained by the stacked ensemble is realized for the sunspots data. Here, the stacked snapshot ensembles outperform the next best method by 13.8%, while the performance win for the other four datasets is in a significantly lower range between 1.0% and 3.4%. Looking at the illustrated data in Fig. 2, this is an indication that our approach is particularly suitable for time series with seasonalities of varying intensity. Peaks of different amplitudes are handled well by the stacked snapshot ensemble, which a single model fails to do with a high degree of precision.
- Extending snapshot ensembles by the introduction of a meta-learner leads to a great boost in performance compared to the simple mean combiner.
- The ensemble forecasts are significantly different from the estimates of the remaining models, based on the paired t-test for significance.
- The single optimized LSTM only shows comparative performance if the structure of the dataset is approximately stationary over time, as in the case of the maximum temperatures series. This supports our hypothesis that snapshot ensembles are particularly suitable for cases where patterns are spread across multiple sequence lengths.

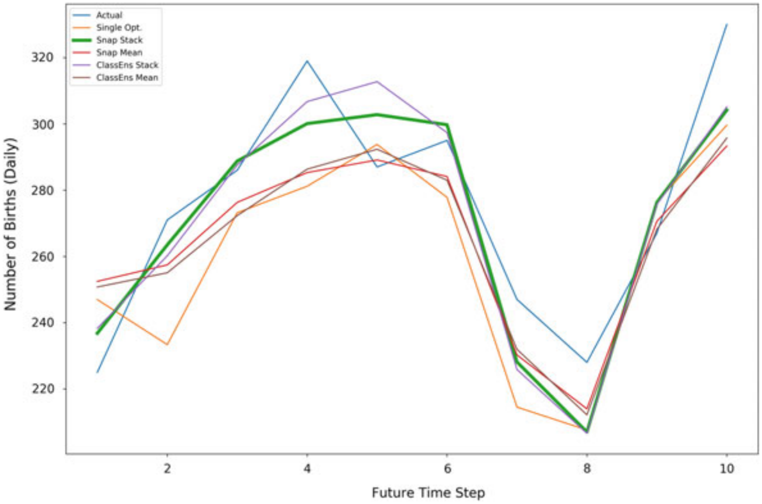


Fig. 5 Exemplary forecast

- Reslicing the input data according to the FFT after each snapshot leads to base learners with high diversity. This enables the meta-learner to exploit different knowledge that is encoded across the snapshots. As an example, the ordered FFT sequence lengths for the birth rates dataset are as follows: 365, 183, 73, 61, 37, 91, 41, 30, 10, 52, 11, 26, 852, 28, 14, 568, 341, 16, 20, and 465. This clearly shows how FFT extracts potent periodicities from the time series as the yearly and monthly seasonalities are immediately detected.

An exemplary 10-step ahead forecast is shown in Fig. 5. Here, the first holdout sequence of the birth rates series along with its model estimates is illustrated. One can see the significant improvements that are attributed to the meta-learner, leading to reduction in forecasting error.

The code for the experiments is available on GitHub.[2]

Future Work and Conclusion

Snapshot ensembles based on FFT sequence lengths are an efficient method to extract diverse patterns from data. We have shown that they yield superior forecasting performance in comparison to the standard optimization of a single LSTM and an ensemble of fully independently trained LSTMs, without the need for additional computational costs. It turned out that these results are stable across different datasets although the relative performance boost differs depending on the underlying data structure. Our

[2] https://github.com/saschakrs/TS-SnapshotEnsemble, accessed June 1, 2018.

approach enables the automated generation of robust time series forecasts without the assumption of a specified data distribution. This makes the framework a valuable application for systems that require the future estimation of one or more key performance indicators that develop over time.

There is further potential regarding the design of the ensemble architecture: Besides the configuration of the individual base learners, different combiner functions might improve the overall performance for certain problems. In addition to this, we found that five epochs per snapshot lead to good overall performance of the ensemble, however, this parameter could be higher for very complex learning tasks.

It is also possible to extend the ensemble by different model types. Integrating autoregressive models or state-space representations could increase model diversity and thereby lead to a greater performance win by the combiner function.

Finally, LSTM snapshot ensembles are currently limited to univariate time series. Evaluating their applicability to the multivariate case is another challenge worth investigating. It would also be interesting to evaluate the applicability of stacked snapshot ensembles to different sequence learning tasks such as machine translation.

References

1. Adhikari, R.: A neural network based linear ensemble framework for time series forecasting. Neurocomputing **157**(2015), 231–242 (2015)
2. Adhikari, R., Agrawal, R.K.: A linear hybrid methodology for improving accuracy of time series forecasting. Neural Comput. Appl. **25**(2), 269–281 (2014)
3. Aladag, C.H., Egrioglu, E., Kadilar, C.: Forecasting nonlinear time series with a hybrid methodology. Appl. Math. Lett. **22**(9), 1467–1470 (2009)
4. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994)
5. Borovykh, A., Bohte, S., Oosterlee, C.W.: Conditional time series forecasting with convolutional neural networks. J. Computat. Financ. (2018)
6. Cerqueira, V., et al.: Arbitrated ensemble for time series forecasting. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham (2017)
7. Elfeky, M.G., Aref, W.G., Elmagarmid, A.K.: Periodicity detection in time series databases. IEEE Trans. Knowl. Data Eng. **17**(7), 875–887 (2005)
8. Gers, F.A., Eck, D., Schmidhuber, J.: Applying LSTM to time series predictable through time-window approaches. In: Neural Nets WIRN Vietri-01, pp. 193–200. Springer (2002)
9. Gothwal, H., Kedawat, S., Kumar, R.: Cardiac arrhythmias detection in an ECG beat signal using fast fourier transform and artificial neural network. J. Biomedi. Sci. Eng. **4**(04), 289 (2011)
10. Hamilton, J.D.: Time Series Analysis, vol. 2. Princeton University Press, Princeton (1994)
11. He, Z., Gao, S., Xiao, L., Liu, D., He, H., Barber, D.: Wider and deeper, cheaper and faster: tensorized LSTMs for sequence learning. In: Advances in Neural Information Processing Systems, pp. 1–11 (2017)
12. Hipel, K.W., McLeod, A.I.: Time Series Modelling of Water Resources and Environmental Systems, vol. 45. Elsevier (1994)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
14. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)

15. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: Advances in Neural Information Processing Systems, pp. 2042–2050 (2014)
16. Huang, G., Li, Y., Pleiss, G., Li, Z., Hopcroft, J., Weinberger, K.: Snapshot ensembles: train 1 get M for free. In: Proceedings of the International Conference on Learning Representations (ICLR 2017) (2017)
17. Krstanovic, S., Paulheim, H.: Ensembles of recurrent neural networks for robust time series forecasting. In: International Conference on Innovative Techniques and Applications of Artificial Intelligence, pp. 34-46. Springer (2017)
18. Lichman, M.: UCI Machine Learning Repository (2013). http://archive.ics.uci.edu/ml
19. Lngkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. Pattern Recognit. Lett. **42**, 11–24 (2014)
20. Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long short term memory networks for anomaly detection in time series. In: Proceedings Presses Universitaires de Louvain, vol. 89 (2015)
21. Oliveira, M., Torgo, L.: Ensembles for time series forecasting. In: JMLR: Workshop and Conference Proceedings, vol. 39, pp. 360–370 (2014)
22. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning, pp. 1310–1318 (2013)
23. Pratt, H., et al.: FCNN: Fourier Convolutional Neural Networks. Machine Learning and Knowledge Discovery in Databases, Springer, Cham (2017)
24. Sharma, D., Issac, B., Raghava, G.P.S., Ramaswamy, R.: Spectral Repeat Finder (SRF): identification of repetitive sequences using Fourier transformation. Bioinformatics **20**(9), 1405–1412 (2004)
25. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and Its Applications: With R Examples. Springer, New York (2010)
26. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, pp. 3104–3112 (2014)
27. Van Den Oord, A., et al.: WaveNet: a generative model for raw audio. In: SSW (2016)
28. Wang, L., Zou, H., Su, J., Li, L., Chaudhry, S.: An ARIMA-ANN hybrid model for time series forecasting. Syst. Res. Behav. Sci. **30**(3), 244–259 (2013)
29. Welch, P.: The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. IEEE Trans. Audio Electroacoust. **15**(2), 70–73 (1967)
30. Wen, R., Torkkola, K., Narayanaswamy, B.: A Multi-Horizon Quantile Recurrent Forecaster. NIPS 2017 Time Series Workshop (2017)
31. Zhang, L., Suganthan, P.N.: Benchmarking ensemble classifiers with novel co-trained Kernal Ridge regression and random vector functional link ensembles [Research Frontier]. IEEE Computat. Intell. Maga. **12**(4), 61–72 (2017)
32. Zhang, P.G.: Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing **50**(2003), 159–175 (2003)

Change Detection for Streaming Data Using Wavelet-Based Least Squares Density–Difference

Nenad Mijatovic, Rana Haber, Mark Moyou, Anthony O. Smith and Adrian M. Peter

Abstract Here, we present a novel algorithm for detecting changes in a continuous time series stream based on the ℓ_2 distance between two distributions. The distributions are non-parametrically modeled using wavelet expansions, inspiring the name of our method: *Wavelet-based Least Squares Density–Difference* (WLSDD). Using the least squares method, we show that the ℓ_2 distance between two wavelet expanded densities results in a closed-form expression of their coefficients. This circumvents the need to evaluate the densities and, instead, allows us to work directly with the differences between the corresponding scaling and wavelet coefficients. The method demonstrated superior change detection performance on both synthetic and real datasets, stationary or nonstationary, in comparison to other competing techniques.

Keywords Nonparametric direct density–difference estimation · Wavelets · Change point detection · ℓ_2 distance

N. Mijatovic (✉) · R. Haber · M. Moyou · A. O. Smith · A. M. Peter
Florida Institute of Technology, 150 W. University Blvd, Melbourne, FL 32901, USA
e-mail: nmijatov2005@my.fit.edu
URL: https://research.fit.edu/ice/

R. Haber
e-mail: rhaber2010@my.fit.edu

M. Moyou
e-mail: mmoyou@my.fit.edu

A. O. Smith
e-mail: anthonysmith@fit.edu

A. M. Peter
e-mail: apeter@fit.edu

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_8

Introduction

With the advent of the Internet of Things (IoT), there has been a rapid increase in connected network devices that are capable of continuously monitoring their surrounding environment. These sensor observations are often streamed as time series data and enable real-time reporting of system health, status updates, and interactive engagement. In [20], the authors established eight basic rules for processing real-time streaming data, some of which motivated this research: Rule 3 required that the system has a built-in mechanism to handle imperfections and Rule 8 required the system to respond to data changes instantaneously.

In this work, we propose the *Wavelet-based Least Squares Density–Difference* (WLSDD) framework that is able to achieve these desired objectives and identify reliable change points in stationary and nonstationary data streams in an efficient manner. The WLSDD method non-parametrically models the density–difference between distributions at two time instances through a wavelet expansion of their densities. Then, based on the algebraic property that the ℓ_2 distance between two such densities is available in closed-form, we formulate a change detection algorithm that is triggered on significant deviations from this metric. Numerous experimental evaluations illustrated the outstanding performance of WLSDD over other contemporary methods.

Related Work

A recent comprehensive survey of the change point methods in [2] serves as a good description of the current state-of-the-art algorithms. Here, we provide a brief taxonomic summary of these methods, from the nonparametric approaches of change detection to the different density–difference measures.

Early works on change identification problems suggested using Kolmogorov–Smirnov [8] and Mann-Whitney U [18] nonparametric tests. Supervised methods that utilized the Bayesian framework gained popularity recently, see [1, 3]. Unfortunately, in many applications, the selection of an appropriate prior and likelihood-building blocks of the Bayesian approach is not straightforward. The authors in [25] suggested the generative additive model with two main components: a piecewise linear trend and seasonality. The breaks in the trend component were contributed to the change points. Hence, this model was called the Breaks For Additive Seasonal and Trend (BFAST) and was applied to the sensed image time series data in order to detect phonological changes caused by seasonal climatic variations, see [26]. A similar model, called prophet [24], added a holiday component to accommodate human behavior in social networks.

Many unsupervised methods monitor a measure between two different distributions; any excursion from a given threshold is labeled as a change point. The methods utilize a two-window paradigm: the data from the "referenced" window are compared to a set from the current window. Usually, estimating the density–difference is a two-step method: first, distributions related with two retrospective sliding windows are estimated, and then a difference measure is calculated based on the results from the previous step. Common examples of measures are entropy between the distributions [4], relative entropy or Kullback–Leibler (KL) divergence [10], cumulative sum [11], and cosine distance [23]. Killick et al. [14] suggested a penalized likelihood approach accompanied by a normality transformation and detrending steps to detect changes in the variance of oceanographic time series data. Kifer et al. [13] proposed a user-understandable interpretation of the change called the relativized discrepancy. A common framework for detecting both outliers and change points from nonstationary streaming data was detailed in [30]. With each new data point, the framework used a probabilistic approach to relearn the parameters of the autoregressive (AR) model by forgetting past data (exponential decay), incrementally updated a probability density function and assigned a score with respect to the baseline. Any rapid excursions from the scores indicated change points. Cha [6] comprehensively reviewed different distance measures between distributions.

Unfortunately, in many real situations, the direct estimation of probability densities may not be an easy task. In [15], the authors used nonparametric estimation based on the relative Pearson divergence in order to estimate the density–ratio estimation directly. The authors in [12] worked on estimating the logarithm of the likelihood ratio (importance) of probability densities. The improved Kullback–Leibler Importance Estimation Procedure (KLIEP) was implemented for online applications [22]. The authors presented a likelihood ratio as the linear model for basis functions. The model parameters were determined by solving the convex optimization problem derived from the Kullback–Leibler (KL) divergence criteria. For basis functions, the authors used the Gaussian kernel with a spread parameter that was determined from the cross-validation procedure. Yet another approach that calculated differences between densities, without explicitly calculating individual densities, was presented in [21]. The authors proposed an ℓ_2 distance, calculated by applying the least squares fit to a density–difference model. They applied their model to the change point detection by thresholding the density–difference distance between Kernel Density Estimators (KDEs). Their work was termed Least Squares Density–Difference (LSDD) and serves as the progenitor of our work.

Wavelet-Based Density Estimators for Batch Processing

Wavelet-based density estimators belong in the category of nonparametric orthogonal series estimators, founded in Čencov's seminal work [5]. Provided with a set of orthogonal bases $\{\varphi_k\}_{k\in\mathbb{Z}}$, we can present an unknown density function as a linear combination of orthogonal bases, $p(x) = \sum_k b_k\varphi_k(x)$, where the coefficient b_k, for of a given dataset $\{x_n\}_{n=1}^N, x_n \in \mathbb{R}$ (N is a dataset size[1]), is estimated using $\hat{b}_k = \frac{1}{N}\sum_{n=1}^N \varphi_k(x_n)$.

Čentsov's theory can be directly applied to approximate probability density functions using a wavelet representation as

$$\hat{p}(x) = \sum_{l=1}^{L} \hat{\alpha}_{j_0,l}\phi_{j_0,l}(x) + \sum_{j=j_0}^{J}\sum_{l=1}^{L_j} \hat{\beta}_{j,l}\psi_{j,l}(x), \tag{1}$$

where $\phi_{j_0,l}(x) = 2^{j_0/2}\phi\left(2^{j_0}x - l\right)$ is the scaling function (a.k.a. *father* wavelet) with the starting resolution j_0, $\psi_{j,l}(x) = 2^{j/2}\psi\left(2^j x - l\right)$ is the wavelet function (a.k.a. *mother* wavelet) with resolution j, l is the translation parameter, L is the total number of scaling translates, L_j is the total number of wavelet translates for resolution j, and J is the ending resolution for the wavelet basis (see [7, 16] for more details). Assuming that we are provided with independent and identically distributed (i.i.d.) data samples, $\{x_n\}_{n=1}^N$, the scaling and wavelet coefficients can be approximated as

$$\hat{\alpha}_{j_0,l} = \frac{1}{N}\sum_{n=1}^{N}\phi_{j_0,l}(x_n) \text{ and } \hat{\beta}_{j,l} = \frac{1}{N}\sum_{n=1}^{N}\psi_{j,l}(x_n), \tag{2}$$

and then the probability density function can be estimated using (1). We chose to work with orthogonal, compactly supported wavelet bases (such as the Haar, Daubechies, Symlets, and Coiflets basis families of order P). Vidakovic [27] provided an excellent overview of algorithms for probability density estimation.

However, a majority of algorithms assume that all N data samples are available, causing them to regard estimation as a batch process.

Wavelet-Based Density Estimators for Stream Processing

In the case of streaming data, using (2) can be an issue if the current number of samples, N, becomes too large: the new data will have no effect on the coefficients. To avoid this issue, different approaches that utilize sliding windows of size N_w, were proposed, as illustrated in Fig. 1. Wegman and Caudle [29] used an exponential smoothing technique that forced older coefficients, estimated using elderly data samples, to vanish. García-Treviño and Barria [9] recognized an inability of the exponential aging approach to adapt to moderate changes. Therefore, they suggested to update only the relevant scaling and wavelet coefficients, presented as $\hat{b}_{j,l}$ for brevity, impacted by the new datum x_n, such as

[1] N can represent a corresponding window size too, as used in section "Change Point Detection Using WLSDD"

$$\hat{b}^{n}_{j,l} = \hat{b}^{n-1}_{j,l} + \hat{b}^{n,\text{add}}_{j,l} - \hat{b}^{n,\text{remove}}_{j,l}, \tag{3}$$

$$\hat{b}^{n,\text{add}}_{j,l} = \begin{cases} \frac{\varphi_{j,l}(x_n)}{N_w} & \text{if } 2^j l \leq x_n \leq 2^j\,(l + 2P - 1) \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

$$\hat{b}^{n,\text{remove}}_{j,l} = \begin{cases} \frac{\varphi_{j,l}\left(x_{n-N_w}\right)}{N_w} & \text{if } 2^j l \leq x_{n-N_w} \leq 2^j\,(l + 2P - 1) \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

In (4) and (5), the basis function $\varphi_{j,l}$ is either the corresponding scaling of resolution j_0 or the wavelet of resolution j basis of order P. Recall that the addition term (4) is applicable to the new datum, x_n, provided it is within the support range of the appropriate jth function for the lth translation. Similarly, aging (5) is applied to the last data sample in the sliding window, x_{n-N_w}, again, provided it belongs to the support of the jth basis for the lth translation.

Density–Difference Using Wavelets

In this section, we provide theoretical and algorithmic approaches for our density–difference estimation method, giving rise to a change detection application. We seek a measure between two distributions $p_1(x)$ and $p_2(x)$, related with two i.i.d. datasets of size N, $\{x_{1n}\}_{n=1}^{N}$, and $\{x_{2n}\}_{n=1}^{N}$, respectively.

Direct Density–Difference Estimator Based on Wavelets

Similar to [21], we want to estimate the true density–difference function, $f(x)$, between the distributions $p_1(x)$ and $p_2(x)$, by solving

$$\arg\min_{g} \int (g(x) - f(x))^2\, dx, \tag{6}$$

where $g(x)$ presents the estimated density–difference function, which we represent using the wavelet expansion as

$$g\,(x) = \sum_{l} \alpha_{j_0,l}\phi_{j_0,l}(x) + \sum_{j=j_0}^{J} \sum_{l} \beta_{j,l}\psi_{j,l}(x), \tag{7}$$

or, the equivalent matrix form

$$g(x) = \boldsymbol{\Phi}^T(x)\boldsymbol{\alpha} + \sum_{j=j_0}^{J} \boldsymbol{\Psi}_j^T(x)\boldsymbol{\beta}_j, \tag{8}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^{L\times 1}$ and $\boldsymbol{\beta}_j \in \mathbb{R}^{L_j\times 1}$ are the vectors of the scaling (at starting resolution j_0) and wavelet (at resolution j) coefficients, defined as

$$\boldsymbol{\alpha} = \left[\alpha_{j_0,1} \cdots \alpha_{j_0,L}\right]^T \text{ and } \boldsymbol{\beta}_j = \left[\beta_{j,1} \cdots \beta_{j,L_j}\right]^T, \tag{9}$$

and $\boldsymbol{\Phi}(x)$ and $\boldsymbol{\Psi}_j(x)$ are the matrices of the scaling (at starting resolution j_0) and wavelet (at resolution j) functions, respectively.

Since we are interested in the estimated function $g(x)$ that minimizes the mean squares error, (6) can be simplified as

$$\arg\min_g \int \left(g^2(x) - 2g(x)f(x)\right) dx. \tag{10}$$

Using the orthonormality properties applicable for the scaling and wavelet functions defined as $\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \boldsymbol{I} \in \mathbb{R}^{L\times L}$, $\boldsymbol{\Psi}_j\boldsymbol{\Psi}_j^T = \boldsymbol{I} \in \mathbb{R}^{L_j\times L_j}$, $\boldsymbol{\Psi}_j\boldsymbol{\Psi}_k^T = \boldsymbol{0} \in \mathbb{R}^{L_j\times L_k}$ for $j \neq k$, and $\boldsymbol{\Phi}\boldsymbol{\Psi}_j^T = \boldsymbol{0} \in \mathbb{R}^{L\times L_j}$, definition of the true density–difference, $f(x) = p_1(x) - p_2(x)$, and the wavelet transformation of $g(x)$ based on (8), our optimization problem (10) can be further presented as

$$\begin{aligned}\arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta}_j} \Bigg\{&\boldsymbol{\alpha}^T\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \int \boldsymbol{\Phi}(x)\left(p_1(x) - p_2(x)\right) dx \\ &+ \sum_{j=j_0}^{J} \left(\boldsymbol{\beta}_j^T\boldsymbol{\beta}_j - 2\boldsymbol{\beta}_j^T \int \boldsymbol{\Psi}_j(x)\left(p_1(x) - p_2(x)\right) dx\right)\Bigg\}.\end{aligned} \tag{11}$$

Note that (11) can be reformulated as a dual-optimization problem that depends on the scaling and wavelet coefficients. Thus, our problem can be decoupled into two independent cases: (1) optimal scaling parameters using

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \left\{\boldsymbol{\alpha}^T\boldsymbol{\alpha} - 2\boldsymbol{\alpha}^T \left(\int \boldsymbol{\Phi}(x)p_1(x)dx - \int \boldsymbol{\Phi}(x)p_2(x)dx\right)\right\} \tag{12}$$

and (2) optimal wavelet parameters, per resolution j as

$$\boldsymbol{\beta}_j^* = \arg\min_{\boldsymbol{\beta}_j} \left\{ \boldsymbol{\beta}_j^T \boldsymbol{\beta}_j - 2\boldsymbol{\beta}_j^T \left(\int \boldsymbol{\Psi}_j(x) p_1(x) dx - \int \boldsymbol{\Psi}_j(x) p_2(x) dx \right) \right\}. \tag{13}$$

The integrals in both (12) and (13) are appropriate expectations of the scaling and wavelet functions for the given distributions $p_1(x)$ and $p_2(x)$, defined as

$$\mathbf{h}_{\boldsymbol{\Phi}}(x) = \mathbf{E}\left\{\boldsymbol{\Phi}(x)\right\}_{p_1(x)} - \mathbf{E}\left\{\boldsymbol{\Phi}(x)\right\}_{p_2(x)} \text{ and} \tag{14}$$

$$\mathbf{h}_{\boldsymbol{\Psi}_j}(x) = \mathbf{E}\left\{\boldsymbol{\Psi}_j(x)\right\}_{p_1(x)} - \mathbf{E}\left\{\boldsymbol{\Psi}_j(x)\right\}_{p_2(x)}, \tag{15}$$

where $\mathbf{E}\{\cdot\}$ is the expectation operator. For a given discrete and finite set, $\{x_n\}_{n=1}^N$, $x_n \in \mathbb{R}$, drawn from distribution $p(x)$, the expected values of both scaling and wavelet matrices become $\boldsymbol{\Phi} \in \mathbb{R}^{L\times N}$ and $\boldsymbol{\Psi}_j \in \mathbb{R}^{L_j \times N}$, defined as

$$\boldsymbol{\Phi} = \left[\boldsymbol{\phi}_{j_0}(x_1) \ldots \boldsymbol{\phi}_{j_0}(x_N)\right] \text{and}\ \boldsymbol{\Psi}_j = \left[\boldsymbol{\psi}_j(x_1) \ldots \boldsymbol{\psi}_j(x_N)\right]. \tag{16}$$

The vectors of the scaling $\boldsymbol{\phi}_{j_0}(x_n) = \left[\phi_{j_0,1}(x_n) \cdots \phi_{j_0,L}(x_n)\right]^T \in \mathbb{R}^{L\times 1}$ and wavelet $\boldsymbol{\psi}_j(x_n) = \left[\psi_{j,1}(x_n) \cdots \psi_{j,L_j}(x_n)\right]^T \in \mathbb{R}^{L_j\times 1}$ bases in (16) are calculated at different sample points x_n $(n = 1, \cdots, N)$ in the domain.

Using (16), the expectations in (14) and (15) can be approximated as

Algorithm 1: Detecting Change in Streaming Data with WLSDD

```
input  : data stream, $\{x_n\}_{n=1}^{end}$ ($n$ is a data sample index), $N_w$, $N_d$, and $thr$
output: list of change points, $\{\tau\}$
begin
    set $\tau \longleftarrow \emptyset$ and change point index $\tau_{ccp} \longleftarrow 0$;
    for $n \leftarrow 1$ to $N_d + N_w$ do
        add (4), remove (5), update (3) coeff. in current and referenced windows;
    end
    for $n \leftarrow N_d + N_w + 1$ to $end$ do
        add (4), remove (5), update (3) coeff. in current and referenced windows;
        calculate density–difference, $d_{WLSDD}$, using (2) and (24);
        if $d_{WLSDD} \geq thr$ and $n \geq \tau_{ccp} + N_d + N_w$ then
            change detected :
            update $\tau_{ccp} \longleftarrow n$ and $\tau \longleftarrow \tau \cup \tau_{ccp}$ with the change point index $n$;
        end
    end
end
```

$$\hat{\mathbf{h}}_{\boldsymbol{\Phi}} \approx \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\phi}_{j_0}(x_{1n}) - \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\phi}_{j_0}(x_{2n}), \tag{17}$$

$$\hat{\mathbf{h}}_{\boldsymbol{\Psi}_j} \approx \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\psi}_{j}(x_{1n}) - \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\psi}_{j}(x_{2n}). \tag{18}$$

It is important to note that both $\hat{\mathbf{h}}_{\boldsymbol{\Phi}} = \left[\hat{h}_{\phi_{j_0,1}} \cdots \hat{h}_{\phi_{j_0,L}}\right]^T \in \mathbb{R}^{L\times 1}$ and $\hat{\mathbf{h}}_{\boldsymbol{\Psi}_j} = \left[\hat{h}_{\psi_{j,1}} \cdots \hat{h}_{\psi_{j,L_j}}\right]^T \in \mathbb{R}^{L_j\times 1}$ are vectors, where the lth elements are calculated as

$$\hat{h}_{\phi_{j_0,l}} = \frac{1}{N}\sum_{n=1}^{N} \phi_{j_0,l}\left(x_{1n}\right) - \frac{1}{N}\sum_{n=1}^{N} \phi_{j_0,l}\left(x_{2n}\right) \text{ and} \tag{19}$$

$$\hat{h}_{\psi_{j,l}} = \frac{1}{N}\sum_{n=1}^{N} \psi_{j,l}\left(x_{1n}\right) - \frac{1}{N}\sum_{n=1}^{N} \psi_{j,l}\left(x_{2n}\right). \tag{20}$$

Using results (17) and (18), our pair of optimization problems (12) and (13) can be solved in the closed form after applying the least squares method as

$$\boldsymbol{\alpha}^* = \hat{\mathbf{h}}_{\boldsymbol{\Phi}} \text{ and } \boldsymbol{\beta}_j^* = \hat{\mathbf{h}}_{\boldsymbol{\Psi}_j}. \tag{21}$$

Therefore, the estimated density–difference, $\hat{\mathbf{g}} \in \mathbb{R}^{N\times 1}$, can be computed using

$$\hat{\mathbf{g}} = \boldsymbol{\Phi}^T \hat{\mathbf{h}}_{\boldsymbol{\Phi}} + \sum_{j=j_0}^{J} \boldsymbol{\Psi}_j^T \hat{\mathbf{h}}_{\boldsymbol{\Psi}_j}, \tag{22}$$

which is commonly referred to as the Wavelet-based Least Squares Density–Difference (WLSDD) estimator.

In order to calculate the density–difference, we define the ℓ_2-norm distance as $d_{WLSDD} = \hat{\mathbf{g}}^T \hat{\mathbf{g}}$, which, after applying (22) and the orthonormality rules, can be calculated in the following closed form

$$d_{WLSDD} = \hat{\mathbf{h}}_{\boldsymbol{\Phi}}^T \hat{\mathbf{h}}_{\boldsymbol{\Phi}} + \sum_{j=j_0}^{J} \hat{\mathbf{h}}_{\boldsymbol{\Psi}_j}^T \hat{\mathbf{h}}_{\boldsymbol{\Psi}_j}. \tag{23}$$

Using the approximations for computing coefficients based on the available datasets (2), we can observe that (17) can be presented as $\hat{\mathbf{h}}_{\boldsymbol{\Phi}} = \boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2$ and (18) as $\hat{\mathbf{h}}_{\boldsymbol{\Psi}_j} = \boldsymbol{\beta}_{1,j} - \boldsymbol{\beta}_{2,j}$, where $\boldsymbol{\alpha}_1$ and $\boldsymbol{\beta}_{1,j}$ are scaling and wavelet coefficients calculated using $\{x_{1n}\}_{n=1}^{N}$ dataset and $\boldsymbol{\alpha}_2$ and $\boldsymbol{\beta}_{2,j}$ are scaling and wavelet coefficients estimated using $\{x_{2n}\}_{n=1}^{N}$ dataset. Finally, our WLSDD density–difference measure can be calculated in terms of the coefficients as

$$d_{WLSDD} = (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2)^T (\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2) + \sum_j \left(\boldsymbol{\beta}_{1,j} - \boldsymbol{\beta}_{2,j}\right)^T \left(\boldsymbol{\beta}_{1,j} - \boldsymbol{\beta}_{2,j}\right). \quad (24)$$

Change Point Detection Using WLSDD

Our change point detection using WLSDD method is presented in Algorithm 1. In order to expose a change in the streaming data, it uses a sliding two-window approach. This strategy, as shown in Fig. 1, is organized as follows: the referenced window contains N_w samples of the stream from the past and the current window contains the last N_w samples, including the current, last seen sample, x_n of index n. We introduce delay, N_d, between the current and referenced windows (where we assume $N_d \leq N_w$ in order to detect likely change as soon as possible). At the very beginning—within the first $N_d + N_w$ samples—the algorithm is trained to learn the initial density–difference (baseline) between the referenced and current windows. For each new sample in each sliding window, we analyze the scaling and wavelet coefficients in an efficient manner. We begin with adding new scaling and wavelet coefficients using (4) and removing old coefficients using (5); then, we update all applicable coefficients by applying (3). After this update, we calculate the density–difference between the referenced and current windows using (24). Then, we compare d_{WLSDD} against the threshold value: if d_{WLSDD} falls behind the threshold, we slide both windows and the process repeats; otherwise, the algorithm reports a change at the corresponding index n. Thereafter, upon change detection, the algorithm continuously adapts by relearning the new density–differences without performing any change thresholding until the next $N_d + N_w$ samples pass.

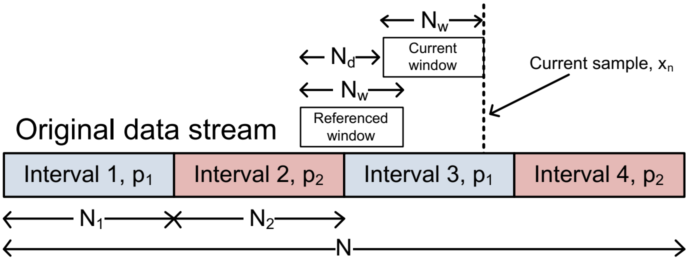


Fig. 1 Change point two-window approach with original data stream samples from two alternating distributions, p_1 and p_2

Results

In this section, we begin with an introduction of other density–difference measures and then we display the change detection results obtained from the synthetic and real streaming data.

Some Other Density–Difference Measures

In order to evaluate the performance of our WLSDD method, we compared it with LSDD as well as the Kolmogorov–Smirnov (KS) test, log-likelihood, entropy, and Kullback–Leibler divergence. For the last three methods, we assumed two different approaches for estimating distributions: (1) the underlying Gaussian distributions for log-likelihood (LL), entropy (E), and Kullback–Leibler (KL) measures and (2) histograms for log-likelihood (LLH), entropy (EH), and Kullback–Leibler (KLH) measures, see [19]. Assuming the referenced p_r and current p_c distributions, other distance measures of interest are defined as

$$LL \text{ or } LLH = \sum_i \log_2 \frac{p_{ci}}{p_{ri}}, \tag{25}$$

$$E \text{ or } EH = -\sum_i (p_{ri} - p_{ci}) \log_2 (p_{ri} - p_{ci}), \text{ and} \tag{26}$$

$$KL \text{ or } KLH = \sum_i p_{ci} \log_2 \frac{p_{ci}}{p_{ri}}. \tag{27}$$

In (25), (26), and (27), index i represents either the current sample index in the case of the LL, E, and KL methods (Gaussian assumptions) or the corresponding histogram bin in the case of the LLH, EH, and KLH methods. The same two-window approach, as introduced in section "Change Point Detection Using WLSDD", is used to calculate other density–difference measures.

Synthetic Static Dataset

First, to assess the performance of our WLSDD method, we observed the density–difference value obtained by comparing the samples from two different distributions in the offline setup. We generated samples drawn from different densities constructed using a mixture of Gaussians, see [28]. From each distribution, we drew 2,000 samples and calculated the density–differences using WLSDD and LSDD (refer to [21]). In Table 1, we compared the density–difference values between the two methods. The table contains the optimal wavelet family and the starting resolution j_0 for WLSDD, obtained using the cross-validation method.

Table 1 Density–Difference (DD) between static data from two distributions, p_1 and p_2. Overall, our WLSDD method estimates true DD more accurately when compared to LSDD

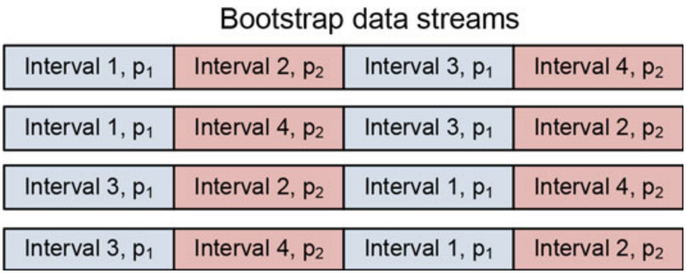
id	Distribution p_1	Distribution p_2	True DD	LSDD	WLSDD	Wavelet	j_0
1	Asym. Claw	Asym. Dbl. Claw	0.059	0.064	**0.063**	coif1	1
2	Str. Skewed Uni.	Sep. Bimodal	0.590	0.560	**0.591**	sym7	3
3	Bimodal	Trimodal	0.007	0.005	**0.007**	coif3	1
4	Bimodal	Skewed Bimodal	0.037	0.052	**0.050**	db5	1
5	Claw	Dbl. Claw	0.106	0.093	**0.109**	sym10	2
6	Gaussian	Kurtotic Uni.	0.257	0.209	**0.234**	coif2	2
7	Asym. Claw	Dbl. Claw	0.052	**0.058**	0.064	sym10	3
8	Skewed Uni.	Str. Skewed Uni.	0.838	0.810	**0.841**	sym7	3
9	Bimodal	Sep. Bimodal	0.111	0.118	**0.117**	coif3	1
10	Skewed Bimodal	Sep. Bimodal	0.162	0.183	**0.179**	db5	0
11	Gaussian	Gaussian	0.000	**0.000**	0.011	db8	1
12	Gaussian	Skewed Gaussian	0.125	**0.139**	0.140	db8	1
13	Asym. Claw	Kurtotic Uni.	0.412	0.371	**0.413**	coif2	5
14	Asym. Dbl. Claw	Dbl. Claw	0.004	0.001	**0.004**	sym7	4
15	Asym. Dbl. Claw	Skewed Bimodal	0.036	0.047	**0.040**	db2	0
16	Skewed Uni.	Trimodal	0.142	0.146	**0.141**	coif3	0
17	Skewed Uni.	Claw	0.214	0.219	**0.215**	db6	2
18	Claw	Kurtotic Uni.	0.265	0.238	**0.265**	db6	5
19	Str. Skewed Uni.	Kurtotic Uni.	0.999	0.920	**0.992**	db7	4
20	Dbl. Claw	Trimodal	0.009	0.008	**0.009**	db9	2

Overall, our method expressed results that were similar to the true density–difference; in 17 out of 20 tests, WLSDD outperformed the LSDD results. These are the expected results since the wavelet-based density estimators perform better when compared to the kernel-based ones (refer to [17] for more details).

Synthetic Streaming Dataset with Alternating Distributions

In order to evaluate the change point detection performance of our method in the streaming setup, we fabricated synthetic datasets. Since we were only interested in change detection, we suggested the following bootstrap approach in order to estimate the performance of our method on the relatively small number of change points available from the synthetic streams.

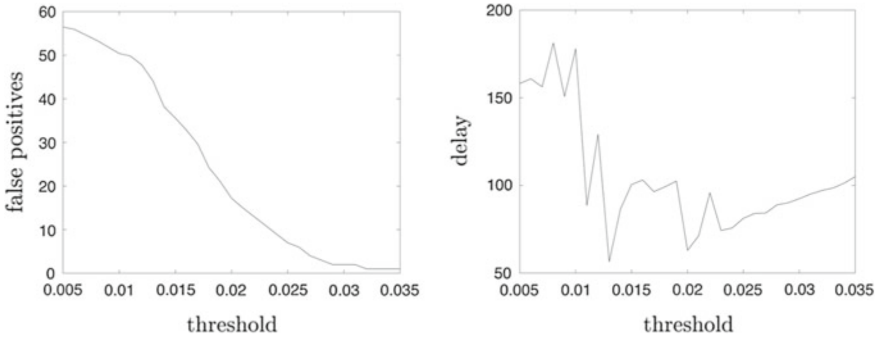
First, we generated the original synthetic streams as follows: (1) 10,000 samples were drawn from each pair of distributions, p_1 and p_2, labeled with appropriate *id*s in Table 1; (2) the samples from each distribution were divided into five different



Bootstrap data streams

Interval 1, p_1	Interval 2, p_2	Interval 3, p_1	Interval 4, p_2
Interval 1, p_1	Interval 4, p_2	Interval 3, p_1	Interval 2, p_2
Interval 3, p_1	Interval 2, p_2	Interval 1, p_1	Interval 4, p_2
Interval 3, p_1	Interval 4, p_2	Interval 1, p_1	Interval 2, p_2

Fig. 2 Bootstrap approach for determining an optimal change detection threshold (permutations of original stream intervals)



(a) False positives vs. threshold

(b) Delay vs. threshold

Fig. 3 Optimal threshold using bootstrap for the bimodal and separated bimodal dataset: **a** larger threshold (>0.3) assures smaller number of false positives and **b** faster change detection

groups, called intervals that contained $N_1 = N_2 = 2{,}000$ random samples from a given distribution; (3) the intervals from different distributions were alternated, as illustrated in Fig. 1 (to preserve space, only four intervals are shown). For example, a dataset with $id = 9$ in Table 1 was constructed with alternating bimodal and separated bimodal intervals. Based on our design for the synthetic datasets, we expected WLSDD to detect changes every 2,000 samples.

Then, using the original synthetic data stream (see Fig. 1), we created bootstrap streams by permuting appropriate intervals for the given distributions, while keeping an alternating approach between two distributions and true change points at multiples of 2,000 samples. Our approach is shown in Fig. 2 where the desired bootstrap data streams were obtained from the original stream (again, only four intervals are presented in Figs. 1 and 2 for illustration purposes).

Next, utilizing the bootstrap data streams, we calculated the performances for different threshold values (swept from 0 to true DD values, as listed in Table 1) as the following: the average accuracy A_b was calculated as the ratio between the number of accurately detected changes and the true number of change points across the bootstrap datasets; the average number of false positives F_b was found as the mean of the faulty detected change points across the bootstrap datasets (any suggested change, $N_d + N_w$ samples after the true change point within an interval, was considered as a faulty detection); and the average time of delay D_b was calculated as the mean difference between the positively detected change instances and the appropriate true change instances. For the optimal density–difference threshold, thr_{opt}, we selected a value for which the average accuracy was high and the average number of false positives and average delay values were low. The optimal threshold values for different bootstrap datasets are summarized in Table 2, section (1). The optimal window size N_w and offset N_d were obtained using cross-validation. We chose relatively low values for both the referenced and current window sizes, N_w, as well as the offset, N_d (Fig. 3).

Finally, following our methodology of generating the synthetic datasets for the bootstrap approach, we created similar testing synthetic data streams that contained 10,000 samples in total, with 10 alternating intervals with random data from two different distributions (as per pair *id*s from Table 1). Thus, each dataset expressed nine true change points every 1,000 samples. For our WSLDD approach, we used the optimal threshold values, thr_{opt}, obtained from our bootstrap method; refer to Table 2, section (1). For LSDD, we used the same thresholds as for WLSDD. For all other methods, we set the appropriate thresholds manually.

In Fig. 4, we illustrate the performance of all compared algorithms using the synthetic dataset that alternates between bimodal and separated bimodal intervals ($id = 9$ distributions from Table 1). Our method detected changes soon after the true change points, without any false positives. Many methods either erroneously reported changes or failed to detect some alterations between different intervals.

In summary, our method expresses the highest accuracy $\hat{A}$ in detecting changes in the 10 synthetic datasets; see Table 2 section (2). Also, it shows the lowest delay $\hat{D}$ in change detection, making it suitable for applications that need to respond rapidly to changes. Along with LSDD, our WLSDD method reports the least number of false positives $\hat{F}$. All other methods display very inconsistent results for different data streams.

Real Datasets

Here, we tested change difference measures on two real datasets, called Well-log and CENSREC-1-C speech corpus, respectively, both of which are popular among researchers interested in change detection.

First, the Well-log dataset contains sudden abruptions in the mean values of nuclear magnetic responses that were caused by the different geophysical structures of the rocks supporting the well, see [1]. In order to react faster to changes, we defined smaller values for our current and referenced window sizes of 40 samples with a delay of 20 samples between the windows (refer to Algorithm 1). For WLSDD, we used the Symlets wavelet family with the scaling functions at starting resolution $j_0 = -2$. With a carefully specified threshold, our method was capable of reliably detecting change points with little delay (due to the two-window approach),

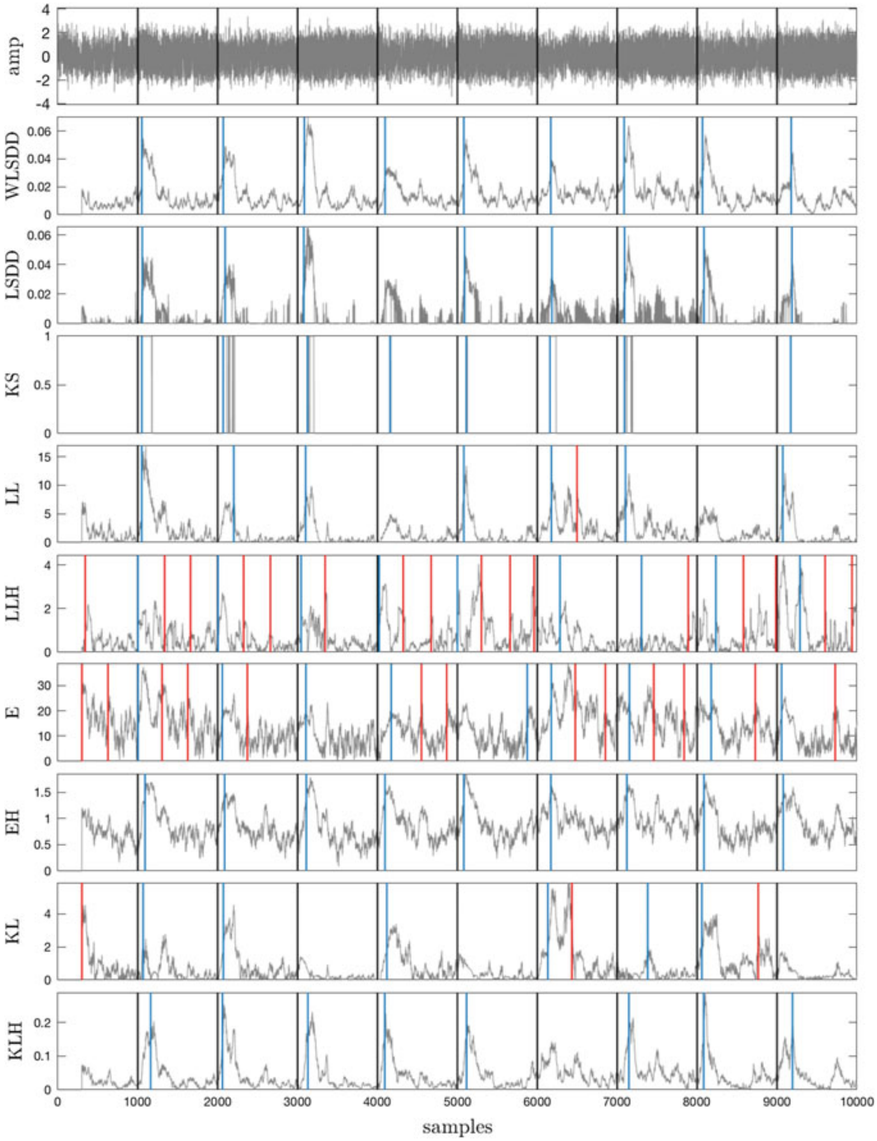


Fig. 4 Change point detection using nine different methods (in gray). The top figure (*amp*) presents the synthetic data samples (bimodal and separated bimodal alternating intervals, $id = 9$). The true change points, detected change points, and false-positive change points are indicated by the black, blue, and red vertical lines, respectively. *Our WLSDD method successfully detected all of the change points.* LSDD, KS, LL, KL, and KLH all missed at least one change point. LL, LLH, E, and KL faultily reported non-existing change points

Table 2 Performance results for change point detection. (1) Using the bootstrap method, the estimated WLSDD accuracy A_b, false positives F_b, and delay D_b are presented for the optimal threshold thr_{opt}. (2) The accuracy A, total number of false positives F, and averaged delay D for different methods and data streams, are summarized. WLSDD shows the best overall accuracy $\hat{A}$ and delay $\hat{D}$, and along with LSDD has the least number of false positives (average $\hat{F}$ and total F_t)

	(1)WLSDD Bootstrap					(2) Change Point Detection Results									
id	N_w	N_d	thr_{opt}	Results			WLSDD	LSDD	KS	LL	LLH	E	EH	KL	KLH
1	300	200	0.024	A_b	0.867	A	1.000	1.000	1.000	1.000	0.375	1.000	0.875	0.625	0.625
				F_b	1.333	F	1	1	2	1	0	1	6	2	0
				D_b	137.856	D	140.625	215.125	129.000	120.750	202.667	150.000	111.857	172.200	308.400
2	200	100	0.085	A_b	1.000	A	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.556	1.000
				F_b	3.000	F	0	0	1	0	0	2	10	0	0
				D_b	60.178	D	49.778	77.667	37.889	60.333	110.222	51.667	100.000	130.400	64.222
3	500	400	0.008	A_b	1.000	A	1.000	1.000	0.444	0.778	0.889	1.000	1.000	0.778	1.000
				F_b	8.125	F	0	0	0	0	0	1	1	0	0
				D_b	431.844	D	315.222	383.556	399.250	327.000	500.625	596.889	326.444	262.143	423.333
4	500	400	0.010	A_b	0.889	A	1.000	1.000	1.000	1.000	0.889	1.000	1.000	1.000	0.778
				F_b	6.200	F	0	0	0	0	2	0	2	0	0
				D_b	226.600	D	147.556	202.444	181.889	229.222	599.125	224.444	406.000	254.556	416.714
5	300	200	0.048	A_b	1.000	A	1.000	0.556	1.000	0.889	1.000	1.000	1.000	0.667	1.000
				F_b	0.000	F	0	0	2	0	9	2	5	3	2
				D_b	137.222	D	144.333	152.000	143.111	183.625	165.222	140.000	111.333	123.667	354.556
6	200	100	0.056	A_b	1.000	A	1.000	0.778	0.556	1.000	1.000	1.000	1.000	0.778	1.000
				F_b	1.000	F	0	0	2	9	12	21	9	5	15
				D_b	69.311	D	84.444	94.571	149.000	163.333	220.000	137.889	191.667	103.000	153.556
7	400	300	0.047	A_b	0.889	A	1.000	0.556	1.000	1.000	1.000	0.889	0.778	0.667	1.000
				F_b	3.200	F	0	0	0	0	2	2	5	2	5
				D_b	196.175	D	205.333	260.200	148.667	149.333	478.444	270.000	305.571	199.667	335.444
8	200	100	0.100	A_b	1.000	A	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
				F_b	0.833	F	0	0	1	0	0	0	0	0	0
				D_b	49.350	D	51.778	70.778	27.333	56.444	81.667	52.000	66.778	79.333	77.000
9	200	100	0.032	A_b	1.000	A	1.000	0.889	0.889	0.778	1.000	1.000	1.000	0.667	0.889
				F_b	1.000	F	0	0	0	1	16	13	0	3	0
				D_b	97.089	D	98.000	108.125	116.875	112.571	131.222	196.889	101.778	138.000	122.250
10	200	100	0.019	A_b	1.000	A	1.000	1.000	1.000	1.000	1.000	0.889	1.000	0.556	1.000
				F_b	2.000	F	0	0	0	0	17	1	0	5	0
				D_b	68.822	D	56.889	69.000	66.444	66.000	154.444	138.375	76.667	64.000	76.667
						$\hat{A}$	**1.000**	0.878	0.889	0.945	0.915	0.978	0.965	0.729	0.929
						$\hat{F}$	**0.100**	**0.100**	0.800	1.100	5.800	4.300	3.800	2.000	2.200
						F_t	**1**	**1**	8	11	58	43	38	20	22
						$\hat{D}$	**129.396**	163.347	139.946	146.861	264.364	195.815	179.809	152.697	233.214

as displayed in Fig. 5. Similarly, LL detected changes, but the amplitude range of the measure was very large, making it very difficult to specify a threshold value. On the other hand, LSDD, KS, and KL can suffer from some false positive change point detection: LSDD between 350 and 425, KS around the 315 point mark, and KL in many ranges (from 125 to 159, 290 to 310, etc.)

Next, we tested our approach on audio signals used for speech detection in public establishments from the Noisy Speech Detection Evaluation Environment

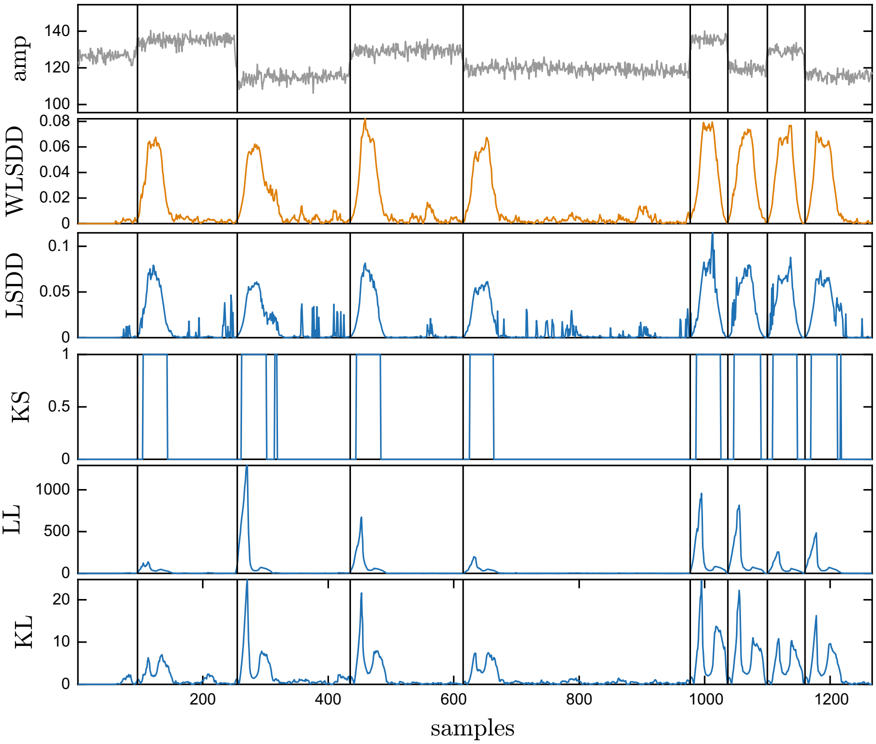


Fig. 5 Change point detection measures using the Well-log dataset, presented as the top figure (amp) with changes in mean values. The manually annotated change points are illustrated as the black vertical lines. Note that our WLSDD method provides smooth distance measures, whereas LSDD, KS, and KL are much more jittery in areas where no change points are available

(CENSREC-1-C[2]) corpus. Different silent and speech intervals alternated in the time series signals from the corpus. The change detection measures, using both WLSDD and LSDD methods, are presented in Fig. 6. The top figure presents a part of the original audio signal to allow for easier representation. The silence and speech periods were manually segregated with change points presented as the black vertical lines on all three figures. Our method perfectly detected the speech signals, while LSDD reported high values in the silence (low noise) intervals. Furthermore, it is important to note that the change measures reported by WLSDD follow natural speech variations during the spoken intervals due to its wavelet-based nature.

[2] http://research.nii.ac.jp/src/en/CENSREC-1-C.html.

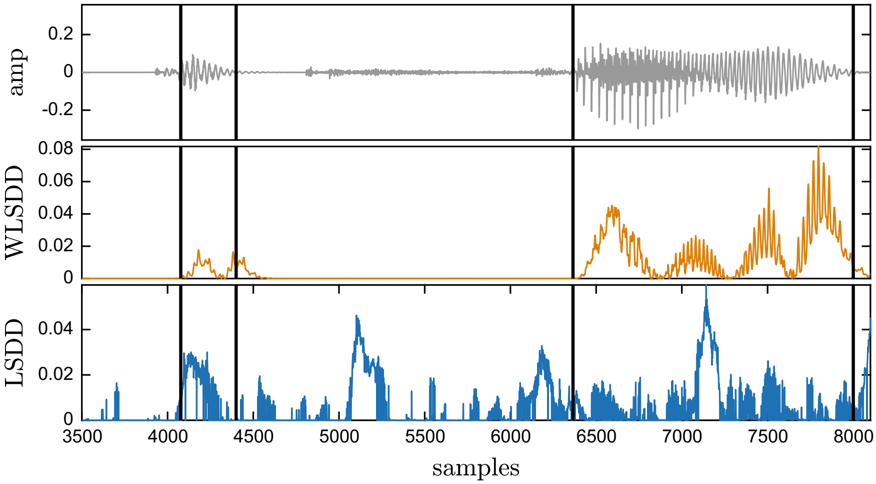


Fig. 6 Change point detection measures using the CENSCREC-1-C speech data (the only subset of data is presented). The top figure (amp) presents the original dataset (noise + silent and speech periods). The manually annotated change points are shown as the black vertical lines. Note that our WLSDD method is less sensitive to noise when compared to LSDD (around 4800 and 5730)

Conclusion

In this paper, we presented a new method for determining the difference between two distributions using nonparametric wavelet expansions. Our WLSDD method was applied to a change point application, performing well on both synthetic and real data sets. As well as outperforming all other methods, it retained the highest point detection accuracy and lowest detection delay. Furthermore, we tried with the best performer for the lowest false positive detection among all competitors. In our future work, we plan to investigate regularization in order to mitigate the noise impact on our ℓ_2 density–difference measure.

References

1. Adams, R.P., MacKay, D.J.: Bayesian online changepoint detection (2007). arXiv preprint arXiv:0710.3742
2. Aminikhanghahi, S., Cook, D.J.: A survey of methods for time series change point detection. KAIS **51**(2) (2017)
3. Barry, D., Hartigan, J.A.: A Bayesian analysis for change point problems. JASA **88**(421) (1993)
4. Berthold, M.R., Hand, D.J.: Intelligent Data Analysis: An Introduction. Springer (2007)
5. Čencov, N.N.: Evaluation of an unknown distribution density from observations. Soviet Math. **3** (1962)
6. Cha, S.: Comprehensive survey on distance/similarity measures between probability density functions. City **1**(2) (2007)

7. Daubechies, I.: Ten Lectures on Wavelets. In: CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM (1992)
8. Deshayes, J., Picard, D.: Off-line statistical analysis of change-point models using non parametric and likelihood methods. In: Detection of Abrupt Changes in Signals and Dynamical Systems. Springer (1985)
9. García-Treviño, E.S., Barria, J.A.: Online wavelet-based density estimation for non-stationary streaming data. CSDA **56**(2) (2012)
10. Hershey, J.R., Olsen, P.A.: Approximating the Kullback-Leibler divergence between Gaussian mixture models. In: Proceedings of ICASSP, vol. 4. IEEE (2007)
11. Inclan, C., Tiao, G.C.: Use of cumulative sums of squares for retrospective detection of changes of variance. JASA **89**(427) (1994)
12. Kawahara, Y., Sugiyama, M.: Change-point detection in time-series data by direct density-ratio estimation. In: ICDM (2009)
13. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proceedings of International Conference on VLDB, vol. 30 (2004)
14. Killick, R., Eckley, I.A., Ewans, K., Jonathan, P.: Detection of changes in variance of oceanographic time-series using changepoint analysis. Ocean Eng. **37**(13) (2010)
15. Liu, S., Yamada, M., Collier, N., Sugiyama, M.: Change-point detection in time-series data by relative density-ratio estimation. Neural Netw. **43** (2013)
16. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE trans. PAMI **11**(7) (1989)
17. Peter, A.M., Rangarajan, A.: Maximum likelihood wavelet density estimation with applications to image and shape matching. TIP **17** (2008)
18. Pettitt, A.: A non-parametric approach to the change-point problem. Appl. Stats (1979)
19. Sebastião, R., Gama, J.: Change detection in learning histograms from data streams. In: Portuguese Conference on AI. Springer (2007)
20. Stonebraker, M., Çetintemel, U., Zdonik, S.: The 8 requirements of real-time stream processing. ACM Sigmod Rec. **34**(4) (2005)
21. Sugiyama, M., Kanamori, T., Suzuki, T., Plessis, M.C.d., Liu, S., Takeuchi, I.: Density-difference estimation. Neur. Comp. (2013)
22. Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P.V., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: Advances in NIPS (2008)
23. Tan, P., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson (2007)
24. Taylor, S.J., Letham, B.: Forecasting at scale. Am. Stat. **72**, 1 (2018)
25. Verbesselt, J., Hyndman, R., Newnham, G., Culvenor, D.: Detecting trend and seasonal changes in satellite image time series. Rem. Sens. Environ. **114**(1) (2010)
26. Verbesselt, J., Hyndman, R., Zeileis, A., Culvenor, D.: Phenological change detection while accounting for abrupt and gradual trends in satellite image time series. Rem. Sens. Environ. **114**(12) (2010)
27. Vidakovic, B.: Statistical Modeling by Wavelets, vol. 503. Wiley (1999)
28. Wand, M.P., Jones, M.C.: Comparison of smoothing parameterizations in bivariate kernel density estimation. ASA **88**(422) (1993)
29. Wegman, E.J., Caudle, K.A.: Density estimation from streaming data using wavelets. In: Proceedings in Compstat. Springer (2006)
30. Yamanishi, K., Takeuchi, J.I.: A unifying framework for detecting outliers and change points from non-stationary time series data. In: International Conference on KDD (2002)

Selection of Neural Network for Crime Time Series Prediction by Virtual Leave-One-Out Tests

Stanisław Jankowski, Zbigniew Szymański, Zbigniew Wawrzyniak, Paweł Cichosz, Eliza Szczechla and Radosław Pytlak

Abstract The goal of this paper is the application of the virtual leave-one-out methodology to the selection of optimal neural network structure for time series prediction. The experiments are performed on the real dataset of spatiotemporal crime incidence for forecasting in the time coordinate. Due to the idea of local linearization, the estimation of generalization can be obtained in analytical form; hence, the method is computationally efficient.

Keywords Neural network · Virtual leave-one-out · Crime forecasting · Influential statistics

Introduction

Large-scale computing infrastructures and sensing technologies have produced a variety of spatial and time-dependent datasets exploiting human dynamics data [1]. The areas of urban computing [2] are intimately connected with transportation systems, social applications, and public safety and security. In the latter issue, understanding of criminal patterns within the crime datasets by the model of hot spots [3]

S. Jankowski · Z. Szymański (✉) · Z. Wawrzyniak · P. Cichosz · R. Pytlak
Warsaw University of Technology, Warsaw, Poland
e-mail: z.szymanski@ii.pw.edu.pl

S. Jankowski
e-mail: sjankows@elektron.elka.pw.edu.pl

Z. Wawrzyniak
e-mail: z.wawrzyniak@ise.pw.edu.pl

P. Cichosz
e-mail: p.cichosz@elka.pw.edu.pl

R. Pytlak
e-mail: r.pytlak@mini.pw.edu.pl

E. Szczechla
Scott Tiger S.A., Warsaw, Poland
e-mail: eliza.szczechla@tiger.com.pl

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_9

results in the possibility of forecasting of crime [4]. The widely used assumption of hot spots and sophisticated spatiotemporal statistical models have been proposed for criminal events [5]. Temporal constraints are a severe determinant in spatial-temporal patterns of property crime in conjunction with the locations of hot spots [6]. Prediction of criminal events over time is associated with chronological sequence of observations (time series, TS). The developed TS analysis techniques for forecasting based on machine learning data-driven approaches could perform well [7].

The data are preprocessed by using the ARIMA model of time series forecasting [8]. The data are analyzed by autocorrelation and partial autocorrelation functions. It can be stated that the raw time series should be differentiated in order to remove the trend. It can be concluded that time series can be modeled as autoregression model. Neural networks are used to improve the quality of prediction by nonlinear components [9–11].

The goal of this paper is to apply the virtual leave-one-out methodology (VLOO) to the selection of the optimal neural network structure for time series prediction. Our experiments are performed on a real dataset of spatiotemporal crime incidence for forecasting in the time coordinate [12, 13].

The main task is to select the optimal network structure: the number of inputs—data from past periods and the number of hidden neurons needed to represent the nonlinear function of a predictor. The criterion of selection is the expected generalization capability usually based on test dataset results. The cross-validation is considered as a very efficient method, often in the form of leave-one-out. However, this approach requires retraining of the neural network for each example of the training set [14, 15].

Analyzed Dataset

Input Data

The original dataset contained 615,961 records describing events from unstructured police reports. Crime data used for this study were extracted from anonymized internal police records provided for the purpose of scientific research funded by grant of Polish National Office for Research and Development. The only data fields used in the study were date and type (the data structure is shown in Table 1. The data studied in this work cover the period 2008 to the end of 2014. The anonymized records contain information about 12 types of events such as robbery, burglary, misdeed, violence, and interventions not assigned to any of the categories listed.

The original incident count data were aggregated by day (Fig. 1)—the aggregation was performed by summing up the total number of events and the events of individual types. The processed data were analyzed in subsequent stages of the presented work.

For the sake of simplicity in this project, we do not consider the crime type forecasting problem.

Table 1 Description of data record

Field name	Size	Description
Src	varchar(10)	Source id
Date	Datetime	Date of the event
Day of week	tinyint(4)	Day of week
Type	varchar(20)	Event type
County	varchar(30)	–
Free_kids	tinyint(4)	Boolean flag indicating pupil free day
Free_aldults	tinyint(4)	Boolean flag indicating a free day
Long_weekend	tinyint(4)	Boolean flag indicating extended weekend
National_holiday	tinyint(4)	Boolean flag indicating national holiday
Religious_holiday	tinyint(4)	Boolean flag indicating religious holiday
geo_long	Double	Event location—WGS84 longitude
geo_lat	Double	Event location—WGS84 latitude
Political_assembly	tinyint(4)	Boolean flag indicating political assembly
Youth_assembly	tinyint(4)	Boolean flag indicating youth assembly
Mass_event	tinyint(4)	Boolean flag indicating other mass event
Supporter	tinyint(4)	Boolean flag indicating football match

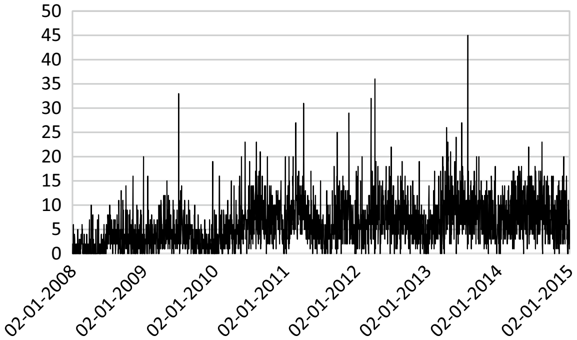


Fig. 1 Original time sequence—the number of events per day (misdeed events)

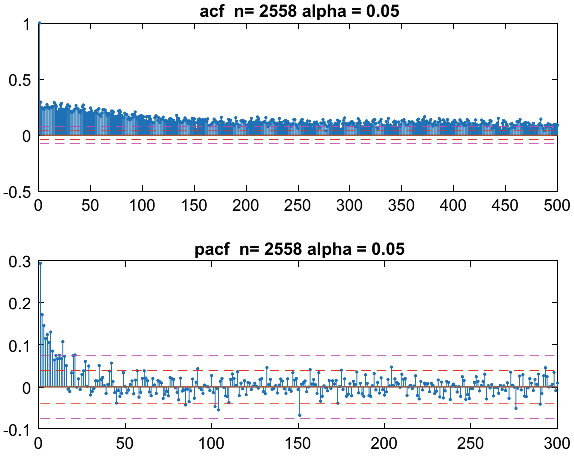
Data Processing

Preprocessing of time series for neural network modeling involves removing of the mean value, differentiation, and normalization. In the first stage, the mean value 6.22 was subtracted from the time sequence shown in Fig. 1.

The autocorrelation ACF and partial autocorrelation PACF function graphs were performed (Fig. 2) before processing the data to check whether the differentiation of the time series and removal of the seasonal component should be performed.

The stationary time series manifests fluctuations around the mean value and its autocorrelation function quickly decreases to 0. If the series has positive values

Fig. 2 Autocorrelation function ACF and partial autocorrelation function PACF of the time series



of the autocorrelation function for many delay values (e.g., 10 or more), it means differentiation should be performed. Differentiation tends to introduce negative autocorrelations. Figure 3 shows differentiated times series according to the formula

$$x_{diff}(n) = x(n) - x(n-1) \tag{1}$$

The last processing step was data normalization according to the formula

$$x_{diff_{norm}}(n) = \frac{x_{diff}(n)}{std\left(x_{diff}\right)} \tag{2}$$

Fig. 3 Differentiated time series

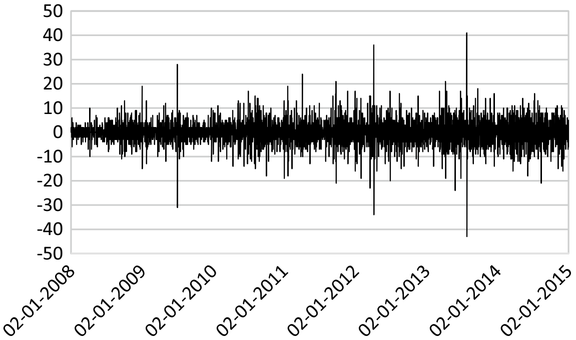


Table 2 The structure of learning data

NN input				NN output
Y(0)	Y(1)	…	Y(k)	Y(k + 1)
Y(1)	Y(2)	…	Y(k + 1)	Y(k + 2)
…	…	…	…	…
Y(i)	Y(i + 1)	…	Y(k + i)	Y(k + i+1)

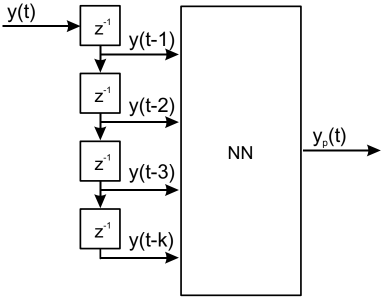


Fig. 4 Model structure

Data for NN

The structure of the input data fed to the neural network is explained in Table 2. The input of the neural network is a vector of k + 1 consecutive values from the time series (Fig. 4). The value of parameter k depends on the implemented model structure.

NN Model of Time Series

Nonlinear prediction of time series can be implemented as a neural network—multilayer perceptron. The problem of neural network design is a number of inputs equal to time series delays and the number of hidden neurons necessary to represent the complexity of a given time series. The number of inputs is suggested by a linear ARIMA model obtained by the autocorrelation and partial autocorrelation functions [8, 16].

Figure 5 presents the three-layered structure of a neural network consisting of inputs, hidden neurons, and an output neuron. Activation function used is hyperbolic tangent. The output neuron is linear [9–11]. The parameters of the neural network model are shown in Table 3.

The total number of parameters (weights) of a three-layer neural network consisting of an input layer, hidden neurons layer, and a one-neuron output layer is described by the formula

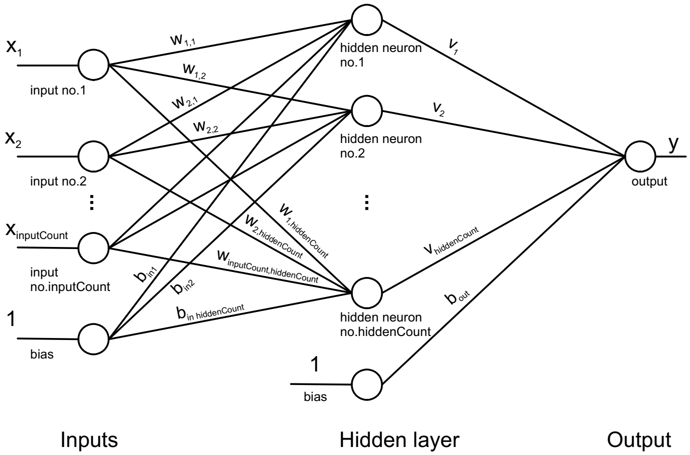


Fig. 5 Neural network structure

Table 3 The parameters of neural network model

Parameter	Count
Input layer weights	No. of inputs
Hidden layer weights	No. of inputs * no. of hidden neurons
Hidden layer bias	No. of hidden neurons
Output layer weights	No. of hidden neurons
Output layer bias	1

$$N_{param} = N_i(1 + N_h) + 2 \cdot N_h + 1 \tag{3}$$

where N_i is the number of inputs and N_h is the number of neurons in the hidden layer.

At the verification stage, the properties of the model residuals are checked, which should have white noise properties. It means that the autocorrelation coefficients of the residues should not differ significantly from zero. For this purpose, the graphs of autocorrelation functions for a series of model residues are analyzed.

The problem of creating neural models of time series is to avoid overfitting, which is the result of a large number of network parameters, i.e., hidden neurons. There are advanced methods of neural networks testing to help solve this problem effectively. The virtual leave–one-out cross-validation test is particularly recommended. The so-called sparseness characterizes the obtained models, i.e., the minimum number of parameters (weights).

Notation NN(*m*–*n*) will be used in following sections to describe network architecture, where *m* denotes the number of inputs and *n* denotes the number of neurons in the hidden layer.

Local Overfitting Control—Virtual Leave-One-Out

A locally linear model can approximate a nonlinear model. In the vicinity of the goal function minimum **w***, a nonlinear model g(**x**, **w***) can be approximated by using the Taylor series expansion:

$$g(\mathbf{x}, \mathbf{w}) = g(\mathbf{x}, \mathbf{w}^{*}) + \mathbf{Z}(\mathbf{w} - \mathbf{w}^{*}) \tag{4}$$

where $\mathbf{x} = [y(i), y(i-1),\ldots, y(i-k), 1]$ is the input vector, *k*—the maximum time delay. **Z**(*N*, *q*) is the Jacobian matrix of the nonlinear neural network of size (*N*, *q*), *N* > *q*, where *N*—the number of training examples, *q*—the number of the neural network weights (Table 3). Each column of **Z** consists of N partial derivatives of the model output with respect to a given weight and describes the influence of one weight update **w** on the model output. Each row of **Z** describes the influence of one learning vector **x** on the model output.

$$z_i = \left(\frac{\partial g(\mathbf{x}, \mathbf{w})}{\partial w_i}\right)_{\mathbf{w}=\mathbf{w}^{*}} \tag{5}$$

$$\mathbf{Z}_{N\times q} = \frac{\partial J}{\partial w_i} = \left(\frac{\partial (y_p - g(\mathbf{x}, \mathbf{w}))^2}{\partial w_i}\right) = -2(y_p - g(\mathbf{x}, \mathbf{w}))\frac{\partial g(\mathbf{x}, \mathbf{w})}{\partial w_i} \tag{6}$$

If the output error:

$$y_p - g(\mathbf{x}, \mathbf{w}) = -\frac{1}{2} \tag{7}$$

then the gradient of the goal function equals the gradient of the network output.

The Jacobian **Z** can be obtained using backpropagation of the error − 1/2 after the network learning is completed. The local linear approximation of the neural network weights has the least-squares form:

$$\mathbf{w}_{LS} = \mathbf{w}^{*} + (\mathbf{Z}^{T}\mathbf{Z})^{-1}\mathbf{Z}^{T}[y_p - g(\mathbf{x}, \mathbf{w}^{*})] \tag{8}$$

By introducing the **H** matrix of the orthogonal projection of the **Z** matrix onto the solution subspace, the effect of withdrawing *k*th example from the training set can be estimated as follows:

$$\begin{aligned} &\mathbf{H} = \mathbf{Z}(\mathbf{Z}^{T}\mathbf{Z})^{-1}\mathbf{Z}^{T} \\ &h_{kk} = \mathbf{z}^{kT}(\mathbf{Z}^{T}\mathbf{Z})^{-1}\mathbf{z}^{k} \\ &\sum_{k=1}^{N} h_{kk} = q, \quad 0 \le h_{kk} \le 1 \end{aligned} \tag{9}$$

The diagonal elements h_{kk} of the **H** matrix are leverages—*k*th components of orthogonal projections. The leverages are equal:

$$h_{kk} = \mathbf{z}^{kT}(\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{z}^k = \sum_{l=1}^{q}\sum_{j=1}^{q} Z_{kl}Z_{kj}(\mathbf{Z}^T\mathbf{Z})^{-1}_{lj} \tag{10}$$

It can be shown that the virtual leave-one-out method can estimate the effect of withdrawing one example of the training set on the network weights [12, 13]:

$$\mathbf{w}_{LS}^{(-k)} = \mathbf{w}^* + (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{z}^k \frac{r_k}{1-h_{kk}} \tag{11}$$

where the residual r_k is

$$r_k = y_p^k - g(\mathbf{x}^k, \mathbf{w}) \tag{12}$$

The leverages can be calculated by the singular value decomposition SVD. The virtual leave-one-out residual equals

$$r_k^{(-k)} = \frac{r_k}{1-h_{kk}} \tag{13}$$

If all leverages were identical then

$$h_{kk} = \frac{q}{N} \tag{14}$$

If $h_{kk} = 0$, then a model is exact for kth example. If $h_{kk} = 1$, then the influence of kth example is extremely high. The leverages can be interpreted as a measure of influence of each training example on the neural network structure. Therefore, our goal is to select a model whose leverages are approximately equal.

Quantitative criteria of model selection [12, 13] are defined as follows. Prediction error of the virtual leave-one-out test E_p is equal:

$$E_p = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\frac{r_k}{1-h_{kk}}\right)^2} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(r_k^{(-k)})^2} \tag{15}$$

This value can be compared to the training mean square error TMSE:

$$TMSE == \sqrt{\frac{1}{N}\sum_{k=1}^{N} r_k^2} \tag{16}$$

The quantity E_p can characterize the models that overfit a subset of training examples. The relations hold

$$\forall k: \quad h_{kk} = \frac{q}{N} \Rightarrow E_p = \frac{N}{N-q} TMSE \tag{17}$$

The leverages distribution can be characterized by the quantity μ:

$$\mu = \frac{1}{N} \sum_{k=1}^{N} \sqrt{\frac{N}{q} h_{kk}} \tag{18}$$

If $\mu = 1$, then $h_{kk} = \frac{q}{N}$ for $k = 1,\ldots, N$; hence this quantity can be used as an index of overfitting if $\mu \leq 1$.

As each neural network is characterized by 2 quantities, the result of model testing can be presented by $E_p - \mu$ plot. Upon the presented methodology, the selected neural network model should be described by the minimum E_p and the maximum (close to 1) μ.

The influence of training examples on the model can also be analyzed by confidence intervals. The confidence interval of the predicted output value if the kth example that is virtually withdrawn is equal:

$$\pm t_{\alpha}^{N-q} s \sqrt{\frac{h_{kk}}{1 - h_{kk}}} \tag{19}$$

where t_{α}^{N-q} is t Student value for *N-q* degrees of freedom, α is the confidence level, s is the standard deviation of training set prediction.

Results

The described method was applied to a real dataset discussed in the section "Analyzed Dataset". The experiments were performed on several subsets of the whole dataset containing 200 samples. The first 100 samples were used as a training subset, and the remaining samples were used for prediction tests. Model learning and prediction tests were performed as one-step-ahead predictor.

In order to design the neural network optimization, the starting point for the neural network structure was the ARIMA model. The source data was preprocessed and differentiated. Previously prepared ARIMA model indicated that no MA component is needed and it will be reasonable to use 3–5 delays.

The set of examined models comprises neural networks with 3–14 inputs and 3–40 hidden neurons. The network training was performed five times for each network architecture. The initial weight values were randomly selected in order to avoid local minima [9, 10]. Synthetic results of model quality assessment are shown as in the $E_p - \mu$ plot for selected neural networks (Fig. 6). The preferred models are in the proximity of E_p equal to 0.8 and μ equal to 0.94.

The detailed results are shown for two models:

- the model was shown a bad generalization property comprising 15 hidden neurons and 4 inputs—called NN (4–15).

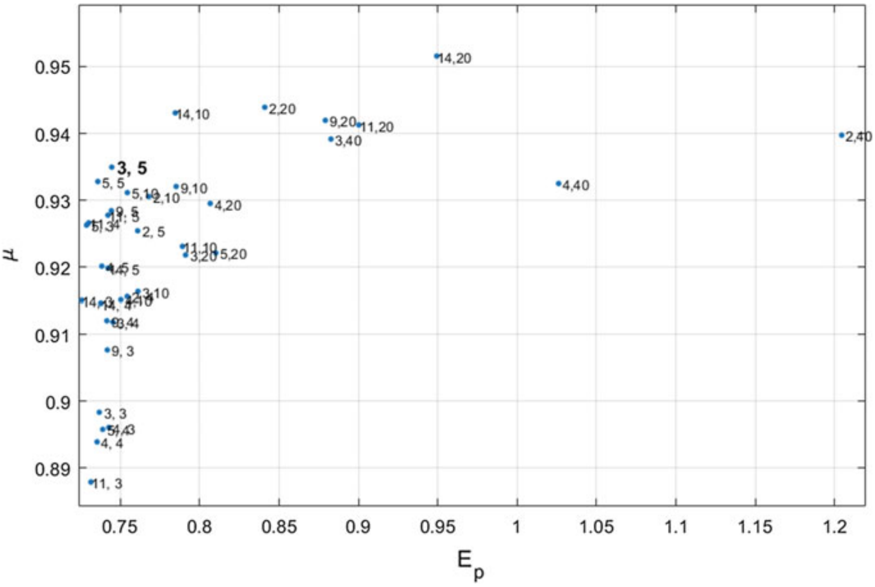


Fig. 6 Prediction error of the virtual leave-one-out test E_p versus quantity μ—the E_p–μ plot. Each point represents one neural network, the numbers in brackets denote, respectively, the number of inputs, and the number of hidden neurons

- for comparison, the model best matching the VLOO criteria comprising five hidden neurons and three inputs—called NN (3–5),

Synthetic Illustrations for the NN (4–15) Model According to the VLOO Criteria

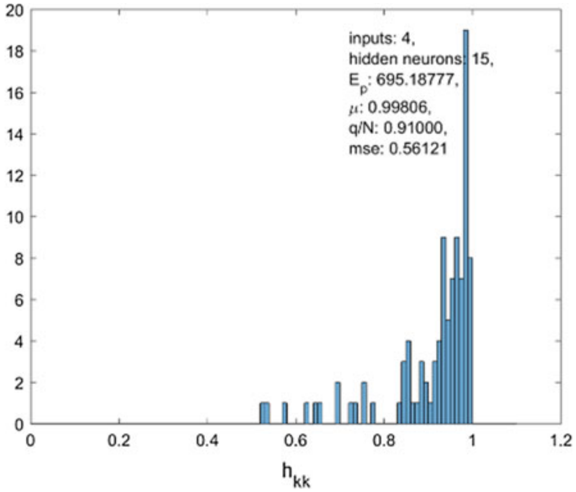
Figure 7 shows the leverage histogram for the NN (4–15) model. The distribution of leverages is concentrated around value 1, so the model is too sensitive to a fraction of the training data—it is overfitted.

Prediction results of the NN (4–15) model are shown in Fig. 9. The green dots indicate the influential points (leverage values greater than 0.9).

The autocorrelation function of prediction error (Fig. 11) for the NN (4–15) model shows that white noise criterion is not satisfied, because some lag values exceed confidence interval.

The confidence interval (Fig. 13) width for the time series prediction on learning set indicates that NN (4–15) model reliability is not very high.

Fig. 7 Leverage histogram for the NN (4–15) model



Synthetic Illustrations for the NN (3–5) Model According to VLOO Criteria

Figure 8 shows the leverage histogram for the NN (3–5) model. The distribution of leverages is concentrated around value close to 0, so the model is not sensitive to a fixed fraction of the training data—the overfitting does not occur.

Prediction results of the NN (3–5) model are shown in Fig. 10. No influential points are present.

The autocorrelation function of the prediction error (Fig. 12) for the NN (3–5) model shows that the white noise criterion is satisfied because all lag values are within the confidence interval.

According to (19), the confidence interval for the time series prediction on the learning set indicate that the selected NN (3–5) model is more reliable than NN (4–15) model due to more uniform leverage distribution.

The sliding window prediction is summarized in Tables 4 and 5. The average MSE value of the NN (3–5) model on the test set equals 1.303 and is lower than the corresponding value of the NN (4–15) model; hence, the generalization ability of the NN (3–5) model is better (despite the similar MSE values on the training set).

Conclusions

Criminological data describe social phenomena difficult for formal modeling. The numerical database allows creating mathematical models of the black box type. It means that there is no cause–effect interpretation—instead the numerical relations between the past and the future of the investigated phenomenon are found.

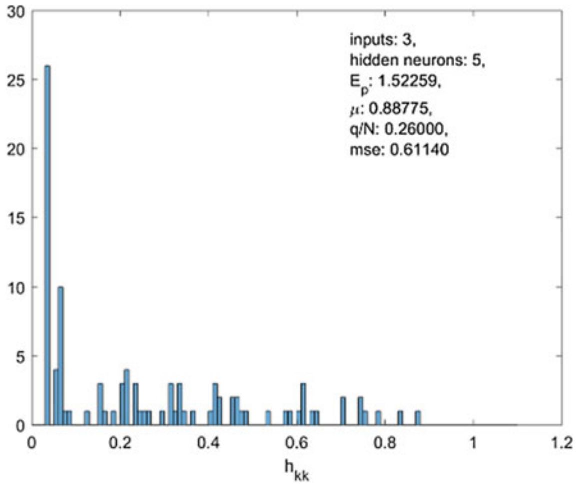


Fig. 8 Leverage histogram for the NN (3–5) model

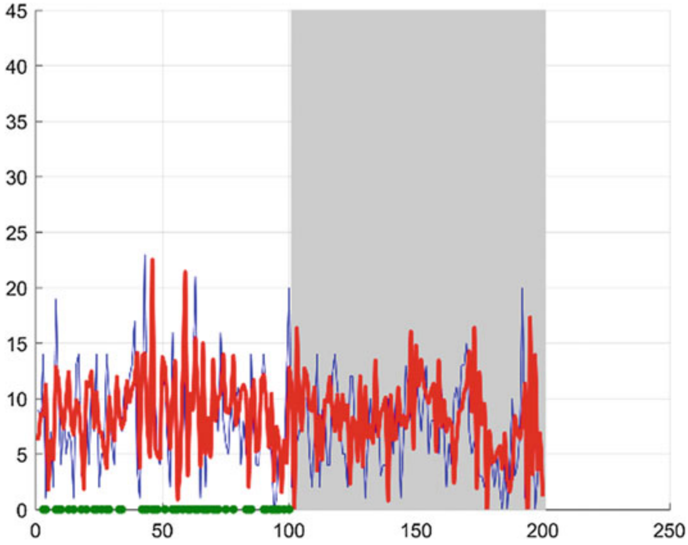


Fig. 9 Prediction results of the NN (4–15) model (blue—original data, red—prediction). Gray background indicates the testing set. White background indicates the learning set

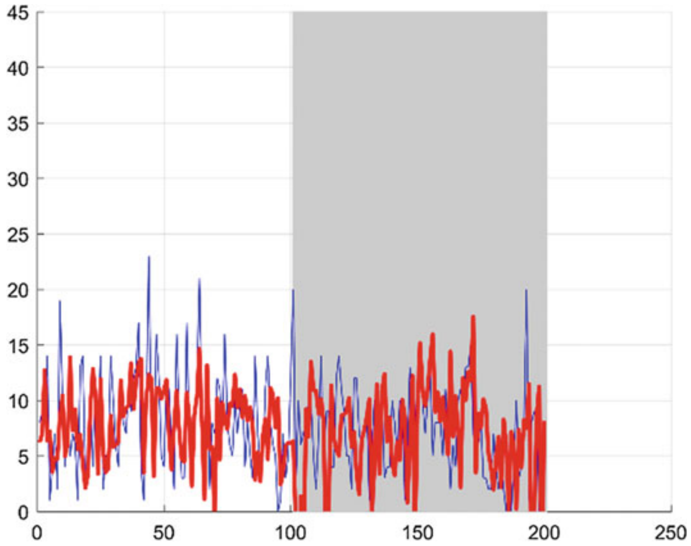
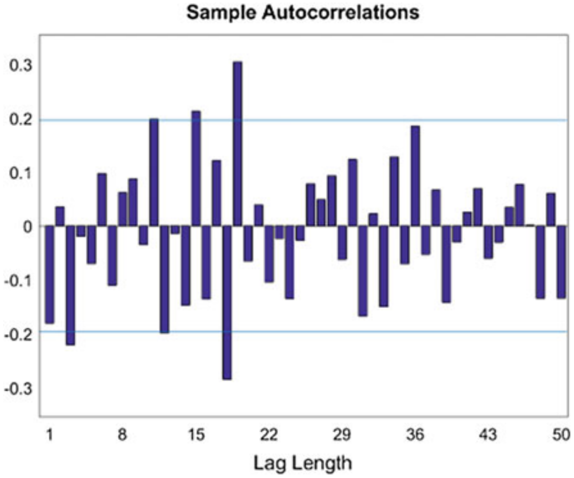


Fig. 10 Prediction results of the NN (3–5) model (blue—original data, red—prediction). Gray background indicates the testing set. White background indicates the learning set

Fig. 11 Autocorrelation function of prediction error for the NN (4–15) model



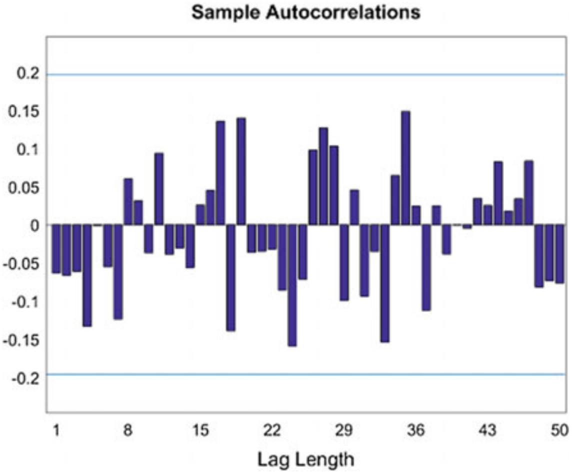


Fig. 12 Autocorrelation function of prediction error for the NN (3–5) model

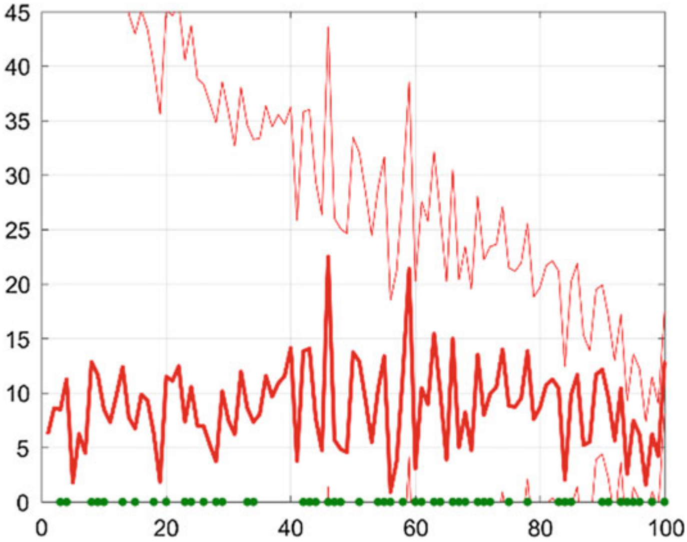


Fig. 13 Confidence interval for the time series prediction by the NN (4–15) model on the training set

Table 4 Mean square prediction error on the training and test sets—NN(4–15) model

Subset	Training set	Test set	E_p	μ
1	0.32025	1.34786	48.25361	0.99839
2	0.71700	1.49739	167.01352	0.99897
3	0.63195	3.16212	183.30357	0.99829
4	0.83721	1.99718	46.85441	0.99899
5	0.57766	0.76377	32.89198	0.99881
6	0.41966	0.58519	47.61199	0.99856
7	0.44554	2.07539	50.38480	0.99842
8	1.14641	4.48117	72.59150	0.99866
9	1.06016	0.75750	84.69553	0.99892
10	0.49907	0.96839	21.35108	0.99886
11	0.62024	1.13369	51.80788	0.99863
12	0.75298	1.16708	30.25622	0.99860
Average	**0.669011**	**1.661394**	**69.75134**	**0.99867**

Table 5 Mean square prediction error on the training and test sets—NN(3–5) model

Subset	Training set	Test set	E_p	μ
1	0.37799	1.1517	0.86410	0.93742
2	0.56507	1.37707	1.24035	0.94234
3	0.62260	1.56501	1.05008	0.95571
4	1.16928	1.49958	1.58339	0.94182
5	0.65316	0.63076	1.15383	0.96405
6	0.51095	0.56752	1.30319	0.96321
7	0.40103	3.06500	3.38401	0.92481
8	0.94954	2.11953	1.31979	0.97204
9	0.63216	0.76378	1.81024	0.94652
10	0.54018	0.75295	1.13184	0.95257
11	0.79046	1.06503	1.00031	0.97868
12	0.79046	1.08028	2.63900	0.95455
Average	**0.666907**	**1.303184**	**1.54001**	**0.95281**

The main goal of the work was to create a nonlinear predictor in the form of a neural network with the structure of a multilayer perceptron. It should be emphasized that the statistical predictions of time series generally give results close to real in the average sense. Statistical models are ambiguous, and their results depend on the structure and numerical details. The selection of the model ensuring optimal generalization is of great practical importance—the prediction quality should remain high even for the data unknown during the creation of the model.

We applied the virtual leave-one-out approach based on an analytical approximation of the effect of withdrawing one example from the training set without withdrawing by using the local linearization of the neural network predictor [12, 13]. This approach enables the general model diagnostics by performing the influential statistics [14]. In general, this approach is attractive in the theory of statistical learning systems as, for example, new semi-supervised classifiers [17].

Application of the virtual leave-one-out procedure allows the selection of neural models with the homogeneous influence of the training examples—thus avoiding the overfitting. This method is computationally efficient and can also be used to dataset diagnostics by discovering the most influential training examples.

The prediction obtained by the selected neural network model is more accurate (low standard error deviation, low level of confidence interval) as can be stated in Tables 4 and 5, and Figs. 7, 8, 9, 10, 11, 12, 13, and 14.

It can be concluded that the virtual leave-one-out test is the efficient tool for the optimal selection of neural network predictors of crime time series in the application for crime prediction performed on the real dataset of spatiotemporal crime incidence.

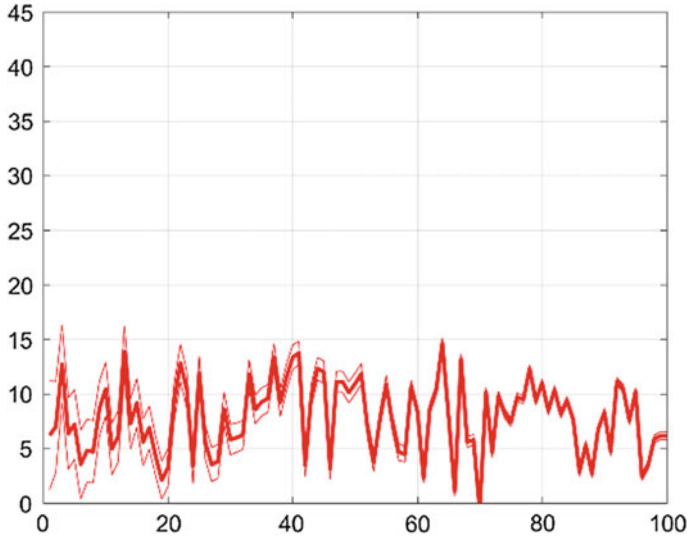


Fig. 14 Confidence interval for the time series prediction by the NN (3–5) model on the training set

Acknowledgements The work was funded by the grant DOB-BIO7/05/02/2015 of Polish National Office for Research and Development.

References

1. Sagl, G., Resch, B., Hawelka, B., Beinat, E.: From social sensor data to collective human behaviour patterns: analysing and visualising spatio-temporal dynamics in urban environments. Proc. GI-Forum 54–63 (2012)
2. Zheng, Y., Capra, L., Wolfson, O., Yang, H.: Urban computing: concepts, methodologies, and applications. ACM Trans. Intell. Syst. Technol. **5**(3), 38 (2014)
3. Eck, J., Chainey, S., Cameron, J., Wilson, R.: Mapping crime: understanding hotspots. Technical report, U.S. Department of Justice (2005)
4. Groff, E.R., La Vigne, N.G.: Forecasting the future of predictive crime mapping. Crime Prev. Stud. **13**, 29–58 (2002)
5. Wang, X., Brown, D.E.: The spatio-temporal modeling for criminal incidents. Secur. Inform. **1**(1), 2 (2012)
6. Ratcliffe, J.H.: A temporal constraint theory to explain opportunity-based spatial offending patterns. J. Res. Crime Delinq. **43**(3), 261–291 (2006)
7. Si, X.S., Wang, W., Hu, C.H., Zhou, D.H.: Remaining useful life estimation—a review on the statistical data driven approaches. Eur. J. Oper. Res. **213**(1), 1–14 (2011)
8. Box, G.E.P, Jenkins, G.M., Reinsel, G.C., Ljung, G.M.: Time Series Analysis: Forecasting and Control. Wiley (2015)
9. Dreyfus, G.: Neural Networks—Methodology and Applications. Springer, Berlin Heidelberg (2005)
10. Bishop, C. M.: Pattern Recognition and Neural Networks, Clarendon Press (2006)
11. Zhang, G.P.: Time series forecasting using hybrid ARIMA and neural network model. Neurocomputing **30**, 159–175 (2003)
12. Monari, G., Dreyfus, G.: Withdrawing an example from the training set: an analytic estimation of its effect on a non-linear parameterised model. Neurocomputing **35**, 195–201 (2000)
13. Monari, G., Dreyfus, G.: Local overfitting control via leverages. Neural Comput. **14**, 1481–1506 (2002)
14. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning—Data Mining, Inference, and Prediction. Springer (2008)
15. Zhang, Y., Yang, Y.: Cross-validation for selecting a model selection procedure. J. Econ. **187**, 95–112 (2015)
16. Woodward, W.A., Gray, H.L., Elliot, A.C.: Applied Time Series Analysis with R. CRC Press, Taylor and Francis Group (2017)
17. Jankowski, S., Szymański, Z., Piątkowska-Janko, E.: Semi-supervised least-squares support vector classifier based on virtual leave one out residuals. In: Proceedings of the 9th International Workshop on Pattern Recognition in Information Systems 2009, pp. 71–80 (2009)

FPGA-Based Echo-State Networks

Erik S. Skibinsky-Gitlin, Miquel L. Alomar, Vincent Canals, Christiam F. Frasser, Eugeni Isern, Fabio Galán-Prado, Alejandro Morán, Miquel Roca and Josep L. Rosselló

Abstract The hardware implementation of Echo State Networks (ESN) can be applied to situations where a quick response is needed in relation to how a certain signal will evolve. This is due to the possibility of connecting the ESN's neurons in parallel, which accelerates the calculation process considerably. In this article, we present a proposal for the compact implementation of ESN in Field Programmable Gate Arrays (FPGAs). To maximize the number of neurons, the synapses are implemented using adders and multiplexers instead of multipliers. The hardware implementation has been tested for the prediction of the Santa Fe time-series dataset. The proposed approach allows for significant savings in terms of energy, hardware resources, and computing time compared to other recently published solutions.

Keywords Echo state networks · FPGA · Time-series forecasting

Introduction

Machine learning applications have exploded in numbers in recent years. This is happening due to the paradigmatic shift from machine learning being it's own purpose, i.e., "machine learning to study how machine learning works" to being applied

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness (MINECO), the Regional European Development Funds (FEDER), and the Comunitat Autnoma de les Illes Balears under grant contracts TEC2014-56244-R, TEC2017-84877-R and a fellowship (FPI/1513/2012) financed by the European Social Fund (ESF) and the Govern de les Illes Balears (Conselleria d'Educació, Cultura i Universitats).

E. S. Skibinsky-Gitlin · M. L. Alomar · V. Canals · C. F. Frasser · E. Isern · F. Galán-Prado · A. Morán · M. Roca · J. L. Rosselló (✉)
Physics Department, Universitat de les Illes Balears, Cra. Valldeomssa km. 7.5, Palma de Mallorca, Balears, Spain
e-mail: j.rossello@uib.es

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_10

to other ends. Its uses and ways of implementation are very wide. This work is in the scope of Recurrent Neural Networks (RNN) and Reservoir Computing (RC) as implementation of the network, and in the field of time-series forecasting as machine learning application.

Numerous applications require the use of specific hardware to implement machine learning applications. Those systems take advantage of the inherent parallelism in the neural processing, (as opposed to typical software machine learning techniques that in the case of neural networks, are a simulation of what otherwise would happen on hardware) that may be beneficial in terms of speed, power, reliability, and cost [6, 22]. As an example, Hardware Neural Networks (HNNs) are necessary for high-volume processing and real-time applications, such as image search and data mining [13, 24].

In HNNs, as in any other neural network, we have two essential parts, the neuron and the axon. The neuron performs a nonlinear operation of its input and the axon adjusts the weight that the output of the neuron will have on the input of the next neuron, or at the output of the system. A neuron has one output (which may fan out to many other neurons via the axons) but several inputs. These inputs are the sum of the values of other neurons multiplied by the weight of each of the axons carrying that value, as shown in Fig. 1.

A great research effort has been made to develop efficient HNN implementations [3, 5, 7, 9, 25] of one-pass incremental—learning of temporal patterns with a bounded memory constraint . However, the synapses' implementation (the gray box in Fig. 1) constrains the viability of implementing massive networks in a single chip. This is because if we want a full parallel neural network to take advantage of designing specific hardware, it is necessary to provide the multiply and sum circuitry for each neuron and these circuits usually grow linearly in silicon area with the neuron's fan-in. This can be catastrophic for large and complex neural networks For these reasons, the use of approximate multipliers [18, 26] has been proposed to reduce hardware at the cost of accuracy loss. Our team has proved that this accuracy loss in axons' weight description will not affect the neural network, at least in the reservoir computing scheme.

Our application for a HNN is to build a reservoir, i.e., a dynamic system which interacts with some input in a complex way. In our case the reservoir is a digital neural network, in particular an Echo State Network (ESN). In order to use the reservoir, an output has to be provided. To this end a machine learning technique is used, as it will be explained in section "Methodology". RC is not limited to neural networks nor digital circuits. There is some research nowadays on optical systems [23], analog electronics [30], memristors [14], and others, as the famous bucket of water [10].

Echo State Networks are suited to time-series prediction and classification tasks [19] and has been successfully applied in numerous domains, such as robot control [4], image/video processing [12], or financial forecasting [17]. Therefore, fast hardware designs implementing ESN systems [1, 2] is of interest for these applications, which require real-time intensive data processing [29].

In this article, we present a proposal for ESN hardware implementation using few hardware resources. The proposed design presents low power characteristics.

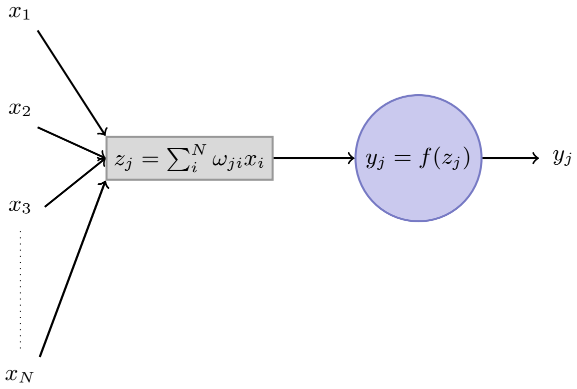


Fig. 1 Typical paradigm of a neuron in any neural network: It receives some inputs from other neurons, it performs a weighed sum of the values (gray box) then performs some nonlinear function (purple circle) and outputs the result to other neurons

As a demonstration of the validity of the approach, we implement a large reservoir network within an FPGA and evaluate its performance for a traditional benchmark on time-series processing (Santa Fe prediction task). The results are compared with some previously published works.

Methodology

Echo State Networks

ESN differs from other recurrent neural networks in that the synaptic weights between neurons are kept fixed and only the connections from the network to a measurement output layer are modified by learning. This reduces the training to a classical linear regression problem. The architecture of a reservoir computing system consists of a total of N internal processing nodes (the neurons) each one providing a given value $x_{k,i}$, where $i \in \{1, 2, .., N\}$ is the neuron index, k represents the evolution during time ($k \in \{1, 2, ..., L\}$) and L is the total number of samples taken from the reservoir. Therefore, the time evolution of internal nodes of the reservoir is described by a matrix with L rows and N columns $\mathbf{X}$ (the design matrix). The state of the network at a given time k is defined by the kth row of the design matrix $\mathbf{x}(k)$ and the time evolution of a given node is stored in the ith column $\mathbf{x}_i$. The output response of the reservoir is computed in two phases. First, the current reservoir state [$\mathbf{x}(k)$] is updated according to a nonlinear function of the weighted sum of the neuron inputs [M external time-dependent inputs $\mathbf{u}(k) = (u_1(k), u_2(k), ..., u_M(k))$], and N internal ones coming from the reservoir's neurons evaluated in the previous time step, $\mathbf{x}(k-1)$ following the expression:

$$\mathbf{x}(k) = f\left[\mathbf{W}_{in}\mathbf{u}(k) + \mathbf{W}\mathbf{x}(k-1)\right] \tag{1}$$

where f is the activation function $f : \mathbb{R}^N \to \mathbb{R}^N$, $\mathbf{W}_{in}$ and $\mathbf{W}$ are two $N \times M$ and $N \times N$ weight matrices, respectively. The network is used to evaluate a total of Q outputs $[\hat{\mathbf{y}}(k) = \left(\hat{y}_1(k), \hat{y}_2(k), ..., \hat{y}_Q(k)\right)]$ that are obtained preforming linear combinations of the reservoir states.

$$\hat{\mathbf{y}}(k) = \mathbf{x}^T(k)\mathbf{W}_{out} \tag{2}$$

where $\mathbf{W}_{out}$ is a $N \times Q$ weight matrix obtained using a linear regression with respect to the expected outputs $[\mathbf{y}(k) = \left(y_1(k), y_2(k), ..., y_Q(k)\right)]$.

Training Method

Assuming we can take a total of L measurements for $\mathbf{x}(k)$ and $\mathbf{y}(k)$, we define $\mathbf{Y}$ as the feature matrix of $L \times Q$ that will be approximated by the network (composed of L row vectors of Q elements $\mathbf{y}(k)$). Then we have that $\mathbf{W}_{out}$ is estimated using the Moore–Penrose pseudo-inverse:

$$\mathbf{W}_{out} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \tag{3}$$

In the reservoir computing scheme, matrices $\mathbf{W}_{in}$ and $\mathbf{W}$ are taken fixed while $\mathbf{W}_{out}$ is conveniently trained using expression (3) or other similar linear fitting.

Low-Cost Hardware Implementation Echo State Networks

The number of neurons employed in ESNs is usually high (typically between 50 and 1000, although some applications require much larger networks to achieve the desired accuracy [12]), which makes particularly challenging the hardware implementation of these systems. Given the large number of products to be implemented due to the high number of synapses, multipliers expend a significant portion of the integrated circuit resources. We limit the possible weights to integer powers of two and sums of powers of two so that shift registers can be employed instead of multipliers.

For a standard ANN implementation that uses backpropagation as learning algorithm, the constraint on the weights is not desirable since it leads to lower network performance [20]. Nonetheless, we show that for ESN with fixed connections, the proposed approach only implies a minor accuracy loss. For the estimation of the output $[\hat{\mathbf{y}}(k)]$ from expression (2) we use the dedicated embedded multipliers integrated in the FPGA device (DSP blocks). The overall area impact of this computation inside the FPGA is relatively low since the logical elements needed to implement the neural network are not used and the number of multipliers at the output layer is significantly lower than the synapses' multipliers (that would limit the maximum neural fan-in). Regarding the network topology, a simple cyclic architecture presents

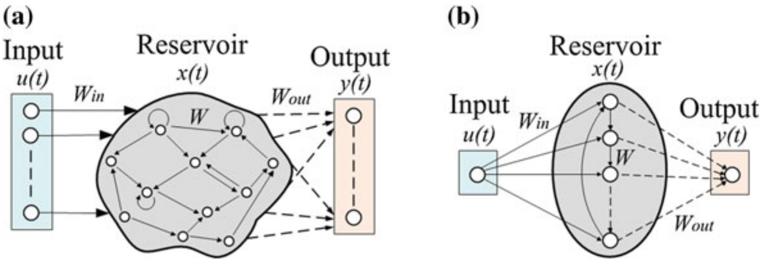


Fig. 2 The general ESN (**a**) where an n-dimensional input vector $u(t)$ is mapped into the reservoir by a weight matrix W_{in} and an m-dimensional output $y(t)$ is mapped out by the machine learning trained matrix W_{out}. In our case (**b**), the input and the output are one-dimensional and the reservoir is ring-shaped, or cyclic

a similar performance to the classical random one while it minimizes the number of connections [28] and optimizes the packing efficiency. In Fig. 2 we show a schematic comparison between a general ESN system and our case of a simple cyclic reservoir (SCR) with one input ($u(t)$) and one output ($Q = 1$). For simplicity, the connections between internal units have the same weight r whereas the inputs are connected to the reservoir with a weight that is positive $v =| v |$ or negative $v = - | v |$ with the same probability and same absolute value. To improve the experimental results, parameters r and v are first adjusted numerically to find the optimum weight configuration.

Figure 3a illustrates a general circuit design for a two-input sigmoid neuron necessary to build the cyclic reservoir when only one input signal is processed. The fixed-point two's complement notation is assumed for all signals so that both positive and negative values can be represented. The first neuron's input $[u(t)]$ refers to the external input signal (to be processed by the network) and the second one $[x_{i-1}(t-1)]$ to the state of a neighboring neuron evaluated at the previous time step. A resolution of n bits is considered for the input and of m bits for the weights (v and r). The multiplier's output is truncated to n bits taking the most significant of the result, but a higher or lower resolution could be employed depending on the desired accuracy.

A simple piece-wise linear approximation with three segments [7] is used for the implementation of the activation function, due to its simple implementation. More accurate designs (e.g., [5, 25]) could be employed to improve the network's performance at the cost of higher hardware requirements. The scheme of Fig. 3a can be simplified to that of Fig. 3b when the weight resolution is limited to a few bits, more specifically $m = 4$. In this case, the full multipliers can be substituted by shift-and-add blocks. Such "multiplier-less" approach enables great hardware saving at the cost of constraining the possible values of the connection weights. The shift-and-add block is depicted in Fig. 3c. Basically, it performs a multiplication of the input signal $[u(t)]$ by the corresponding weight (v) with a pair of shift registers and an adder. Some additional circuitry is included to perform the negation of the shifted values in case it is necessary. A multiplexer is employed to provide either the number that

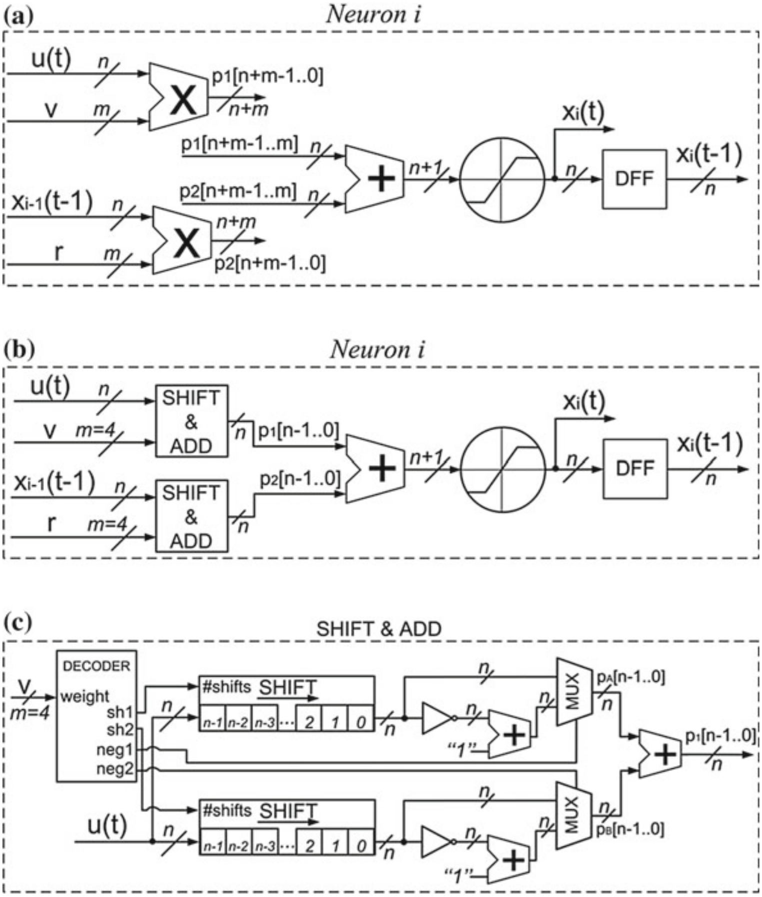


Fig. 3 **a**–**c** Neuron design: **a** general circuit design of the neuron; **b** reduced implementation scheme when the weight resolution is limited to few bits ($m = 4$) and the multipliers are replaced by simple shift-and-add blocks; **c** description of the shift-and-add block

directly results from the shift register or its corresponding negative value depending on a selection signal. A decoder configures the shift registers (with the number of required shifts, $sh1$ and $sh2$) and controls the activation of the negations ($neg1$ and $neg2$) as a function of the weight value (v). By way of example, a single right shift of the input ($sh1 = 1$) performs multiplication by 0.5 while two shifts ($sh2 = 2$) are equal to a factor of 0.25. The direct addition (with $neg1 = neg2 = 0$, indicating that no negation of the shifted values is necessary) of these two shifted magnitudes results in weight $v = 0.75$. The weight value $v = 0.875$ can be implemented by selecting no shifts and no negation for the first shift register ($sh1 = 0$, $neg1 = 0$) and three shifts with a negated output for the second one ($sh2 = 3, neg2 = 1$) so that the input signal $u(t)$ is weighted by the factor $v = 1 - 0.125 = 0.875$. A negative

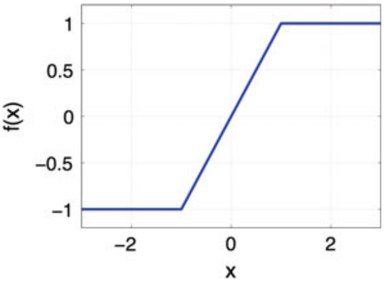


Fig. 4 Piecewise function used as nonlinear activation function

factor, for instance $v = -0.5$, may be obtained through the negation of both shifted magnitudes ($neg1 = neg2 = 1$), where each one is obtained with two displacements ($sh1 = sh2 = 2$) so that $v = -0.25 + (-0.25) = -0.5$. In the case it is desired a generic design, the circuit of Fig. 3c can be used. For FPGA implementations in which the training of r and v are done off-line, an ad hoc design implying the implementation of the exact shifts for each neuron input can be used, further simplifying the hardware and increasing the processing speed. The transfer function f in Eq. (1) determines the behavior of the analog (discrete-time) neuron. For the special case of this work, in which the training is performed at the output layer and not implementing a backpropagation algorithm, a piecewise linear function is able to provide very good fitting results along with a compact hardware implementation. In Fig. 4 we show the nonlinear function used for each internal neuron of the reservoir. The function is easily reproduced in hardware using a few gates and a multiplexer.

Benchmark Prediction Task

In time-series prediction or forecasting the objective is to predict future values based on previously observed ones. Thus, the input sequences are mapped onto a real-valued output sequence that represents one-step or several-step ahead predictions of the desired variable. That is, the value of the series at the current time is introduced each time step as input to the system and the time-series value corresponding to the next (or several) time step must be predicted. In this work, we test the proposed ESN hardware with respect to a widely used benchmark that is the Santa Fe [31] time-series prediction task. The time-series processing is divided into two steps:

– Part of the time series is used for off-line training using an R script. The optimum v and r values are selected along with the output weights ($\mathbf{W}_{out}$). Then, the network is automatically generated using a Hardware Description Language Code (VHDL).
– The rest of the time series is digitized to 16bits two's complement that is transferred to the on-chip RAM memory of the FPGA for its processing.

The typical processing task is to predict the next sample of the time series before it has been injected into the reservoir computer (one-step ahead prediction).

The performance of this task is evaluated using the Normalized Mean Square Error (NMSE):

$$NMSE = \frac{\sum_{i=1}^{L} \left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{L} \left(y_i - \bar{y}\right)^2} \tag{4}$$

where $\mathbf{y} = \big(y(1), y(2), .., y(L)\big)$ is the time series to be predicted (target), $\hat{\mathbf{y}} = \big(\hat{y}(1), \hat{y}(2), .., \hat{y}(L)\big)$ is the predicted value provided by the output layer of the reservoir in the following Eq. (2), and $\bar{y}$ is the mean value of $\mathbf{y}$. Parameter L is the number of samples.

The VHDL code is automatically generated from an R script and is composed of three parts: RAM memory (containing the input to be processed), the cyclic reservoir (constructed using the programmable logic elements of the FPGA), and the output layer (using the dedicated multipliers of the FPGA). The code is used to configure an ALTERA Cyclone IV (EP4CE22F17C6N) FPGA chip.

The Santa Fe prediction task A representative example is the Santa Fe laser time-series prediction task, a widely used benchmark [28]. The task consists in forecasting an experimental recording of the output power of a far-infrared laser operating in the chaotic regime. It is usually evaluated for one-step-ahead predictions. Data is available at [31]. In this work, we employ 4000 samples of the original laser dataset, the first 75% for training and the remaining 25% for testing.

The goal for the Santa Fe task is to predict the next sample in the chaotic time trace before it has been injected into the reservoir computer (one-step-ahead prediction).

Results

We compare the performance of the proposed model with respect to different widely used benchmark tasks and comparing with some previously published works. Regarding the area impact of the proposed methodology, it represents an 86% of area reduction if compared with the standard digital realization when two inputs for each neuron are used and therefore the number of synapses' multipliers are reduced to the minimum in the conventional digital approach.

The proposed methodology is used to synthesize echo state networks with cyclic topology (SCR) (Fig. 2), encoded using VHDL on an ALTERA Cyclone IV FPGA. The performance of the system is tested for the Santa Fe time-series prediction task [31]. Networks with different sizes with up to 200 neurons are implemented and analyzed using the proposed "multiplier-less" approach (Fig. 3b) with a precision of 16 bits ($n = 16$). The training is performed following the methodology mentioned in section "Echo State Networks". A convenient numerical model of the hardware reservoir is employed for the learning phase. Finally, once the prediction error has been scanned for all the possible network configurations (values of r and v), the hardware realization is set up with the optimum weights and evaluated using the test set. The parameters providing better fitting ($r = 0.875$ and $v = 1$) are obtained and

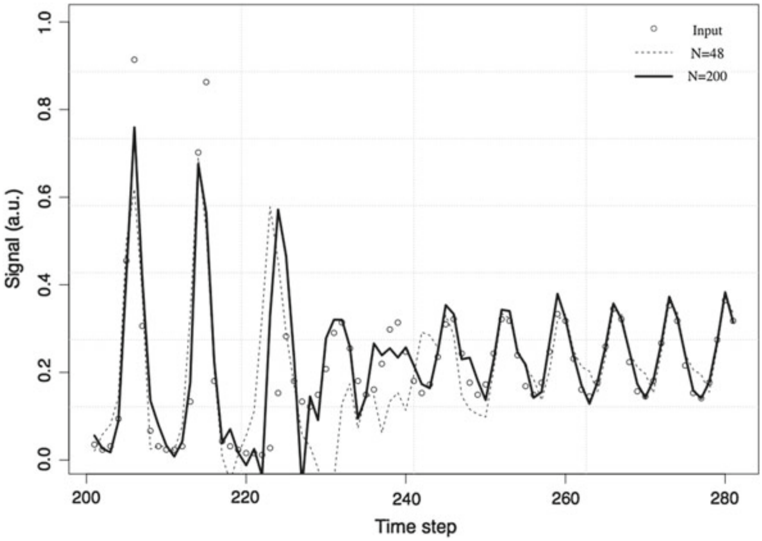


Fig. 5 Fragment of the Santa Fe time-series test set: original values and one-step ahead predictions performed by the proposed reservoir implementation with $N = 48$ and $N = 200$ neurons. As can be appreciated, the network is able to adapt to the abrupt changes of the input

the FPGA is configured with those values. The test set is then stored into the internal RAM memory, thus providing a new input value to the reservoir every time step (each N clock cycles). The resulting outputs (individual neuron states) are processed by the FPGA providing the output $\hat{\mathbf{y}}$ each N clock cycles. This computation is performed following Eq. (2) in a total of N clock cycles so that the processing speed of the proposed design is f/N, where f is the clock frequency that for all the experiments performed in this work is fixed to 50 MHz. This output is extracted from the FPGA with a logic analyzer and used to calculate the system's performance as the error between the estimated and targeted values. For this task we employ a total of 4000 samples of the original laser dataset, the first 3000 for training and the remaining 1000 for testing.

Figure 5 illustrates the experimental predictions obtained through the proposed design when using 48 and 200 neurons. As can be observed, the fitting of the measurements taken in the FPGA is improved when increassing N.

In Table 1 we show the performance of the proposed model measured in terms of the NMSE, speed (in points predicted per second) and power dissipation for the processing of the Santa Fe time-series prediction task. Comparison with previously published models is also included in the table. We distinguish between theoretical values of NMSE obtained from high-precision numerical calculations (Soft.) and NMSE values obtained from experimental settings (Hard.). These two values can differ significantly due to the intrinsic complexity of experimental settings that may present both system and quantization noise. This is evidenced in references [8, 11],

Table 1 Performance results of the Santa Fe experiments for the proposed design and some previously published models

Technology	References	N	NMSE (Hard.)	NMSE (Simulation)	Speed (pps)	Power (W)	PDP ($W \cdot \mu s$)
Optoelectronic	[23]	200	–	0.02	–	–	–
Optoelectronic	[8]	388	0.106	–	$1.3 \cdot 10^7$	150	11.5
Optoelectronic	[11]	400	–	0.021	–	–	–
Numerical	[28]	200	–	0.008	–	–	–
Numerical	[28]	50	–	0.018	–	–	–
Analog circuit	[30]	400	0.031	–	–	–	–
FPGA	[2]	50	0.131	–	1142	0.083	72.6
FPGA	[1](16b)	50	0.075	–	763	<1.5	1966
FPGA	[1](12b)	50	0.12	–	12207	<1.5	123
FPGA	This work	200	0.079	0.0766	$2.5 \cdot 10^5$	<1.5	6
FPGA	This work	48	0.148	0.144	10^6	<1.5	1.5

where the expected NMSE provided by software is considerably lower than the measured one. We also show the results of different studies that are purely numerical as the work in [28] showing the expected performance of Simple Cycle Reservoir designs or the paper in [23] where a semiconductor ring laser with optical feedback (SRL) is numerically simulated. We also provide in the table the Power-Delay Product (PDP) achieved by the experimental settings (a classical figure of merit of the overall hardware performance). As can be appreciated the proposed design is able to provide a factor of two of lower PDP when compared with [8, 11].

Conclusions

In this work, we have presented a digital implementation of ESN. It is shown that the connection weights can be limited to a few discrete values without compromising the system's performance. The validity of the resulting implementation has been demonstrated for the Santa Fe time-series prediction task. Performance comparisons with previous works are shown showing that the proposed model present competitive results. Those comparisons show that the proposed model represent a considerable improvement in terms of speed and power dissipation while is able to provide a similar accuracy than previous models. The proposed solution allows the parallel computation of all the recurrent network with a processing speed of the order of MHz and therefore it is suited to supporting real-time signal processing applications. On the other hand, the proposed approach can be useful to perform specialized systems implementing computational intelligence techniques that requires low power consumption. Potential applications include speech recognition [15], robotics [4], wireless sensor networks [21], predictive controllers [16], and the classification of medical signals [27] among others. To summarize, it has been shown that the use of low-resolution weights to implement the internal synapses of the neurons has little effect on the system's performance while it allows a considerable reduction of the hardware since the use of binary multipliers are avoided. This observation makes possible a very compact implementation of massive reservoir networks with parallel processing capabilities.

References

1. Alomar, M.L., Canals, V., Perez-Mora, N., Martínez-Moll, V., Rosselló, J.L.: FPGA-based stochastic echo state networks for time-series forecasting. Comput. Intell. Neurosc. **2016** (2016)
2. Alomar, M.L., Soriano, M.C., Escalona-Morán, M., Canals, V., Fischer, I., Mirasso, C.R., Rosselló, J.L.: Digital implementation of a single dynamical node reservoir computer. IEEE Trans. Circuits Syst. II: Express Briefs **62**(10), 977–981 (2015)
3. Ando, K., Yamauchi, K.: One-pass incremental -Learning of temporal patterns with a bounded memory constraint. In: Proceedings of the International Conference on Time Series and Forecasting, pp. 1253–1264 (2018)
4. Antonelo, E.A., Schrauwen, B.: On learning navigation behaviors for small mobile robots with reservoir computing architectures. IEEE Trans. Neural Netw. Learn. Syst. **26**(4), 763–780 (2015)
5. Baptista, D., Morgado-Dias, F.: Low-resource hardware implementation of the hyperbolic tangent for artificial neural networks. Neural Comput. Appl. **23**(3–4), 601–607 (2013)
6. Baptista, F.D., Morgado-Dias, F.: Automatic general-purpose neural hardware generator. Neural Comput. Appl. **28**(1), 25–36 (2017)
7. Basterretxea, K., Tarela, J.M., del Campo, I.: Digital design of sigmoid approximator for artificial neural networks. Electron. Lett. **38**(1), 35–37 (2002)
8. Brunner, D., Soriano, M.C., Mirasso, C.R., Fischer, I.: Parallel photonic information processing at gigabyte per second data rates using transient states. Nature Commun. **4** (2013)
9. Carrasco-Robles, M., Serrano, L.: Accurate differential tanh(nx) implementation. Int. J. Circuit Theory Appl. **37**(5), 613–629 (2009)
10. Fernando, C., Sojakka, S.: Pattern recognition in a bucket. In: Banzhaf, W., Ziegler, J., Christaller, T., Dittrich, P., Kim, J.T. (eds.) Advances in Artificial Life, pp. 588–597. Springer, Berlin, Heidelberg (2003)
11. Hicke, K., Escalona-Morán, M., Brunner, D., Soriano, M.C., Fischer, I., Mirasso, C.R.: Information processing using transient dynamics of semiconductor lasers subject to delayed feedback. IEEE J. Sel. Topics Quantum Electron. **19**(4), 1501610–1501610 (2013)
12. Jalalvand, A., Wallendael, G.V., Walle, R.V.D.: Real-time reservoir computing network-based systems for detection tasks on visual contents. In: Proceedings—7th International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2015, pp. 146–151 (2015)
13. Krizhevsky, A., Sutskever, I., Geoffrey E., H.: ImageNet classification with deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **25** (NIPS2012), 1–9 (2012)
14. Kulkarni, M.S., Teuscher, C.: Memristor-based reservoir computing. In: 2012 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), pp. 226–232 (July 2012). https://doi.org/10.1145/2765491.2765531
15. Lee, M., Hwang, K., Park, J., Choi, S., Shin, S., Sung, W.: FPGA-based low-power speech recognition with recurrent neural networks. In: IEEE Workshop on Signal Processing Systems, SiPS: Design and Implementation, pp. 230–235 (2016)

16. Li, H., Zhang, D., Foo, S.Y.: A stochastic digital implementation of a neural network controller for small wind turbine systems. IEEE Trans. Power Electron. **21**(5), 1502–1507 (2006)
17. Lin, X., Yang, Z., Song, Y.: Short-term stock price prediction based on echo state networks. Expert Syst. Appl. **36**(3 PART 2), 7313–7317 (2009)
18. Lotrič, U., Bulić, P.: Applicability of approximate multipliers in hardware neural networks. Neurocomputing **96**, 57–65 (2012)
19. Lukoševičius, M., Jaeger, H., Schrauwen, B.: Reservoir computing trends. KI - Künstliche Intelligenz **26**(4), 365–371 (2012)
20. Marchesi, M., Orlandi, G., Piazza, F., Uncini, A.: Fast neural networks without multipliers. IEEE Trans. Neural Netw. **4**(1), 53–62 (1993)
21. Mathews, E., Poigné, A.: An Echo State Network based pedestrian counting system using wireless sensor networks. In: 2008 International Workshop on Intelligent Solutions in Embedded Systems (WISES 2008), pp. 1–14 (2008)
22. Misra, J., Saha, I.: Artificial neural networks in hardware: a survey of two decades of progress. Neurocomputing **74**(1–3), 239–255 (2010)
23. Modeste Nguimdo, R., Verschaffelt, G., Danckaert, J., Van Der Sande, G.: Simultaneous computation of two independent tasks using reservoir computing based on a single photonic nonlinear node with optical feedback. IEEE Trans. Neural Netw. Learn. Syst. **26**(12), 3301–3307 (2015)
24. Morro, A., Canals, V., Oliver, A., Alomar, M.L., Galan-Prado, F., Ballester, P.J., Rossello, J.L.: A Stochastic Spiking Neural Network for Virtual Screening (2017)
25. Nascimento, I., Jardim, R., Morgado-Dias, F.: A new solution to the hyperbolic tangent implementation in hardware: Polynomial modeling of the fractional exponential part. Neural Comput. Appl. **23**(2), 363–369 (2013)
26. Nedjah, N., De MacEdo Mourelle, L.: Reconfigurable hardware for neural networks: Binary versus stochastic. Neural Comput. Appl. **16**(3), 249–255 (2007)
27. Raghunathan, S., Gupta, S.K., Ward, M.P., Worth, R.M., Roy, K., Irazoqui, P.P.: The design and hardware implementation of a low-power real-time seizure detection algorithm. J. Neural Eng. **6**(5), 056005 (2009)
28. Rodan, A., Tiño, P.: Minimum complexity echo state network. IEEE Trans. Neural Netw. **22**(1), 131–144 (2011)
29. Rossello, J.L., Alomar, M.L., Morro, A., Oliver, A., Canals, V.: High-density liquid-state machine circuitry for time-series forecasting. Int. J. Neural Syst. **26**(5), 1550036 (2016)
30. Soriano, M.C., Ortín, S., Keuninckx, L., Appeltant, L., Danckaert, J., Pesquera, L., van der Sande, G.: Delay-based reservoir computing: noise effects in a combined analog and digital implementation. IEEE Trans. Neural Netw. Learn. Syst. **26**(2), 388–393 (2015). https://doi.org/10.1109/TNNLS.2014.2311855
31. Weigend, A.S., Gershenfeld, N.A.: Results of the time series prediction competition at the Santa Fe Institute. In: IEEE International Conference on Neural Networks - Conference Proceedings. vol. 1993-Janua, pp. 1786–1793 (1993)

Econometric Models, Financial Forecasting and Risk Analysis

Conditional Heteroskedasticity in Long-Memory Model “FIMACH” for Return Volatilities in Equity Markets

A. M. M. Shahiduzzaman Quoreshi and Sabur Mollah

Abstract This paper incorporates conditional heteroskedasticity properties in the long-memory model and applies the model on squared returns of BRICS (Brazil, Russia, India, China, and South Africa) and the United States equity markets to capture the volatility of the stock return. The conditional first- and second-order moments are provided. The CLS, FGLS, and QML are discussed and 2SQML estimator is proposed employing a nonstationary mean function. The simulation study suggests that the proposed 2SQML estimator performs better than the other three estimators. Both in simulation and empirical studies, we find that the proposed model FIMACH outperforms FIGARCH in terms of eliminating serial correlations.

Keywords Long-memory · Conditional heteroskedastic · Return volatility

Introduction

The volatility of stock returns reflects the response to macroeconomic news and rumors. Engle and Patton [1], and Poon and Granger [2] stress that volatility surface has empirically been proved to have persistence for a long time against market shocks. The long-memory phenomenon in time series was first considered by Hurst [3, 4]. In these studies, he explains the long-term storage requirements of the Nile river. He shows that the cumulated water flows in a year depend not only on the water flows in recent years but also on water flows in years much earlier prior to the present year. Mandelbrot and Van Ness [5] explain and advance Hurst’s studies by employing fractional Brownian motion. In analogy with Mandelbrot and Van Ness [5], Granger [6], Granger and Joyeux [7], and Hosking [8] develop Autoregressive Fractionally Integrated Moving Average (ARFIMA) models to account for the long-memory in time series data. However, an empirical study regarding the usefulness of ARFIMA

A. M. M. S. Quoreshi (✉)
Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden
e-mail: shahiduzzaman.quoreshi@bth.se

S. Mollah
Sheffield University Management School, University of Sheffield, Sheffield S10 1FL, UK
e-mail: s.mollah@sheffield.ac.uk

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_11

model is conducted by Bhardwaj and Swanson [9], who find strong evidence in favor of ARFIMA in absolute, squared, and log-squared stock index returns. In this regard, Ding and Granger [10] point out that a number of other processes can also have the long memory property. Further, a Fractionally Integrated Generalized Autoregressive Conditional Heteroskedasticity (FIGARCH) is primarily developed by Baillie et al. [11] but later modified by Chung [12]. Nevertheless, Quoreshi [13] develops an Integer-valued ARFIMA (INARFIMA) model to account for the long memory property in a high-frequency count data framework.

This paper incorporates conditional heteroskedasticity properties in the long-memory model and applies the model on squared returns of BRICS (Brazil, Russia, India, China, and South Africa) and the United States. The new model is called Fractionally Integrated Moving Average Conditional Heteroskedasticity (FIMACH). This model is designed, in a similar fashion to Quoreshi [13], for non-integer data. One important difference in estimation between the introduced model and the model in ARFIMA class is that this model class can study the heteroskedasticity property on the level series while the ARFIMA-FIGARCH class studies the same on the fractionally differenced series through Fourier transformation. One obvious advantage of the FIMACH model over the ARFIMA-FIGARCH class is that the model can easily be extended to multivariate settings for the level series. The model may additionally be used to measure the reaction times for macroeconomic news or rumors, and captures information spread through the system. The model is specified in terms of first- and second-order moments conditioned on historical observations. We argue that the mean function for the estimations of the long-memory parameter be used instead of autocorrelation function even though the mean function may not converge theoretically. Since the time series are always finite in length in the empirical application. we may assume that the impact of lags approaches zero in mean function as the lag length approaches infinity. This motivates to test whether the estimations employing the mean function perform better or not. We perform a Monte Carlo simulation, where we find that ARFIMA or FIGARCH is not suitable for data that are generated according to the FIMACH model. Empirically, we find evidence of long-memory for squared stock return of the United States and BRICS countries. It is also found that the FIMACH model outperforms both FIGARCH and ARFIMA models in terms of eliminating serial correlations.

The paper is organized as follows. The ARFIMA-FIGARCH model class is discussed, and the FIMACH model is introduced in the section “Model”. The estimation procedure of FIMACH is discussed in the section “Estimation”. The section “Monte Carlo Experiment” presents a brief Monte Carlo experiment. The description of the empirical data is presented in the section “Empirical Data”. The empirical results on the stock return volatilities are presented in the section “Empirical Results”, and the concluding comments are included in the section “Conclusion”.

Model

Granger and Joyeux [7] and Hosking [8] introduce the ARFIMA(p, d, q) class of models of the discrete-time real-valued series x_t

$$\alpha(L)(1-L)^d x_t = \beta(L)u_t. \tag{1}$$

The moving average representation of ARFIMA (0, d, 0) of the series x_t is

$$x_t = u_t + d_1 u_{t-1} + d_2 u_{t-2} + d_3 u_{t-3} \ldots$$

or

$$x_t = (1+L)^{-d} u_t. \tag{2}$$

Note that x_t has long memory in a sense that the variable has a slow decaying autocorrelation function and the parameters $d_j = \Gamma(j+d)/[\Gamma(j+1)\Gamma(d)]$, $j = 0, 1, 2, \ldots$ where $d_0 = 1$, and where u_t is a zero-mean serially uncorrelated process. Approximating $d_j \simeq Aj^{-d}$, for $j \geq 1$, Granger and Joyeux [7] propose the following representation of fractionally integrated MA (∞) model

$$y_t = A\sum_{j=1}^{\infty} j^{-d} u_{t-j} + u_t. \tag{3}$$

According to Granger and Jouex [7], the series has the following variance:

$$V(y) = A\sigma_u^2 \sum_{j=1}^{\infty}\left(1 + j^{2(d-1)}\right). \tag{4}$$

From the theory of infinite series, it is known that $\sum_{i=1}^{\infty} j^{-d}$ converges for $d > 1$, otherwise, it diverges. They conclude that the variance of x_t and y_t differ only in finite quantity. Hence the variance for x_t is finite provided $d < \frac{1}{2}$, but infinite if $d \geq \frac{1}{2}$. Geweke and Porter-Hudak [14] show that $\sum_{i=1}^{\infty} d_i < \infty$ if and only if $d < 0$. In a similar fashion of ARFIMA(p, d, q) defined in (1), Baillie et al. [11] introduce Fractionally Integrated Generalized Autoregressive Conditional Heteroskedasticity, FIGARCH (k, d, l) process for u_t^2

$$\varphi(L)(1-L)^d u_t^2 = \alpha_0 + [1-\beta(L)]v_t \tag{5}$$

where all the roots of $\alpha(L)$ and $[1-\beta(L)]$ lie outside the unit circle and $0 < d < 1$. Eq. (5) can be written as

$$[1-\beta(L)]\sigma_t^2 = \alpha_0 + \left[1-\beta(L)-\varphi(L)(1-L)^d\right]u_t^2. \tag{6}$$

The conditional variance for u_t^2 is

$$\begin{aligned}\sigma_t^2 &= \alpha_0[1-\beta(L)]^{-1} + \left\{1-[1-\beta(L)]^{-1} - \varphi(L)(1-L)^d\right\}u_t^2 \\ &\equiv \alpha_0[1-\beta(L)]^{-1} + \theta(L)u_t^2 \end{aligned} \tag{7}$$

where $\theta(L) = \theta_1 L + \theta_2 L^2 + \cdots$. It is assumed that the FIGARCH (k, d, l) process in Eq. (5) be well-defined and the conditional variance be positive almost surely for all t and all $\theta_k(L) \geq 0,\ \text{for}\, k = 1, 2, \ldots$.

We assume that $r_t = p_t - p_{t-1}$ is a stock index return time series, where p_t is the price for the index at time t. If the expected value r_t is zero, we get r_t^2 as the variance at time point t. Let the degree of stock index return volatility, proxied by the squared return r_t^2, have a slow decaying autocorrelation function. Note that standard deviation is widely used as a measure of volatility which is just the square root of the variance. In order to simplify, we further assume that x_t represents r_t^2, stock index return volatility. Here, we assume that the u_t is i.i.d. sequence of random variables with unconditional mean $E(u) = \lambda$ and variance $V(\alpha u) = \alpha^2 \emptyset^2$ where $V(u) = E(u)^2 - \lambda^2 = \emptyset^2$. Conditionally, it holds that $E(u|u) = u$ and $V(\alpha u|u) = \alpha^2 V(u|u)$ where $V(u|u) = u^2 - 2\lambda u + \lambda^2$. Employing these assumptions on x_t, the conditional mean and variance for the Moving Average representation of ARFIMA (0, d, 0) of Granger and Jouex [7] are

$$E(x_t|Y_{t-1}) = E_{t-1} = \lambda + \sum_{i=1}^{m} d_i u_{t-i} \tag{8a}$$

$$V(x_t|Y_{t-1}) = V_{t-1} = \emptyset^2 + \sum_{i=1}^{m} d_i^2\left(u_{t-i}^2 - 2\lambda u_{t-i} + \lambda^2\right) \tag{8b}$$

where Y_{t-1} is the information set available at time $t-1$ and $m = \infty$. The conditional mean and variance vary with u_{t-j}. Hence, this model is clearly different from the models discussed above. Since the conditional variance varies with u_{t-j}, there is a conditional heteroskedasticity property of moving average type that Brännäs and Hall [15] called MACH(q). As λ and $\emptyset^2$ are not functions of time and $\left|\sum_{j=1}^{\infty} d_j\right| \leq \left|\sum_{j=1}^{\infty} d_j^2\right|$ for $d \in [-1, 1]$, it is sufficient that $\sum_{j=1}^{\infty} d_j < \infty$ for x_t to be a stationary sequence. It is worth noting that neither Granger and Joyeux [7] nor Baillie et al. [11] have specified the conditional mean function of the process. The reason could be that the mean of the process is nonstationary and hence may not be appropriate for the estimation procedure. However, assuming $d_i = 0$ for very large m, we get finite mean and variance. We call the model Fractionally Integrated Moving Average Conditional Heteroskedasticity FIMACH (d) where d represents the long-memory parameter. The main difference between FIMACH and FIGARCH is that in FIGARCH, the long-memory property of the variance of the error term u_t is modeled while the conditional expected value for x_t is modeled in FIMACH. The difference is more visible in using lag operators. FIGARCH uses $(1-L)^d$ while $(1+L)^{-d}$ is used in FIMACH and hence those models are different. The variance of FIMACH (d) may be compared to what Baillie et al. [11] have specified in Eq. (7) with $\alpha_0[1-\beta(L)]^{-1}=0$. Both of the properties for x_t given in Eq. (8a), (8b), (8c) can be used for estimation, while the variance for u_t given in Eq. (7) is used for estimation. For $d>0$, the variance of FIGARCH is not covariance-stationary [16]. Employing the same argument as for Eq. (4), it is trivial to show that the conditional variance for FIMACH is finite for $d<\frac{1}{2}$.

If the mean function in Eq. (8a) is well-behaved and decreasing for larger lags, we can use the mean function estimating long-memory parameter instead of using auto-covariance and variance functions as Granger and Joyeux [7] and Baillie et al. [11] have suggested, respectively. Diebold [17] proposes also a test for long memory based on time variance function. The autocorrelation functions of the ARFIMA model class are assumed to be a hyperbolic function while the general mathematical expression of the autocorrelation function for FIMACH is considerably complicated to derive, although possible. Assuming $E(u_t u_t|Y_{t-1})=u_t^2$ and $E\left(u_t u_{t-j}|Y_{t-1}\right)=0$ where $j=1,2,\ldots,\infty$, we can provide a simple form of conditional autocorrelation function at lag k for FIMACH as

$$\rho_{k|t-1}=\frac{\sum_{j=0}^{\infty}d_j d_{k+j}u_{t-j-k}^2}{V\left(\sigma_t^2|Y_{t-1}\right)} \tag{8c}$$

where $k=-j,j$ and represents lag, and $d_0=1$. Note that this autocorrelation function varies with u_{t-j} which captures the heteroskedasticity property in the autocorrelation function. The heteroskedasticity in autocorrelation function for absolute return of stock is illustrated by Ding [10], although the authors assume a smooth function for explaining the autocorrelation. The FIMACH model can be extended with random parameters as

$$x_t=u_t+d_1u_{t-1}+d_2u_{t-2}+d_3u_{t-3}\ldots+\sum_{i=1}^{p}\delta_i u_{t-i} \tag{9}$$

where d_i capture the long-memory properties and have the same definitions as in Eq. (1). The δ_i, $i=1,2,\ldots,$ comprise the random parameters and are independent of each other. These parameters capture the short term deviation from the long-memory trend. We name this model FIMACH(d, p) model. The conditional mean and variance of the random coefficients of FIMACH(d, p) representation can be written as

$$E(x_t|Y_{t-1})=E_{t-1}=\lambda+\sum_{i=1}^{m}d_i u_{t-i}+\sum_{i=1}^{p}\delta_i u_{t-i} \tag{10a}$$

$$V(x_t|Y_{t-1})=V_{t-1}=\emptyset^2+\sum_{i=1}^{m}d_i^2\left(u_{t-i}^2-2\lambda u_{t-i}+\lambda^2\right)$$

$$+\sum\nolimits_{i=1}^{p}\delta_i\left(u_{t-i}^2-2\lambda u_{t-i}+\lambda^2\right)\tag{10b}$$

where Y_{t-1} is the information set available at time $t-1$and $m=\infty$. As for Eq. (8a), (8b), assuming $d_i=0$ for very large m, we get finite mean and variance. Note that the moments are conditioned only on the previous observations, Y_{t-1}. The same stationary condition is applicable as for Eq. (8a), (8b). The model can be used to measure mean and median reaction time to macroeconomic news and rumors[1]. The model can easily be extended to a multivariate setting. Hence, the covariance and Granger-Causality between two or several series can easily be studied in the same fashion as the VARMA model. These possibilities are limited in the FIGARCH of ARFIMA class, at least on the level series.

Estimation

Geweke and Porter-Hudak [14] assume that the spectral density for x_t in Eq. (2), where u_t is a stationary linear process with spectral density $f_u(\lambda)$, is

$$f(\lambda)=\left(\sigma^2/2\pi\right)\left\{4sin^2(\lambda)\right\}^{-d}f_u(\lambda)\tag{11}$$

and

$$\ln\{f(\lambda)\}=ln\left\{\sigma^2 f_u(0)/2\pi\right\}-d\,ln\left\{4sin^2\left(\frac{\lambda}{2}\right)\right\}+\left\{\frac{f_u(\lambda)}{f_u(0)}\right\}.\tag{12}$$

The σ^2 is the variance of x_t and $f_u(\lambda)$ is finite and continuous on $[-\pi,\pi]$. Assuming T as the sample size of x_t and letting $\lambda_{j,T}=\frac{2\pi}{T}$, where $j=0,1,\ldots,T-1$, denote the harmonic ordinates, Eq. (12) can be written as

$$\begin{aligned}\ln\left\{I\left(\lambda_{j,T}\right)\right\}&=ln\left\{\sigma^2 f_u(0)/2\pi\right\}-d\,ln\left\{4sin^2\left(\frac{\lambda_{j,T}}{2}\right)\right\}+\left\{\frac{f_u\left(\lambda_{j,T}\right)}{f_u(0)}\right\}\\&\quad+\ln\left\{I\left(\lambda_{j,T}\right)/f\left(\lambda_{j,T}\right)\right\}\end{aligned}\tag{13}$$

where $I\left(\lambda_{j,T}\right)$ denote the periodogram of these harmonic ordinates. The authors propose the least square estimation of Eq. (13) to estimate d. Baillie et al. [11] propose the following Maximum Likelihood Estimation

$$logL(\emptyset;u_1,u_2,\ldots,u_T)=-0.5*T*\log(2\pi)$$

[1] This is the reaction to macroeconomic news/rumors in the ($\mathrm{u_{jt}}$) sequence, we use the mean lag $\sum_{i=0}^{qj}i\alpha_{ji}/w$, where $w=\sum_{i=0}^{qj}\alpha_{ji}$ and $\alpha_{j0}=1$ (see Quoreshi 2012).

$$-0.5\sum_{t=1}^{T}\left[\log\left(\sigma_1^2\right)+u_1^2\sigma_t^{-2}\right] \quad (14)$$

where σ_t^2 is defined as in Eq. (7) and $\emptyset' \equiv (d, \alpha_0, \beta_1, \ldots, \beta_k, \varphi_1, \ldots, \varphi_l)$. The Maclaurin series expansion of $(1-L)^d$ is used [11].

Quoreshi [13, 18] uses mean function for the estimation of integer-valued long-memory models. Although the mean function of the introduced model may be nonstationary, the function is decreasing and smooth. However, in practical implication, we can never employ an infinite series. Since the time series are always finite in length in the empirical application, we may assume that the impact of large lags is zero in the mean function. This motivates to test whether the estimations employing mean function perform better or not. The forecasting properties of the model can be seen the same as the finite length of time series.

If we do not assume a full density function, we may estimate the Quasi Maximum Likelihood (QML) Estimator as discussed by Weiss [19] and Bollerslev and Wooldridge [20] instead of Maximum Likelihood (ML) Estimator. Conditional Least Square (CLS), Feasible Generalized Least Square (FGLS), Generalized Methods of Moments (GMM), and possibly others, e.g., Two-Stage Least Square (2SLS), are candidates for the estimation. In the previous studies, it turns out that FGLS is the best estimator among the three in terms of eliminating serial correlation [13]. The CLS comes in the second position, which is almost as good as FGLS. Here, we only consider CLS, FGLS, and ML class for estimation.

The Conditional Least Square (CLS) estimator for FIMACH(d, p) representation model have the following residual

$$e_t = y_t - E_{t-1} = \sigma_t^2 - \lambda - \sum_{i=1}^{\infty} d_i u_{t-i} - \sum_{i=1}^{p} \theta_i u_{t-i} \quad (15)$$

and the criterion function $S_{CLS} = \sum_{i=m+1}^{T} e_t^2$ is minimized with respect to unknown parameters, i.e., $\psi = \left(\lambda, \theta' \text{and } d'\right)$ where θ' and d' are vectors of parameters with elements θ_i respective d_i. Using a finite maximum lag m in Eq. (15) instead of infinite lags may cause biasing effects. Due to omitted variables, i.e., $u_{t-m-1}, \ldots, u_{t-\infty}$, we may expect a positive bias on the parameters λ, θ_i and d_i [21]. These moment conditions correspond to the normal equations of the CLS estimator that focuses on the unknown parameters of the conditional mean function. Alternatively and equivalently, the properties $\mathrm{E}(e_t) = 0$ and $\mathrm{E}(e_t e_{t-j}) = 0$, j ≥ 1 could be used. Note that the moment conditions for FIMACH(d, 0) can be obtained by setting $\theta_i = 0$. The FGLS estimator minimizes

$$S_{FGLS} = \sum_{t=m+1}^{T} e_t \widehat{V}^{-1} \quad (16)$$

with $\widehat{V}^{-1}$ as given. The variance of error from CLS estimates may be used for approximation of $\widehat{V}^{-1}$ in Eq. (16). Alternatively, $\widehat{V}^{-1}$ can be estimated as specified in Eq. (8b) by employing estimates from CLS. The covariance matrix estimators for CLS and FGLS are

$$Cov\left(\widehat{\psi}_{CLS}\right) = \left(\sum_{t=m+1}^{T} \frac{\partial e_t}{\partial \psi} \frac{\partial e_t}{\partial \psi'}\right)^{-1}$$

$$Cov\left(\widehat{\psi}_{FGLS}\right) = \left(\sum_{t=m+1}^{T} \widehat{V}^{-1} \frac{\partial e_t}{\partial \psi} \frac{\partial e_t}{\partial \psi'}\right)^{-1}.$$

The ML or QML estimator for FIMACH(d, p) representation model have the same residual as in Eq. (15), and maximize the following criterion function

$$\begin{aligned} L\left(\sigma_1^2, \sigma_2^2 \ldots, \sigma_T^2 | Y_{t-1}, \lambda, \theta_i \text{ and } d_i\right) &= \prod_{t=1}^{T} L\left(\sigma_t^2 | Y_{t-1}, \psi_i\right) \\ &= \left(\frac{1}{2\pi V_{t-1}}\right)^{T/2} exp\left(-\frac{\sum_{i=m+1}^{T} e_t^2}{2V_{t-1}}\right) \end{aligned} \tag{17}$$

where $\psi_i = (\lambda, \theta_i \text{ and } d_i)$ and V_{t-1} are as in Eq. (8b). Taking the logarithm of Eq. (17), we may simply use the criterion function and minimize the function as

$$\begin{aligned} & LnL\left(\sigma_1^2, \sigma_2^2 \ldots, \sigma_T^2 | Y_{t-1}, \lambda, \theta_i, d_i \text{ and } \widehat{V}_{t-1}\right) \\ &= -\frac{T}{2} \ln\left(\widehat{V}_{t-1}\right) - \ln(2\pi) - \left(\frac{\sum_{t=m+1}^{T} e_t^2}{2\widehat{V}_{t-1}}\right) \end{aligned} \tag{18}$$

where $\widehat{V}_{t-1}$ is an estimate for V_{t-1} that is to be estimated. Since T, 2 and π are constants, we can equivalently minimize the following criterion function:

$$LnL\left(\sigma_1^2, \sigma_2^2 \ldots, \sigma_T^2 | Y_{t-1}, \lambda, \theta_i, d_i \text{ and } \widehat{V}_{t-1}\right) = -\ln\left(\widehat{V}_{t-1}\right) - \left(\frac{\sum_{t=m+1}^{T} e_t^2}{\widehat{V}_{t-1}}\right). \tag{19}$$

Note that $\widehat{V}_{t-1}$ is to be estimated at the same time as the other parameters. If the estimation is sensitive to the start value of $\widehat{V}_{t-1}$, we can obviously estimate CLS at the first stage and calculate the $\widehat{V}_{t-1}$ which can be used as the start value for QML. We call this estimation procedure Two-Stage Quasi Maximum Likelihood (2SQML) Estimation. The covariance matrix estimators for QML and 2SQML are

$$Cov\left(\widehat{\psi}_{QML}\right) = \left(\sum_{t=m+1}^{T} \widehat{V}^{-1} \frac{\partial e_t}{\partial \psi} \frac{\partial e_t}{\partial \psi'}\right)^{-1}$$

$$Cov\left(\widehat{\psi}_{2SQML}\right) = \left(\sum_{t=m+1}^{T} \widehat{V}^{-1} \frac{\partial e_t}{\partial \psi} \frac{\partial e_t}{\partial \psi'}\right)^{-1}.$$

One important difference in estimation between this model and the model in the ARFIMA class is that this model can study the heteroskedasticity property on the level series while, for example, FIGARCH of ARFIMA class studies the same on the fractionally differenced series through Fourier transformation.

Monte Carlo Experiment

Smith et al. [22] and Quoreshi [13] have studied the bias and misspecification in ARFIMA respective of INARFIMA models. Drost et al. [23] investigated finite sample behavior of semiparametric integer-valued AR(p) models while Brännäs and Quoreshi [21] studied finite-lag misspecification when the data is generated according to an infinite-lag INARFIMA model. In this brief Monte Carlo experiment, we study the bias, MSE, Ljung–Box statistics, AIC, and SBIC properties of the ML estimators for finite-lag specifications, when data is generated according to FIMACH $(d, 0)$. The data generating process is as in Eq. (1), with $d = 0.1, 0.25$ and 0.4 and lag length $m = 70$. The u_t sequence is generated from i.i.d. normal distribution, with mean 25 and standard deviation 10. Six time series with length $T = 4500$ and $T = 900$ are generated. The first 500 observations are discarded to avoid the start-up effect. The results for the Monte Carlo experiment are given in Table 1. We also generate four other series in a similar fashion, with mean 25 and standard deviation 4 and 100 to study the performance of CLS, FGLS, 2SQML, and QML estimators. We also evaluate the FIMACH$(d, 0)$, ARFIMA $(0, d, 0)$, FIGARCH $(1, 1)$, and GARCH $(1, 1)$ models in terms of eliminating serial correlations when the data are generated in accordance with FIMACH$(d,\ 0)$. These results are presented in Table 2.

We set $\widehat{\lambda}$ equal to mean values for the generated series instead of $\lambda = 25$. By doing so, we eliminate the biased effect of λ in the generated series. Hence, we can study the bias of the estimated parameter d due to misspecification of lag length m more appropriately. The Monte Carlo study shows that as m increases toward $m = 70$, the bias in d decreases (Table 1). For m less than 70, we find positive

Table 1 Bias, MSE, Ljung–Box statistics, AIC and SBIC properties of the ML estimators for finite-lag specifications, when data is generated according to FIMACH (0, d, 0) Model with d = 0.1, 0.25 and 0.4 and m = 70 and $\sigma^2 = 100$

Lag	Parameters	T = 4500			T = 9000		
		0.10	0.25	0.40	0.10	0.25	0.40
M10	δ (s.e.)	0.168*** (0.00)	0.432*** (0.00)	0.698*** (0.00)	0.167*** (0.00)	0.431*** (0.00)	0.696*** (0.00)
	MSE	99.115	104.898	130.848	97.579	103.611	129.173
	LB100	130.669	297.170	1047.228	166.566	525.401	2045.455
	LB200	236.154	417.668	1214.240	285.267	651.793	2200.088
	AIC	20659.704	20907.183	21811.986	41201.319	41723.602	43516.296
	SBIC	20730.210	20977.689	21882.492	41279.461	41801.744	43594.438
M30	δ (s.e.)	0.122*** (0.00)	0.308*** (0.00)	0.496*** (0.00)	0.122*** (0.00)	0.307*** (0.00)	0.495*** (0.00)
	MSE	98.180	99.305	103.427	96.901	97.936	101.736
	LB100	101.855	116.564	167.503	108.976	151.998	284.131
	LB200	202.485	219.121	268.172	225.197	268.950	405.750
	AIC	20566.007	20616.690	20797.669	41048.905	41143.489	41482.381
	SBIC	20596.463	20815.250	20996.228	41127.023	41221.607	41560.499
M50	δ (s.e.)	0.108*** (0.00)	0.271*** (0.00)	0.434*** (0.00)	0.108*** (0.00)	0.270*** (0.00)	0.433*** (0.00)
	MSE	98.177	100.234	104.320	97.025	97.903	100.766
	LB100	101.107	111.853	150.678	105.672	116.088	180.584
	LB200	200.968	217.921	264.321	224.083	234.140	299.255
	AIC	20514.152	20606.353	20784.158	41049.001	41129.531	41387.421
	SBIC	20840.585	20932.787	21110.592	41411.071	41491.601	41749.491
M70	δ (s.e.)	0.101*** (0.00)	0.251*** (0.00)	0.402*** (0.00)	0.100*** (0.00)	0.250*** (0.00)	0.401*** (0.001)
	MSE	98.027	99.939	105.527	96.821	97.604	100.086
	LB100	103.897	115.392	160.502	103.300	108.986	134.357
	LB200	200.740	212.267	264.265	222.201	226.818	249.751
	AIC	20455.626	20541.172	20782.114	40989.067	41060.157	41281.722
	SBIC	20909.753	20995.299	21236.241	41492.967	41564.057	41785.621
M90	δ (s.e.)	0.096*** (0.00)	0.238*** (0.00)	0.380*** (0.00)	0.095*** (0.00)	0.237*** (0.00)	0.379*** (0.00)
	MSE	98.110	100.157	106.060	96.917	97.794	100.522
	LB100	110.666	162.701	332.128	108.179	143.667	292.239
	LB200	210.420	267.350	445.421	228.784	266.039	422.550
	AIC	20407.668	20498.686	20750.943	40936.006	41016.225	41261.143
	SBIC	20989.306	21080.324	21332.581	41581.644	41661.864	41906.782

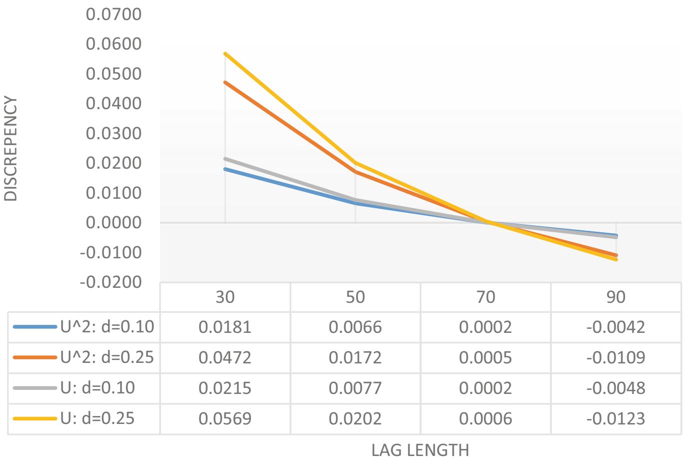
Table 2 Comparison between FIMACH, FIGARCH, and GARCH and between QMLE, 2SQMLE, FGLS, and CLS estimators, when data is generated according to FIMACH (0, d, 0) models with $d = 0.25$ and $m = 70$

		T = 4500				T = 9000			
Lag	Ljung-Box	FIMACH	ARFIMA	FIGARCH	GARCH	FIMACH	ARFIMA	FIGARCH	GARCH
σ = 4	LB100	90.521	138.064	2704.712	2730.476	90.521	138.064	1455.198	1578.196
	LB200	198.883	247.337	2781.968	2805.809	198.883	247.337	1564.312	1687.964
σ = 10	LB100	90.521	138.064	2704.712	2730.476	90.521	138.064	1455.198	1578.196
	LB200	198.883	247.337	2781.968	2805.809	198.883	247.337	1564.312	1687.964
σ = 100	LB100	90.521	138.064	1455.198	1578.196	90.521	138.064	1455.198	1578.196
	LB200	198.883	247.337	1564.312	1687.964	198.883	247.337	1564.312	1687.964
		QMLE	2SQMLE	FGLS	CLS	QMLE	2SQMLE	FGLS	CLS
σ = 4	LB100	679.924	90.521	100.409	100.409	291.590	90.521	291.586	291.592
	LB200	843.340	198.883	200.232	200.232	406.221	198.883	406.217	406.224
σ = 10	LB100	109.710	90.521	100.229	100.229	108.557	90.521	108.557	108.557
	LB200	206.859	198.883	199.617	199.617	226.373	198.883	226.373	226.373
σ = 100	LB100	99.964	90.521	100.139	100.139	102.549	90.521	102.549	102.549
	LB200	198.430	198.883	199.260	199.260	221.728	198.883	221.728	221.728
		σ = 4	σ = 10	σ = 100		σ = 4	σ = 10	σ = 100	
Generated	LB100	1885.1	1885.1	1885.1		1885.1	1885.1	1885.1	
Series	LB200	1991.9	1991.9	1991.9		1991.9	1991.9	1991.9	

bias in d, while the bias is negative for $m = 90$. Biases are smaller for $T = 4500$ when m is less than 70 than those for $T = 9000$. MSE decreases as sample size increases or m increases. Like Quoreshi [13] and Brännäs and Quoreshi [21], we conclude that we may expect a positive biasing effect on the parameters due to omitting variables, i.e. $u_{t-m-1}, \ldots, u_{t-\infty}$. The statistics for AIC and SBIC decrease as lag length increases, and are noted lowest at $m = 90$. Hence, the standard AIC and SBIC need to be corrected in order to choose optimal lag lengths. As expected, the *LB* statistics are, with some exceptions, lowest at $m = 70$. The exceptions may arise due to the conditional heteroskedasticity nature of the dataset. Hence, it is appropriate to evaluate at more than one-time point or to use an average value of *LB* statistics for a number of time points.

When the data is generated according to FIMACH $(d, 0)$, it is not appropriate to use ARFIMA, FIGARCH or GARCH models (Table 2). The ARFIMA model reduces the serial correlation successfully, but it does not perform well like FIMACH. This shows that there is need for using FIMACH instead of ARFIMA when we need to take account of the heteroskedasticity property in the long-memory. FIGARCH and GARCH take account of heteroskedasticity in the short memory. Hence, these models did not perform well. FGLS and CLS perform consistently well, and somewhat better than QML. It turns out that QML is sensitive to start values. Estimating CLS at the first stage and using the CLS estimates as the start value for QML estimator, we estimate 2SQML which performs best out of these estimators.

We have also conducted similar Monte Carlo studies with innovation Uˆ2. The results are quite similar. Average discrepancy (d̂–d) with the number of simulation equals 100 for FIMACH time series, with innovation U^2 respective of U; when the data are generated with lag length 70 and d = 0.1 respective of d = 0.25 are presented in Fig. 1.



	30	50	70	90
U^2: d=0.10	0.0181	0.0066	0.0002	-0.0042
U^2: d=0.25	0.0472	0.0172	0.0005	-0.0109
U: d=0.10	0.0215	0.0077	0.0002	-0.0048
U: d=0.25	0.0569	0.0202	0.0006	-0.0123

Fig. 1 Average discrepancy $(\hat{d} - d)$ with number of simulation equals 100 for FIMACH time series with innovation Uˆ2 respective U when the data are generated with lag length 70 and $d = 0.1$ respective $d = 0.25$

Empirical Data

The daily squared stock MSCI index return series over a period of 17 years from 1995 to 2011 for the Unites States and BRICS are applied in this paper.[2] Each series comprises 4435 observations. The descriptive statistics of the dataset are given in Table 3. The mean squared stock index return for Brazil is about 1800, which is the largest among the BRICS counties. The corresponding number for China is about 1, which is the smallest among the BRICS countries. The corresponding mean for the United States is 172. The skewness, kurtosis, and Jarque–Bera statistics indicate that the data are not from the normal distribution. The squared returns for Brazil and India and the corresponding autocorrelation functions are exhibited in Fig. 2 respective of Fig. 3. The autocorrelation functions for all of the series decay very slowly, which indicates a long-memory behavior in the squared return series.

Empirical Results

The empirical results of the squared stock index returns for the United States and BRICS countries are presented in Table 4a and b. CLS, FGLS, and QML estimators have been employed to evaluate the performance of the estimators. FIMACH (d, 0), FIGARCH (0, d, 0), FIGARCH (1, d, 1), and GARCH (1,1) are estimated to find out the most suitable model for the volatility return. It turns out that CLS, FGLS, and QML estimators have performed equally well for the time series of the United States, Brazil, Russia, and China in terms of reducing serial correlation. However, the QML estimator outperforms CLS and FGLS for the time series of India and South Africa. Hence, we conclude that QML is somewhat a better estimator among these. It is to be noted that the performance of the QML estimator is highly sensitive to the start values, and may turn out worse than that of FGLS or CLS if the start values are not selected carefully. We suggest that the value for the autocorrelation at lag one may be chosen as the start values for the fractional integration parameter. The variance of residuals from the CLS estimator may be used as the start value for $\widehat{V}_{t-1}$ of the QML estimator which we call 2SQML. We also recommend using both QML and CLS estimators in order to determine the best estimates in terms of reducing serial correlation.

We find that the GARCH is not an appropriate model for the time series, as the estimated parameters deviate substantially from the expected values. The FIMACH (0, *d*) turns out to be the best in terms of eliminating serial correlations. This model performs much better than that of FIGARCH (0, d, 0) and FIGARCH (1, d, 1) for all the five time series. For Brazil, the Ljung-Box (LB) statistics for residuals for FIMACH is about 1430, while the corresponding number for FIGARCH is about 16,882 (see Table 4a). The corresponding statistics for the United States are 698 for

[2] Data source: Datastream.

Table 3 This table reports descriptive statistics of squared returns for BRICS (Brazil, Russia, India, China, and South Africa) countries, the United Kingdom and the United States for the period 1995–2011

Country	Mean	Median	Std. Dev.	Minimum	Maximum	Skewness	Kurtosis	Jarque–Bera	Observations
Brazil	1799.718	158.722	6476.492	0.000	130113.100	9.301	125.839	2851677.000***	4435
Russia	216.787	23.981	949.744	0.000	35765.620	19.805	594.868	65009317.000***	4435
India	26.928	2.151	119.300	0.000	4115.351	18.851	542.786	54093083.000***	4435
China	1.032	0.123	4.089	0.000	121.396	14.887	341.988	21393884.000***	4435
South Africa	33.414	4.351	95.426	0.000	1725.239	7.215	78.675	1096490.000***	4435
UK	195.152	46.909	548.819	0.000	11764.870	10.107	152.599	4210143.000***	4435
US	171.966	34.018	472.666	0.000	10158.220	9.380	141.160	3591553.000***	4435

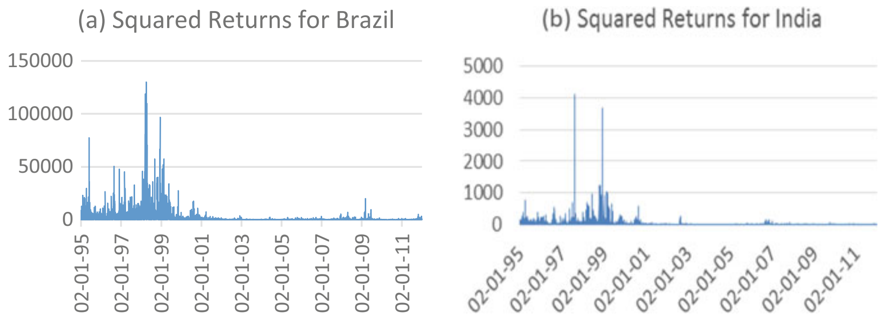


Fig. 2 The daily squared stock index return series over the period of 17 years from 1995 to 2011 for Brazil and India

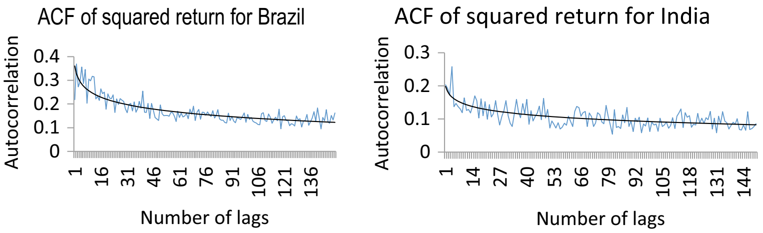


Fig. 3 Autocorrelation function for Brazil and India

FIMACH and 7147 for FIGARCH (Table 4b). The Ljung-Box statistics for standardized residuals for FIGARCH are smaller than that of residuals, but these are much larger than the corresponding statistics for FIMACH. Note that the Ljung-Box statistics for standardized residuals and residuals are the same for all FIMACH estimations. This may imply that the FIMACH model captures heteroskedasticity properly.

Employing the proposed FIMACH $(d, 0)$ model, we find that the squared stock return index for each of the United States and BRICS countries have long-memory properties. The persistence in terms of days varies among the countries. The effects of macroeconomics news and rumors on stock market volatility for India persist up to 35 days while the corresponding number of days for South Africa is 70. The stock market in China reacts initially more than any other BRICS countries since the fractional integration parameter (d) for China is 0.283, which is the largest among the BRICS countries. The stock market in Russia reacts initially with least $(d = 0.167)$ among those countries. The volatility intensity increases for all the seven stock markets when the macroeconomic news or rumors break out, and the impact remains between 35 and 70 days and fades away very slowly with time.

Table 4a. Empirical results for squared stock return index for Brazil, Russia, and India, respectively. The lag lengths are 45, 41, and 35 for respective countries

Coefficients	FIMACH (*d*, 0)			FIGARCH (0, *d*, 0)	FIGARCH (1, d, 1)	GARCH (1, 1)
	QMLE estimates (s.e.)	FGLS estimates (s.e.)	CLS estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)
Constant (Mean)	737.541*** (72.54)	737.541*** (73.65)	737.541*** (72.83)	129.996*** (8.71)	126.228*** (6.28)	101.091*** (8.41)
Constant (Variance)					−46632.882*** (1915.54)	634.873 (487.658)
d	0.208*** (0.029)	0.208*** (0.029)	0.208*** (0.028)	0.661*** (0.004)	0.751*** (0.003)	
α						1.081*** (0.03)
β					0.513*** (0.00)	0.879*** (0.02)
LB_{SD} (100)	1430.501***	1430.501***	1430.501***	5142.817***	4164.267***	1712.273***
Constant (Mean)	107.259*** (12.47)	107.259*** (14.48)	107.258*** (14.22)	24.075*** (1.86)	10.409*** (0.276)	2.901 (1.84)
Constant (Variance)					−1860.043*** (61.58)	5.340 (7.66)
d	0.167*** (0.04)	0.167*** (0.04)	0.167*** (0.04)	0.615*** (0.00)	0.728*** (0.00)	
α						1.302*** (0.10)

(continued)

Table 4a. (continued)

Coefficients	FIMACH (d, 0)			FIGARCH (0, d, 0)	FIGARCH (1, d, 1)	GARCH (1, 1)
	QMLE estimates (s.e.)	FGLS estimates (s.e.)	CLS estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)
β					0.537*** (0.00)	0.808*** (0.03)
LB_{SD} (100)	997.362***	997.362***	997.362***	3363.808***	1348.205***	112.467***
Constant (Mean)	12.709*** (2.57)	17.614*** (1.59)	17.614*** (1.59)	1.499*** (0.11)	1.579*** (0.11)	1.469*** (0.22)
Constant (Variance)					0.311* (0.18)	0.035 (0.23)
d	0.185*** (0.05)	−0.169*** (0.04)	−0.169*** (0.04)	0.549*** (0.11)	0.999*** (0.00)	
α			0.319 (0.18)			1.204*** (0.10)
β					0.934*** (0.02)	0.854*** (0.04)
LB_{SD} (100)	717.844***			1319.669***	1052.742***	1057.400***

Table 4b. Empirical results for squared stock return index for China, South Africa, and the United States, respectively. The lag lengths are 42, 70, and 70 for respective countries

Coefficients	FIMACH			FIGARCH (0, d, 0)	FIGARCH (1, d, 1)	GARCH (1, 1)
	QML estimates (s.e.)	FGLS estimates (s.e.)	CLS estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)
Constant (Mean)	0.283*** (0.11)	0.283*** (0.10)	0.283*** (0.10)	0.220*** (0.02)	0.077*** (0.01)	0.055*** (0.01)
Constant (variance)					−0.021*** (0.00)	−0.000 (0.00)
d	0.312*** (0.11)	0.312*** (0.11)	0.312*** (0.11)	0.490*** (0.00)	0.714*** (0.00)	
α						1.078*** (0.04)
β					0.684 (0.00)	(0.916) (0.02)
LB_{SD} (100)	437.879***	437.879***	437.879***	4259.401***	1871.683***	820.221***
Constant (Mean)	12.782*** (1.37)	20.314*** (1.28)	20.314*** (1.27)	4.181*** (0.298)	3 274*** (0.15)	3.600*** (0.28)
Constant (Variance)					−22.094*** (1.12)	0.587 (0.43)
d	0.194*** (0.02)	−0.166*** (0.02)	−0.166*** (0.02)	0.533 (0.00)	0.689*** (0.00)	
α						1.115*** (0.05)

(continued)

Table 4b. (continued)

Coefficients	FIMACH			FIGARCH (0, d, 0)	FIGARCH (1, d, 1)	GARCH (1, 1)
	QML estimates (s.e.)	FGLS estimates (s.e.)	CLS estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)	QML estimates (s.e.)
β					0.572*** (0.00)	0.867*** (0.03)
LB_{SD} (100)	772.396***	2684.763***	2684.764***	3285.210***	1606.489 ***	1393.617***
Constant (Mean)	70.071*** (8.539)	70.071*** (8.412)	70.071*** (8.403)	50.926*** (2.918)	18.044*** (1.616)	13.701*** (3.349)
Constant (Variance)					−1106.968*** (47.472)	−0.662 (2.993)
d	0.187*** (0.034)	0.187*** (0.035)	0.187*** (0.035)	0.472 (0.016)	0.669*** (0.004)	
α						1.045*** (0.018)
β					0.622*** (0.005)	0.915*** (0.015)
LB_{SD} (100)	693.986***	697.590***	693.986***	4432.055***	2445.256***	1061.702***

Conclusion

This paper introduces a new class of long-memory model for the volatility of stock returns. The model introduced is capable of taking account of heteroskedasticity in long memory. The conditional first- and second-order moments are provided. The CLS, FGLS, and QML are discussed and 2SQML estimator is proposed. In Monte Carlo experiments, we find that it is not appropriate to use ARFIMA, FIGARCH, or GARCH model if the data is generated according to FIMACH $(d, 0)$. The ARFIMA model reduces the serial correlation successfully, but it does not perform well like FIMACH. From the empirical results, we establish that the squared returns of the stock index for the BRICS countries and the United States have long-memory properties. However, the effects of macroeconomics news and rumors on stock return volatility vary among the countries. We also find that the volatility intensity increases for all the seven stock markets when the macroeconomic news or rumors break out, and that the impact remains between 35 and 70 days and fades away very slowly with time. CLS and FGLS estimators perform equally well in terms of residual properties, while the QML estimator performs somewhat better among the three estimators. The results of the simulation study indicate that 2SQML performs best among the five estimators and hence 2SQML is suggested to be used when QML does not perform well. Both in simulation and empirical studies, we find that the proposed model FIMACH outperforms FIGARCH in terms of eliminating serial correlations.

References

1. Engle, R.F., Patton, A.J.: What good is a volatility model? Quant. Financ. **1**, 237–245 (2001)
2. Poon, S.H., Granger, C.W.J.: Forecasting volatility in financial markets: a review. J. Econ. Lit. **41**, 478–539 (2003)
3. Hurst, H.E.: Long-term storage capacity of reservoirs. Trans. Amer. Soc. Civil Engrs **116**, 770–808 (1951)
4. Hurst, H.E.: Methods of using long-term storage in reservoirs. Proc. Inst. Civ. Eng. **1**, 519–543 (1956)
5. Mandelbrot, B.B., Van Ness, J.W.: Fractional Brownian motions, fractional noises and applications. SIAM Rev. **10**, 422–437 (1968)
6. Granger, C.W.J.: Long memory relationships and the aggregation of dynamic models. J. Econ. **14**, 227–238 (1980)
7. Granger, C.W.G., Joyeux, R.: An introduction to long memory time series models and fractional differencing. J. Time Ser. Anal. **1**, 15–29 (1980)
8. Hosking, J.: Fractional differencing. Biometrika **68**(1), 165–176 (1981)
9. Bhardwaj, G., Swanson, N.R.: An empirical investigation of the usefulness of ARFIMA models for predicting macroeconomic and financial time series. J. Econ. **131**(1&2), 539–578 (2006)
10. Ding, Z., Granger, C.W.J., Engle, R.F.: A long memory property of stock market returns and a new model. J. Empir. Financ. **1**, 83–106 (1993)
11. Baillie, R.T., Bollerslev, T., Mikkelsen, H.O.: Fractionally integrated generalized autoregressive conditional hetroskedasticity. J. Econ. **74**, 3–30 (1996)
12. Chung, C.F.: Estimating the fractionally integrated GARCH model. National Taiwan University, Working Paper (1999)
13. Quoreshi, A.M.M.S.: A long-memory integer-valued time series model, INARFIMA, for financial application. Quant. Financ. **12**, 2225–2235 (2014)

14. Geweke, J., Porter-Hudak, S.: The estimation and application of long memory time series models. J. Time Ser. Anal. **4**(4) (1983)
15. Brännäs, K., Hall, A.: Estimation in integer-value moving average models. Appl. Stoch. Model. Bus. Ind. **17**, 277–291 (2001)
16. Bollerslev, T., Mikkelsen, H.O.: Modeling and pricing long memory in stock market volatility. J. Econ. **73**(1996), 151–184 (1996)
17. Diebold, F.X.: Random walks versus fractional integration: power comparisons of scalar and joint tests of the variance-time function. In: Raj, B. (ed.) Advances in Econometrics. Kluwer, Dordrecht (1989)
18. Quoreshi, A.M.M.S.: A bivariate integer-valued long-memory model for high-frequency financial count data. Commun. Stat. Theory Methods **46**(3) (2017)
19. Weiss, A.A.: Asymptotic theory for ARCH models: estimation and testing. Econ. Theory **2**, 107–131 (1986)
20. Bollerslev, T., Wooldridge, J.M.: Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. Econ. Rev. **11**, 143–172 (1992)
21. Brännäs, K., Quoreshi, A.M.M.S.: Integer-value moving average modeling of the transactions in stocks. Appl. Financ. Econ. **20**, 1429–1440 (2010)
22. Smith, J., Taylor, N., Yadav, S.: Comparing the bias and Misspecification in ARFIMA models. J. Time Series Anal. **18**, 507–527 (1996)
23. Drost, F.C., van den Akker, R., Werker, B.J.M.: Efficient estimation of auto-regression parameters and innovation distributions for semiparametric integer-valued AR(p) models. J. Roy. Statist. Soc. B, Statist. Methodol. **71**, 467–485 (2009)

Using Subspace Methods to Model Long-Memory Processes

Dietmar Bauer

Abstract Subspace methods have been shown to be remarkably robust procedures providing consistent estimates of linear dynamical state-space systems for (multivariate) time series in different situations including stationary and integrated processes without the need for specifying the degree of persistence. Fractionally integrated processes bridge the gap between short-memory processes corresponding to stable rational transfer functions and integrated processes such as unit root processes. Therefore, it is of interest to investigate the robustness of subspace procedures for this class of processes. In this paper, it is shown that a particular subspace method called canonical variate analysis (CVA) that is closely related to long vector autoregressions (VAR) provides consistent estimators of the transfer function corresponding to the data generating process also for fractionally integrated processes of the VARFIMA or FIVARMA type, if integer parameters such as the system order tend to infinity as a suitable function of the sample size. The results are based on analogous statements for the consistency of long VAR modelling. In a simulation study, it is demonstrated that the model reduction implicit in CVA leads to accuracy gains for the subspace methods in comparison to long VAR modelling.

Keywords Fractional integration · Subspace algorithms · Long VAR-models

Introduction

For the estimation of linear dynamic models of the VARMA (vector autoregressive moving average) type for (multivariate) times series, the so-called subspace methods have been shown to provide attractive features in many different settings (for a review see e.g. [1]): in the stationary case, a particular type of algorithm termed *canonical variate analysis* (CVA) [9][1] provides estimators of rational transfer functions that are asymptotically equivalent to quasi-maximum likelihood estimators. Hence in the

[1] This algorithm has also been called *canonical correlation analysis* (CCA) in the literature.

D. Bauer (✉)
Econometrics Bielefeld University, Universitätsstrasse 25, 33615 Bielefeld, Germany
e-mail: Dietmar.Bauer@uni-bielefeld.de

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_12

case of Gaussian innovations, efficient estimators are obtained for known order of the data generating system. Consistency for this class of estimators has also been established for $I(1)$ processes (consistency for an adaptation is shown in [2] and consistency of CVA is claimed in [1]; the corresponding result has not yet been published).

Besides the asymptotic properties, the main appeal of subspace methods lies in their conceptual simplicity: they can be interpreted as applying model reduction to an initial high-order vector autoregression (in the following 'long VAR') estimate. Thus, their properties are in many cases easy to understand, problems in the data such as missing values (for example, due to eliminated outliers) can be handled easily and the effects of demeaning and de-trending are in most cases straightforward to understand. Additionally, also numerically efficient implementations are heavily used.

As a final advantage within the algorithm, also information on the suitability of the system order used for estimation is obtained from the reduced rank regression step. In this step, the order can be estimated or tested for.

As CVA provides consistent transfer function estimates for stationary short-memory processes as well as for integrated processes, it appears tempting to investigate the asymptotic properties of system estimators using CVA in the setting of fractionally integrated processes.

In this respect, the main contribution of this paper is to establish consistency of the transfer function estimators when the order of the system tends to infinity at an appropriate rate and the data generating process is a fractionally integrated process of the VARFIMA (vector autoregressive fractionally integrated moving average) or FIVARMA (fractionally integrated vector autoregression moving average) type. The order of the fractional integration here may differ for each component of the process in the range $0 \leq d_j < 0.5$ or $-0.5 < d_j \leq 0$. Based on consistent estimation (including upper bounds for the order of convergence) for the underlying transfer function, initial guesses for subsequent maximum likelihood estimation can be obtained.

Due to the intimate relation between CVA and long VAR modelling, this paper is largely based on and slightly extends the knowledge on the properties of long VAR estimators in the setting of vector processes with different fractional integration orders d_j.

The organization of the paper is as follows: in the next section, a brief description of the CVA algorithm is provided. The main results of the paper are contained in section "Results for Fractionally Integrated Processes". The finite sample properties are investigated in a case study in section "Simulation Study". Section "Conclusions" concludes the paper. Proofs are delegated to the Appendix.

The CVA Subspace Algorithm

Subspace algorithms are used for the identification of linear dynamic systems for (multivariate) time series $(y_t)_{t\in\mathbb{Z}}$, $y_t \in \mathbb{R}^s$, in state-space representation:

$$x_{t+1} = Ax_t + K\varepsilon_t, \quad y_t = Cx_t + \varepsilon_t \tag{1}$$

where $(x_t)_{t\in\mathbb{Z}}, x_t \in \mathbb{R}^n$ denotes the unobserved state process and the matrices $A \in \mathbb{R}^{n\times n}, C \in \mathbb{R}^{s\times n} and K \in \mathbb{R}^{n\times s}$. The process $(\varepsilon_t)_{t\in\mathbb{Z}}$ denotes the s-dimensional innovation process which is a white noise. In this paper, we consider $(\varepsilon_t)_{t\in\mathbb{Z}}$ as an independent identically distributed (iid) sequence.

The stability assumption $|\lambda_{max}(A)| < 1$ (the largest modulus of all eigenvalues of A is smaller than 1) implies that there exists a stationary solution to (1):

$$y_t = \sum_{i=0}^{\infty} k_i \varepsilon_{t-i} = \varepsilon_t + \sum_{i=1}^{\infty} CA^{i-1}K\varepsilon_{t-i}.$$

The impulse response coefficients $k_i = CA^{i-1}K, i \in \mathbb{N}$ corresponding to the system (A, C, K) are related to the rational transfer function

$$k(z) = I_s + \sum_{i=1}^{\infty} CA^{i-1}Kz^i = a^{-1}(z)b(z)$$

for some polynomial matrices $a(z) \in \mathbb{R}^{s\times s}, b(z) \in \mathbb{R}^{s\times s}$ building a VARMA representation of the state-space system. Without restriction of generality, we will only consider minimal state-space representations (cf. Chap. 2 of [6]).

The main idea for estimation using subspace methods is to note that the state Eqs. (1) are linear in the system matrices for known state. Thus if an estimate of the state is available, the system can be estimated using least squares. An estimate for the state is obtained from the equation (see [1] for details):

$$Y^+_{t,f} = \mathcal{O}_f x_t + \mathcal{E}_f E^+_{t,f} = \mathcal{O}_f \mathcal{K}_p Y^-_{t,p} + \mathcal{O}_f \underline{A}^p x_{t-p} + \mathcal{E}_f E^+_{t,f} = \beta_1 Y^-_{t,p} + N^+_{t,f}. \tag{2}$$

Here, $Y^+_{t,f} := [y_t', y_{t+1}', \ldots, y_{t+f-1}']'$, $E^+_{t,f} := [\varepsilon_t', \varepsilon_{t+1}', \ldots, \varepsilon_{t+f-1}']'$ for some integer f, $Y^-_{t,p} := [y_{t-1}', \ldots, y_{t-p}']'$, $\mathcal{K}_p := [K, \underline{A}K, \underline{A}^2K, \ldots, \underline{A}^{p-1}K]$ for $\underline{A} := A - KC$ and $\mathcal{O}_f := [C', A'C', \ldots, (A^{f-1})'C']'$. Expressions for $\mathcal{E}_f$ are given in [1].

In the following, we use $\langle a_t, b_t\rangle := T^{-1}\sum_{t=p+1}^{T-f+1} a_t b_t'$ for sequences $(a_t)_{t\in\mathbb{N}}$ and $(b_t)_{t\in\mathbb{N}}$. Then CVA proceeds as follows:

1. Specify the integer values f, p.
2. Solve the rank restricted regression problem $Y^+_{t,f} = \beta_1 Y^-_{t,p} + N^+_{t,f}$ under the rank constraint $\text{rank}(\beta_1) = n$ (to be specified in this step) to obtain an estimate $\hat{\mathcal{O}}_f\hat{\mathcal{K}}_p := [(\hat{\Xi}_f)^{-1}\hat{U}_n\hat{S}_n][\hat{V}_n'\hat{\Xi}^-_p]$ of β_1 using the SVD (singular value decomposition)

$$\begin{aligned} \hat{\Xi}_f\hat{\beta}_1\hat{\Xi}^-_p = \hat{U}\hat{S}\hat{V}' \quad &= \hat{U}_n\hat{S}_n\hat{V}_n' + \hat{R}_n, \\ \hat{\Xi}_f := \langle Y^+_{t,f}, Y^+_{t,f}\rangle^{-1/2} \ , \quad &\hat{\Xi}^-_p := \langle Y^-_{t,p}, Y^-_{t,p}\rangle^{1/2}. \end{aligned}$$

Here, $\hat{\beta}_1 = \langle Y^+_{t,f}, Y^-_{t,p}\rangle\langle Y^-_{t,p}, Y^-_{t,p}\rangle^{-1}$, $\hat{U}_n \in \mathbb{R}^{fs\times n}$ denotes the matrix whose columns are the left singular vectors to the singular values which are the diagonal entries in[2] $\hat{S}_n := \text{diag}(\hat{\sigma}_1, \hat{\sigma}_2, \dots, \hat{\sigma}_n)$, $\hat{\sigma}_1 \geq \dots \geq \hat{\sigma}_n > \hat{\sigma}_{n+1} > 0$ and $\hat{V}_n \in \mathbb{R}^{ps \times n}$ contains the corresponding right singular vectors as its columns. $\hat{R}_n$ denotes the approximation error.

3. Use the estimated state $\hat{x}_t := \hat{\mathcal{K}}_p Y^-_{t,p}$, $t = p + 1, \dots, T + 1$ to obtain estimates $(\hat{A}, \hat{C}, \hat{K})$ of the system matrices from regression in the system equations.

The choice of f and p influences the asymptotic properties of the estimators. For fractionally integrated processes, we will let $f = p$ as well as the order n tend to infinity at a rate to be defined later on.

Note that for $f = p, n = sp$ typically the then square matrix $\hat{\mathcal{K}}_p$ will be regular and hence the estimated state equals $Y^-_{t,p}$ subject to a basis change. It then follows that the estimated transfer function coincides with the VAR(p) estimate. The CVA estimator for $n < sp$ then can be interpreted as a particular method of model reduction starting from the estimated long VAR model, cf. [4].

Results for Fractionally Integrated Processes

In this paper, we consider fractionally integrated processes of the VARFIMA or FIVARMA type (cf. e.g. [13]):

Definition 3.1 The process $(y_t)_{t\in\mathbb{Z}}$ is called a *vector autoregressive fractionally integrated moving average (VARFIMA)* process if it is obtained as the stationary solution to the vector difference equation:

$$a(L)y_t = b(L)e_t, \qquad e_t = D(L)^{-1}\varepsilon_t$$

where $(a(z), b(z))$ is a stable and invertible VARMA system of left co-prime polynomial matrices $a(z) = I_s + a_1 z + \dots + a_p z^p$, $b(z) = I_s + b_1 z + \dots + b_q z^q$, $D(L) = \text{diag}[(1 - L)^{d_j}]_{j=1,\dots,s}$ is a diagonal matrix of fractionally integrating filters where $|d_j| < 0.5$, $\max_{j=1,\dots,s}|d_j| > 0$ and where $(\varepsilon_t)_{t\in\mathbb{Z}}$ is an iid white noise sequence. L denotes the backward shift operator.

Thus a VARFIMA process applies a VARMA filter to a fractionally integrated noise sequence. The FIVARMA model reverses the order by filtering a VARMA process with a fractionally integrating filter:

Definition 3.2 The process $(y_t)_{t\in\mathbb{Z}}$ is called a *fractionally integrated vector autoregressive moving average (FIVARMA)* process if it is obtained as the stationary solution to the vector difference equation

$$y_t = D(L)^{-1}u_t, \qquad a(L)u_t = b(L)\varepsilon_t,$$

[2] In the unlikely case of identically estimated singular values $\hat{\sigma}_n = \hat{\sigma}_{n+1}$, the basis in the corresponding spaces is chosen randomly.

where $(a(z), b(z))$ is a stable and invertible VARMA system of left co-prime polynomial matrices $a(z) = I_s + a_1 z + \cdots + a_p z^p$, $b(z) = I_s + b_1 z + \cdots + b_q z^q$, $D(L) = \text{diag}[(1-L)^{d_j}]_{j=1,\ldots,s}$ is a diagonal matrix of fractionally integrating filters where $|d_j| < 0.5$, $\max_{j=1,\ldots,s}|d_j| > 0$ and where $(\varepsilon_t)_{t\in\mathbb{Z}}$ is an iid white noise sequence. L denotes the backward shift operator.

In the scalar case $s = 1$, the two definitions are identical as in this case the operators $D(z)$ and $a(z)^{-1}b(z)$ commute, whereas in the vector case they are different if not all d_j are identical (compare [13]). Note also that $|d_j| < 0.5$ contains the case that some (but not all) d_j are zero. Therefore, the processes may contain a mixture of long- and short-memory processes. We will always assume that there exists at least one integration order $d_j \neq 0$ to rule out the case of exclusively short-memory processes, for which the properties of CVA are well known. Hosking [7] derived a number of common properties:

Theorem 3.1 *Let $(y_t)_{t\in\mathbb{Z}}$ be a stationary FIVARMA or VARFIMA process with fractional integration orders $d_1, \ldots, d_s$. Then*

(i) $y_t = \sum_{j=0}^{\infty} k_j \varepsilon_{t-j}$ *where* $\|k_j\| \leq M_k j^{d_+ - 1}$ *where* $d_+ = \max(d_1, \ldots, d_s; d_j \neq 0) < 0.5$.

(ii) $y_t = \varepsilon_t + \sum_{j=1}^{\infty} \phi_j y_{t-j}$ *where* $\|\phi_j\| \leq M_\phi j^{-1-d_-}$ *where* $d_- = \min(d_1, \ldots, d_s; d_j \neq 0) > -0.5$.

The theorem shows that the two orders d_- and d_+ determine the rate of decay of the coefficients in the AR(∞) and MA(∞) representations, both having square summable coefficients.

The processes included in the above setup do not allow for a state-space representation with a finite state dimension. It has been shown by [3] that fractionally integrated processes correspond to infinite state dimension while at the same time approximations with finite state dimension exist in the sense that by allowing the state dimension to grow, a convergent sequence of transfer functions can be obtained. It is the main contribution of this paper to show that this holds for transfer functions estimated using CVA (for the proof see section "Proof of Theorem 3.2"):

Theorem 3.2 *Let the process $(y_t)_{t\in\mathbb{Z}}$ be a VARFIMA or FIVARMA process generated according to Definition 3.1 or Definition 3.2 with d_- and d_+ defined in Theorem 3.1 where the white noise process $(\varepsilon_t)_{t\in\mathbb{Z}}$ is independent identically distributed with zero mean, non-singular variance Ω and finite fourth moments. Let $(\hat{A}_n, \hat{C}_n, \hat{K}_n)$ denote the* CVA *estimates using $f = p = p(T) \to \infty$, choosing a system order n. Further, let $\phi(z) = I_s - \sum_{j=1}^{\infty} \phi_j z^j = k^{-1}(z)$, $\hat{k}_n(z) = I_s + z\hat{C}_n(I_n - \hat{A}_n z)^{-1}\hat{K}_n$, $\hat{\phi}_n(z) = \hat{k}_n^{-1}(z)$.*

(I) If $0 < d_- \leq d_+ < 0.5$ and if $d_- > 2d_+ - 0.5$ then for $p = p(T) \to \infty$ such that $pR_T(d_+) \to 0$ (where $R_T(d) := (T/\log T)^{d-1/2}$), then there exists a choice $n = n(T)$ such that

$$\|\hat{\phi}_n(z) - \phi(z)\|_2 = o(pR_T(d_+)) + O(p^{2d_+ - d_- - 0.5}) = o(1)$$

where $\|f\|_2^2 = \int_{-\pi}^{\pi} \|f(e^{i\omega})\|^2 d\omega$ *denotes the two norm in the space of functions defined on the unit circle.*

(II) If $-0.5 < d_- \leq d_+ < 0$ *and* $2d_+ < 0.5 + 3d_-$ *and if* $p = p(T) \to \infty$ *such that* $p = o((T/\log T)^{1/(2-8d_-)})$, *then there exists a choice* $n = n(T)$ *such that*

$$\|\hat{\phi}_n(z) - \phi(z)\|_2 = o((\log T/T)^{1/2} p^{1-4d_-}) + O(p^{2d_+-3d_--0.5}) = o(1).$$

The theorem allows for different d_- and d_+, however, both must have the same sign. Additional short-memory components of the VARMA type are allowed for (that is, some of the d_j but not all can be zero; of course the case that all d_j are zero is well covered in the literature). The proof of the theorem extends the current knowledge on long VAR approximation in the long-memory setting as it implies that for appropriate increase of p as a function of the sample size autoregressions can be used in order to approximate the data generating process in the FIVARMA and VARFIMA case for different values of d_j.

Note that the results also imply consistency for the transfer function $k(z)$:

$$\|\hat{k}_n(z) - k(z)\|_2 \leq \|\hat{\phi}_n(z)^{-1}\|_2 \|\phi(z)^{-1}\|_2 \|\phi(z) - \hat{\phi}_n(z)\|_2.$$

Thus, convergence for $\hat{k}_n(z)$ is of the same order.

Investigating the order of convergence in more detail, consider the case $d_- = d_+$ and let p be chosen as $p = T^{\epsilon}$. Then in both cases, the total order of convergence consists of two terms where the first term is due to the sample error and becomes small for large T and small p while the second term is due to the approximation error and hence is small for large p. Omitting the term due to $\log T$, we obtain a rate $\epsilon(d)$ compromising between the two terms such that both terms achieve the order $T^{r(d)}$. Figure 1 provides the corresponding plot. It follows that around $d \approx 0$ the 'optimal' rate equals 1/3 which decreases for positive and negatives values. It reaches 0 for $d = 0.5$ and 1/6 at $d = -0.5$. With respect to the rate $r(d)$, we obtain $r(0) = -1/6$

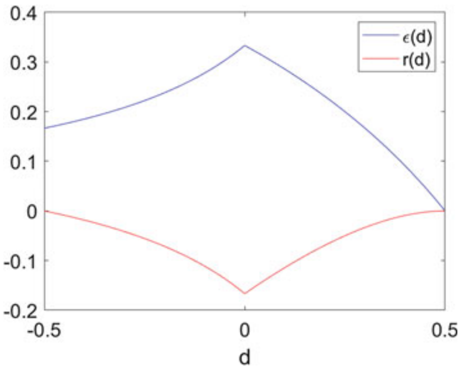


Fig. 1 'Optimal' rate $\epsilon(d)$ and the achieved convergence rate $r(d)$ as a function of $d = d_- = d_+$

which decreases to $r(\pm 0.5) = 0$ indicating that at both boundaries convergence is extremely slow. Note that the graph is not symmetric and not differentiable at 0, both for $\epsilon(d)$ and $r(d)$.

Simulation Study

In this section, the consistency result of the last section is complemented by a small case study on the finite sample properties of the estimators compared to alternative estimators. To this end, 1000 trajectories of univariate ARFIMA models are simulated for 1100 time points where the first 100 are discarded. The AR and the MA polynomial are generated from 4 randomly drawn (for each replication) real poles (inverses are uniformly $[-0.5, 0.5]$ distributed) and zeros (inverses are uniformly $[-0.9, 0.9]$ distributed). The corresponding ARMA processes are filtered using $(1-z)^{-d}$. Three methods of estimation are tested:

- non-parametric estimates of the parameter d using the Whittle likelihood followed by CVA applied to the pre-filtered series
- CVA estimates for the original series using $f = p = 2\hat{p}_{AIC}$ and selecting the order n according to the criterion SVC.
- long AR approximation with lag order chosen using AIC as $\hat{p}_{AIC}$.

Two resulting plots are provided in Fig. 2. The main message from the plot is that the CVA outperforms the long AR approximation in terms of accuracy, both for positive and negative d values and is comparable to the Whittle estimation for most of the frequencies.

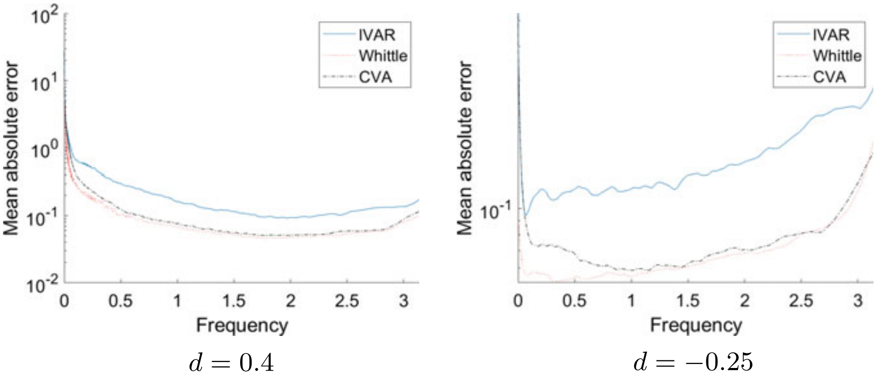


Fig. 2 Mean absolute error for true transfer function estimates for the three estimators

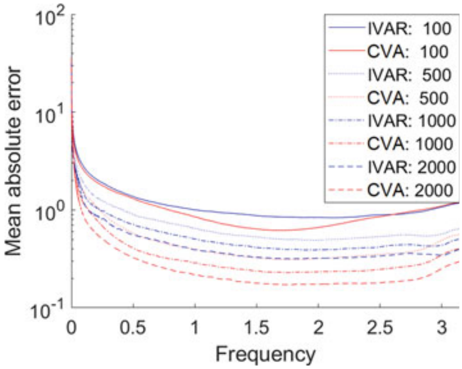


Fig. 3 Mean of norm of transfer function estimation errors for the four sample sizes $T = 100, 500, 1000$ and $T = 2000$

Second, also a bivariate process where the first component is a FIVARMA with $d = 0.4$ and the second is a FIVARMA with $d = 0.1$ is investigated. For a total of 1000 replications of time series of length $T = 100, 500, 1000$ and $T = 2000$, ARMA systems with four poles uniformly in $[-0.5, 0.5]$ and four zeros uniformly in $[-0.9, 0.9]$ are drawn. Then estimation is performed with the long VAR approach and the CVA approach. Figure 3 provides the mean of the one norm of the difference between the estimates and the true transfer function as a function of the frequency. It can be seen that both procedures show a decreasing error for increasing sample size. Additionally, CVA outperforms long AR approximation for most frequencies.

Conclusions

This paper shows that (under appropriate assumptions on the fractional integration orders) the transfer function corresponding to a VARFIMA or FIVARMA process can be estimated consistently using the CVA approach when the maximal lag order p tends to infinity as a function of the sample size. This adds fractionally integrated processes to the list of settings in which CVA provides useful outcomes.

Preliminary simulations verify that the model reduction employed in CVA is beneficial in terms of estimation accuracy in comparison to using the full-order long VAR approximation. The order selection criterion used in the paper is based on a simple heuristic and hence, there appears to be potential for further improvement. This is left for future research.

Acknowledgements The author is grateful to Philipp Sibbertsen and Christian Leschinski for discussions and comments on the contents of the paper.

Appendix

Auxiliary Lemmas

Lemma A.1 *Let* $y_t = \sum_{j=0}^{\infty} k_j \varepsilon_{t-j}$ *where* $(\varepsilon_t)_{t\in\mathbb{Z}}$ *is an iid sequence of random variables having zero mean and finite fourth moments. Let* $\hat{\gamma}_j = \frac{1}{T}\sum_{t=1+j}^{T}(y_t - \bar{y})(y_{t-j} - \bar{y})'$ *where* $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$, $\tilde{\gamma}_j = \frac{1}{T}\sum_{t=1}^{T} y_{t+j}y_t' - \bar{y}\bar{y}'$ *and* $\gamma_j = \mathbb{E}y_t y_{t-j}'$. *Assume that* $k_j = O(j^{d-1})$ *where* $-0.5 < d < 0.5$*. Then,*
(i) $\|\gamma_j\| \leq \kappa j^{2d-1}$ *for* $j > 0$.
(ii) $\mathbb{E}\bar{y}\bar{y}' = O(T^{2d-1})$ *and* $\mathbb{E}\|\bar{y}\|^4 = O(T^{4d-2})$.
(iii) $\mathbb{E}\|\hat{\gamma}_j - \tilde{\gamma}_j\| \leq \tau j/T$ *where* $0 < \tau < \infty$ *does not depend on* j.
(iv) $\hat{\gamma}_j - \gamma_j = O_P(j/T) + (\tilde{\gamma}_j - \gamma_j)$ *where* $\mathbb{E}\text{vec}(\tilde{\gamma}_j - \gamma_j)\text{vec}(\tilde{\gamma}_k - \gamma_k)' = O(Q_T(d)^2)$ *with*

$$Q_T(d) = \begin{cases} T^{2d-1} & , \text{ for } \ 0.25 < d < 0.5 \\ T^{-1/2}\log T & , \qquad d = 0.25, \\ T^{-1/2} & , \ -0.5 < d < 0.25. \end{cases}$$

(v) Let $R_T(d) := (T/\log T)^{d-1/2}$*. For* $H_T R_T(\tilde{d}) \to 0$ *for* $\tilde{d} = \max(0, d)$*, we have*

$$\max_{0\leq j\leq H_T} \|\hat{\gamma}_j - \gamma_j\| = O(R_T(\tilde{d})), \quad \max_{0\leq j\leq H_T} \|\tilde{\gamma}_j - \gamma_j\| = O(R_T(\tilde{d})).$$

Proof The proof uses the results of Theorems 1, 3 and 5 of [8] and Theorems 1 and 2 of [12].

(i) Using $\Omega = \mathbb{E}\varepsilon_t\varepsilon_t'$, we have for some constant $0 < \kappa < \infty$ not depending on j

$$\|\gamma_j\| = \|\sum_{i=0}^{\infty} k_i \Omega k_{i+j}'\| \leq \mu \sum_{i=0}^{\infty} \|k_i\|\|k_{i+j}\| \leq \mu\bar{\mu}^2 \sum_{i=0}^{\infty} i^{d-1}(i+j)^{d-1} \leq \kappa j^{2d-1}$$

since $\|\Omega\| < \mu$, $\|k_i\| \leq \bar{\mu} i^{d-1}$ for some $\bar{\mu} < \infty$, $i < i + j$ and the techniques in the proof of Lemma 3.2 of [3].

(ii) The first part follows directly from (i) in combination with [8, Theorem 1] dealing with each coordinate separately.

Below, we deal without restriction of generality with scalar processes. The vector case is merely notationally more complex. With respect to the second part, note that $\mathbb{E}y_t y_s y_r y_0 = \gamma_{t-s}\gamma_r + \gamma_{t-r}\gamma_s + \gamma_t\gamma_{s-r} + \kappa_4(t, s, r)$ for $\kappa_4(t, s, r) := \sum_{a=-\infty}^{\infty} k_{a+t}k_{a+s}k_{a+r}k_a(\mathbb{E}\varepsilon_t^4 - (\mathbb{E}\varepsilon_t^2)^2)$ where for notational simplicity $k_a = 0, a < 0$ is used. Next,

$$\mathbb{E}\bar{y}^4 = \frac{1}{T^4}\sum_{t,s,r,u=1}^{T} \mathbb{E}y_t y_s y_r y_u = \frac{1}{T^4}\sum_{t,s,r,u=1}^{T} \gamma_{t-s}\gamma_{r-u} + \gamma_{t-r}\gamma_{s-u} + \gamma_{t-u}\gamma_{s-r} + \kappa_4(t-u, s-u, r-u) \quad (3)$$

is the sum of four terms where the first three are identical:

$$T^{-4}\sum_{t,s,r,u=1}^{T} \gamma_{t-s}\gamma_{r-u} = \left(T^{-2}\sum_{t,s=1}^{T} \gamma_{t-s}\right)^2 = (\mathbb{E}\bar{y}^2)^2 = O(T^{4d-2}).$$

The last term equals the fourth term of (A.2) in [8]. Using Lemma 3.2. (i) of [3] in the fourth row of the equation on p. 277 of [8], we obtain

$$T^{-4}\sum_{t,s,r,u=1}^{T} \kappa_4(t-u, s-u, r-u) = O(T^{2d-2})$$

for $0.25 \le d < 0.5$. By dominated convergence, we have that this term is $O(T^{-1})$ for $-0.5 < d < 0.25$. Hence, we obtain the bound $\mathbb{E}\bar{y}^4 = O(T^{4d-2})$ for $d > 0.25$ and $O(T^{-1})$ else.

(iii) For $j > 0$, we obtain

$$\begin{aligned}
\hat{\gamma}_j &= \frac{1}{T}\sum_{t=1+j}^{T} (y_t - \bar{y})(y_{t-j} - \bar{y})' = \frac{1}{T}\sum_{t=1+j}^{T} (y_t y_{t-j}' - \bar{y}y_{t-j}' - y_t\bar{y}' + \bar{y}\bar{y}') \\
&= \frac{1}{T}\sum_{t=1+j}^{T} y_t y_{t-j}' - \bar{y}(\frac{1}{T}\sum_{t=1+j}^{T} y_{t-j})' - (\frac{1}{T}\sum_{t=1+j}^{T} y_t)\bar{y}' + \bar{y}\bar{y}'\frac{T-j}{T} \\
&= \frac{1}{T}\sum_{t=1}^{T} y_{t+j}y_t' - \frac{1}{T}\sum_{t=T-j+1}^{T} y_{t+j}y_t' - \bar{y}(\bar{y} - \frac{1}{T}\sum_{t=T-j+1}^{T} y_t)' - (\bar{y} - \frac{1}{T}\sum_{t=1}^{j} y_t)\bar{y}' + \bar{y}\bar{y}'\frac{T-j}{T} \\
&= \tilde{\gamma}_j - \frac{1}{T}\sum_{t=T-j+1}^{T} y_{t+j}y_t' + \frac{1}{T}\sum_{t=T-j+1}^{T} \bar{y}y_t' + \frac{1}{T}\sum_{t=1}^{j} y_t\bar{y}' - \frac{j}{T}\bar{y}\bar{y}'.
\end{aligned}$$

(iv) From (iii), it follows that $\hat{\gamma}_j = \tilde{\gamma}_j + O_P(j/T)$. The rest follows from [8].

(v) Noting that Theorems 1 and 2 of [12] only use the variance bounds derived above, uniformity of a.s. convergence follows.

Lemma A.2 *Let $(y_t)_{t\in\mathbb{Z}}$ be as in Lemma A.1. Define $k(z) := \sum_{j=0}^{\infty} k_j z^j$ and the spectrum $f(\omega) = \frac{1}{2\pi}k(e^{i\omega})\Omega k(e^{i\omega})^*$ where $\Omega := \mathbb{E}\varepsilon_t\varepsilon_t'$.*

(i) Assume that there exist constants $0 < a, b < \infty$ such that $aI_s \le f(\omega) \le bI_s\overline{f}(\omega)$ where $\overline{f}(\omega) = |1 - e^{i\omega}|^{-2d}$ for $d \ge 0$. Then there exists constants $0 < C_1 < C_2 < \infty$ such that $C_1 I_{ps} \le \mathbb{E}Y_{t,p}^-(Y_{t,p}^-)' \le C_2 I_{ps} p^{2d}$.

(ii) If there exist constants $0 < a, b < \infty$ *such that* $aI_s\underline{f}(\omega) \le f(\omega) \le bI_s$ *where* $\underline{f}(\omega) = |1 - e^{i\omega}|^{-2d}$ *for* $d < 0$, *then there exist constants* $0 < C_1 < C_2 < \infty$ *such that* $C_1 I_{ps} p^{2d} \le \mathbb{E}Y^-_{t,p}(Y^-_{t,p})' \le C_2$.

Proof The proof is a straightforward generalization of the univariate result in Theorem 2 of [11]; compare also Lemma 2 in [12]. Only (i) is proved, the remaining statements follow analogously. Let $\Gamma^-_p = \mathbb{E}Y^-_{t,p}(Y^-_{t,p})'$. Note that the definition of $\overline{f}$ and $\underline{f}$ coincide with the definition given in (9), (10), p. 96 of [11].

The smallest eigenvalue of Γ^-_p equals the minimum of $x'\Gamma^-_p x$ for $x'x = 1$ and the largest corresponds to the maximum. Since $(y_t)_{t\in\mathbb{Z}}$ is assumed to be stationary with spectral density $f(z) := k(z)\Omega k(z)^*/(2\pi)$, it follows that

$$x'\Gamma^-_p x = \int_{-\pi}^{\pi} (\sum_{j=1}^{p} x_j e^{ij\omega})^* f(e^{i\omega})(\sum_{j=1}^{p} x_j e^{ij\omega})d\omega$$

where $x' = [x_1', \dots, x_p']$, $x_j \in \mathbb{R}^s$. If $d > 0$ then

$$\int_{-\pi}^{\pi} (\sum_{j=1}^{p} x_j e^{ij\omega})^* f(e^{i\omega})(\sum_{j=1}^{p} x_j e^{ij\omega})d\omega \ge \int_{-\pi}^{\pi} (\sum_{j=1}^{p} x_j e^{ij\omega})^* (\sum_{j=1}^{p} x_j e^{ij\omega})d\omega a \ge 2\pi a.$$

Also,

$$\int_{-\pi}^{\pi} (\sum_{j=1}^{p} x_j e^{ij\omega})^* f(e^{i\omega})(\sum_{j=1}^{p} x_j e^{ij\omega})d\omega \le b\int_{-\pi}^{\pi} \| \sum_{j=1}^{p} x_j e^{ij\omega} \|^2 \overline{f}(\omega)d\omega.$$

It follows that the function $h(\omega) := \| \sum_{j=1}^p x_j e^{ij\omega} \|^2$ is in the set P_p (p. 97 of [11]). Hence the result holds. □

Lemma A.3 *Let* $(y_t)_{t\in\mathbb{Z}}$ *be as in Theorem 3.2. Then,*

- *For* $0.25 < d_+ < 0.5$, *we have* $\|\mathbb{E}Y^+_{t,f}(Y^-_{t,p})'\|_{Fr} = O((f+p)^{2d_+})$, $\|\mathbb{E}Y^-_{t,p}(Y^-_{t,p})' \|_{Fr} = O(p^{2d_+})$.
- *For* $d_+ = 0.25$, *we have* $\|\mathbb{E}Y^+_{t,f}(Y^-_{t,p})'\|_{Fr} = O((f+p)^{1/2})$, $\|\mathbb{E}Y^-_{t,p}(Y^-_{t,p})'\|_{Fr} = O(\sqrt{p \log p})$.
- *For* $0 \le d_+ < 0.25$, *we have* $\|\mathbb{E}Y^+_{t,f}(Y^-_{t,p})'\|_{Fr} = O((f+p)^{2d_+})$, $\|\mathbb{E}Y^-_{t,p}(Y^-_{t,p})' \|_{Fr} = O(\sqrt{p})$.
- *For* $d_+ < 0$, *we have* $\|\mathbb{E}Y^+_{t,f}(Y^-_{t,p})'\|_{Fr} = O(1)$, $\|\mathbb{E}Y^-_{t,p}(Y^-_{t,p})'\|_{Fr} = O(\sqrt{p})$.

The lemma is an easy consequence of $\|\gamma_l\| \le \mu l^{2d_+-1}$ as shown in Lemma A.1 in combination with $\sum_{j=1}^m j^{\beta-1} = O(m^\beta)$ for $\beta > 0$.

Lemma A.4 *Let the assumptions of Theorem 3.2 hold. Let* $\tilde{d}_- = \min(d_-, 0)$, $\tilde{d}_+ = \max(d_+, 0)$ *and assume that* $p^{2-4\tilde{d}_-}R_T(\tilde{d}_+)^2 \to 0$ *and* $pR_T(\tilde{d}_+) \to 0$. *Then*

$$\begin{aligned}\|\langle Y^+_{t,f}, Y^-_{t,p}\rangle - \mathbb{E}Y^+_{t,f}(Y^-_{t,p})'\|_{Fr} &= O\left(\sqrt{fp}\left(R_T(\tilde{d}_+)\right)\right),\\ \|\langle Y^-_{t,p}, Y^-_{t,p}\rangle - \Gamma^-_p\|_{Fr} &= O(pR_T(\tilde{d}_+)),\\ \|\langle Y^-_{t,p}, Y^-_{t,p}\rangle^{-1} - (\Gamma^-_p)^{-1}\|_2 &= O(p^{1-4\tilde{d}_-}R_T(\tilde{d}_+))\end{aligned}$$

Proof All three parts follow almost immediately from Lemma A.1 (v). With respect to the third statement using [10], p. 397, l. 11, we obtain

$$\|\langle Y^-_{t,p}, Y^-_{t,p}\rangle^{-1} - (\Gamma^-_p)^{-1}\|_2 = F^2 Z_{p,T}/(1 - FZ_{p,T})$$

where $F := \|(\Gamma^-_p)^{-1}\|_2$, $Z_{p,T} := \|\langle Y^-_{t,p}, Y^-_{t,p}\rangle^{-1} - (\Gamma^-_p)^{-1}\|_2/F(\|\langle Y^-_{t,p}, Y^-_{t,p}\rangle^{-1} - (\Gamma^-_p)^{-1}\|_2 + F) \leq \|\langle Y^-_{t,p}, Y^-_{t,p}\rangle - \Gamma^-_p\|_2$. Then the upper bound on p implies that $Z_{p,T} = o(1)$. For $d_- > 0$ and since $\tilde{d}_- = 0$, we have $F < \infty$ and therefore the result follows. For $d_- < 0$, we have $F = O(p^{-2d_-})$ and therefore the upper bound on p has to ascertain that $FZ_{p,T} \to 0$ where $Z_{p,T} = O(pR_T(\tilde{d}_+))$. □

Lemma A.5 *Let the assumptions of Theorem 3.2 hold and let $\phi_j(p)$ denote the coefficients of the long VAR approximation in (4). Then $\sum_{j=1}^p \|\phi_j - \phi_j(p)\|^2 = O(p^{4d_+ - 2d_- - 1})$ if $d_+ \geq d_- \geq 0$ which tends to zero if $d_- > 2d_+ - 0.5$. If $d_+ = d_- = d$, this always holds and the order equals $O(p^{2d-1})$.*

If $d_- \leq d_+ \leq 0$ then $\sum_{j=1}^p \|\phi_j - \phi_j(p)\|^2 = O(p^{4d_+ - 6d_- - 1})$ which tends to zero if $2d_+ < 0.5 + 3d_-$. For $0 \geq d_+ = d_- > -0.5$, this always holds and the rate equals $O(p^{-2d-1})$ in this case.

Proof This result has already been obtained in the scalar case (where automatically $d_+ = d_-$) by [5]. Note that using $f = 1$, we obtain

$$0 = \mathbb{E}Y^+_{t,1}(Y^-_{t,p})' - \mathbb{E}Y^+_{t,1}(Y^-_{t,p})' = \beta_{1,p}\Gamma^-_p - [I_s, 0]\beta\Gamma^-_\infty[I_{ps}, 0]' = \left[\Phi_p - [\Phi]_p\right]\Gamma^-_p - \Phi^-_p\tilde{\Gamma}^-_{2,p}$$

where Φ^-_p denotes the matrix Φ where the first p block columns are omitted. Further, $\tilde{\Gamma}^-_{2,p} = \mathbb{E}Y^-_{t-p,\infty}(Y^-_{t,p})'$. Therefore it is sufficient to compute the Frobenius norm of $\Phi^-_p\tilde{\Gamma}^-_{2,p}$ which contains as a typical element $\sum_{i=1}^\infty \phi_{i+p}\mathbb{E}y_{t-p-i}y'_{t-j}$, $j = 1, \ldots, p$. Using $\|\phi_l\| \leq M_l l^{-1-d_-}$, $\|\gamma_l\| \leq M_g l^{2d_+ - 1}$, we can bound the norm of this entry by $M_l M_g \sum_{i=1}^\infty (i+p)^{-1-d_-}(p+i-j)^{2d_+ - 1}$. For $d_- > 0$, this is of order $O(p^{-d_-}(p-j)^{2d_+ - 1})$. Summing the squares over $j = 1, \ldots, p$ shows that the squared Frobenius norm of $\Phi^-_p\tilde{\Gamma}^-_{2,p}$ in this case is of order $O(p^{4d_+ - 2d_- - 1})$. Since the smallest eigenvalue of Γ^-_p is bounded away from zero for $d_- > 0$, the result follows. The result for $d_+ = d_-$ is obvious.

For $d_+ < 0$, the norm of the typical entry is of order $O(p^{-1-d_-}(p-j)^{2d_+})$ and therefore the squared Frobenius norm of $\Phi^-_p\tilde{\Gamma}^-_{2,p}$ also in this case is of order $O(p^{4d_+ - 2d_- - 1})$. Here the smallest eigenvalue of Γ^-_p tends to zero as p^{2d_-} and hence the inverse adds a factor p^{-2d_-} to the Frobenius norm adding up to order $O(p^{-1+4d_+ - 6d_-}) = o(1)$ if $-0.5 + 2d_+ - 3d_- < 0$. □

Proof of Theorem 3.2

The main insight into the algorithm lies in the fact that for $n = ps$ the CVA estimate $(\hat{A}, \hat{C}, \hat{K})$ equals the long VAR approximation of $(y_t)_{t\in\mathbb{Z}}$ using lag order p [4], the properties of which follow along the lines of [12]:

Lemma A.6 *Let $(y_t)_{t\in\mathbb{Z}}$ be as in Theorem 3.2 denoting*

$$y_t = \sum_{i=1}^{p} \phi_i(p) y_{t-i} + \varepsilon_t(p). \tag{4}$$

Further, let the assumptions on d_-, d_+ be as in Theorem 3.2. Then the OLS estimates of the coefficients in this long VAR approximation fulfil uniformly in $1 \le p \le H_T$ where H_T is such that $H_T^{2-4\tilde{d}_-} R_T(\tilde{d}_+)^2 \to 0$, $R_T(\tilde{d}_+) P_T(d_-, H_T) \to 0$ where $P_T(d, p) = p$ for $d \ge 0$ and $P_T(d, p) = p^{1-4d}$ for $d < 0$

$$\max_{1\le p\le H_T} \left(\sum_{j=1}^{p} \|\hat{\phi}_j(p) - \phi_j(p)\|^2 / P_T(d_-, p)^2 \right)^{1/2} = O(R_T(\tilde{d}_+)).$$

Proof The proof follows the arguments of [12]. The long AR approximation can be written as $y_t = \beta_{1,p} Y_{t,p}^- + \varepsilon_t(p)$. It follows that $\hat{\beta}_{1,p} = \langle y_t, Y_{t,p}^- \rangle \langle Y_{t,p}^-, Y_{t,p}^- \rangle^{-1} = \left[\hat{\phi}_1(p)\ \hat{\phi}_2(p)\ \ldots\ \hat{\phi}_p(p) \right]$. Let

$$\beta_{1,p} = \mathbb{E} y_t (Y_{t,p}^-)' [\mathbb{E} Y_{t,p}^- (Y_{t,p}^-)']^{-1} = \mathcal{H}_{1,p} (\Gamma_p^-)^{-1} = \left[\phi_1(p)\ \phi_2(p)\ \ldots\ \phi_p(p) \right].$$

Then Lemma A.5 implies that $\|\beta_{1,p} - [\Phi]_p\|_{Fr} = o(1)$ where $\sup_p \|[\Phi]_p\|_{Fr} < \infty$ due to the square integrability of the AR coefficients. This implies $\sup_p \|\beta_{1,p}\|_{Fr} < \infty$. Since $\|\gamma_l\|_{Fr} \le \kappa l^{2d_+-1}$, it follows that $\sup_p \|\mathcal{H}_{1,p}\|_{Fr} < \infty$ for $d_+ < 0.25$.

Further, note that $\|(\hat{\Gamma}_p^-)^{-1}\|_2 \le \|(\hat{\Gamma}_p^-)^{-1} - (\Gamma_p^-)^{-1}\|_2 + \|(\Gamma_p^-)^{-1}\|_2 = o(1) + \|(\Gamma_p^-)^{-1}\|_2$. Then $\hat{\beta}_{1,p} - \beta_{1,p} =$

$$\begin{aligned} \hat{\mathcal{H}}_{1,p}(\hat{\Gamma}_p^-)^{-1} - \mathcal{H}_{1,p}(\Gamma_p^-)^{-1} &= (\hat{\mathcal{H}}_{1,p} - \mathcal{H}_{1,p})(\hat{\Gamma}_p^-)^{-1} + \mathcal{H}_{1,p}[(\hat{\Gamma}_p^-)^{-1} - (\Gamma_p^-)^{-1}] \\ &= (\hat{\mathcal{H}}_{1,p} - \mathcal{H}_{1,p})(\hat{\Gamma}_p^-)^{-1} - \beta_{1,p}[\hat{\Gamma}_p^- - \Gamma_p^-](\hat{\Gamma}_p^-)^{-1}. \end{aligned}$$

From these equations (for $d_+ > 0$, we can use the third equation, else the second can be used) in combination with Lemma A.4 (where the norm bounds hold uniformly in the lag length), the result follows. □

Consequently, $p \to \infty$ at the rate given in the theorem implies that for $\hat{\phi}(z) = \sum_{j=0}^{p} \hat{\phi}_j z^j$ and $\phi_p(z) = \sum_{j=0}^{p} \phi_j(p) z^j$ it follows that $\|\hat{\phi}(z) - \phi_p(z)\|_2 \to 0$.

Furthermore, the norm bound implies that

$$\sum_{j=p+1}^{\infty} \|\phi_j\|^2 \leq M_\phi \sum_{j=p+1}^{\infty} j^{-2-2d_-} = O(p^{-1-2d_-})$$

and hence the Fourier series $\sum_{j=0}^{p} \phi_j z^j$ converges in L_2 to $\phi(z)$, that is, $\|\phi_p(z) - \phi(z)\|_2 \to 0$ and thus $\|\hat{\phi}(z) - \phi(z)\|_2 \to 0$.

These two lemmas show that subspace methods with the maximal choice of the order $n = ps$ deliver consistent estimates of the inverse transfer function $\phi(z)$ and thus also of the transfer function.

Finally, some facts on the approximation error for $n < ps$ are provided.

Lemma A.7 *Let $(\hat{A}, \hat{C}, \hat{K})$ denote a system with corresponding state $\hat{x}_t$ such that $\langle \hat{x}_t, \hat{x}_t \rangle = I_N$ and innovation noise variance $\hat{\Omega}$. Then consider the partitioning of the system as (where $\hat{A}_{11} \in \mathbb{R}^{n \times n}$, $\hat{K}_1 \in \mathbb{R}^{n \times s}$, $\hat{C}_1 \in \mathbb{R}^{s \times n}$)*

$$\hat{A} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix}, \quad \hat{K} = \begin{bmatrix} \hat{K}_1 \\ \hat{K}_2 \end{bmatrix}, \quad \hat{C} = \begin{bmatrix} \hat{C}_1 & \hat{C}_2 \end{bmatrix}.$$

(i) The system $(\hat{A}_n, \hat{C}_n, \hat{K}_n)$ obtained from using $\hat{x}_{t,n} = [I_n, 0]\hat{x}_t$ in the regressions in the last step of the CVA *algorithm fulfils*

$$\hat{A}_n = \hat{A}_{11}, \quad \|\hat{K}_n - \hat{K}_1\|_{Fr} = O(\|\hat{C}_2\|_{Fr}(\sqrt{n}\|\hat{C}_2\|_{Fr} + \|\hat{A}_{1,2}\|_{Fr})), \quad \hat{C}_n = \hat{C}_1.$$

(ii)$\|\hat{C}_{:,j}\|_2 = O\left(p^{\tilde{d}_+}\hat{\sigma}_j\right)$ where $\hat{C}_{:,j}$ denotes the jth column of $\hat{C}$.

(iii) Furthermore, let $\|f(z)\|_\infty = \sup_{\omega \in [0,2\pi]} \|f(e^{i\omega})\|_2$ and use the notation $\hat{k}(z) = I_s + z\hat{C}(I - z\hat{A})^{-1}\hat{K}$, $\hat{k}_{n1}(z) = I_s + z\hat{C}_n(I - z\hat{A}_n)^{-1}\hat{K}_1$. Then $\|\hat{k}(z) - \hat{k}_{n1}(z)\|_\infty$

$$\leq \|\hat{C}_2 + \hat{C}_1(zI_n - \hat{A}_{11})^{-1}\hat{A}_{12}\|_\infty \|(zI - \hat{A}_{22} - \hat{A}_{21}(zI_n - \hat{A}_{11})^{-1}\hat{A}_{12})^{-1}(\hat{K}_2 + \hat{A}_{21}(zI_n - \hat{A}_{11})^{-1}\hat{K}_1)\|_\infty.$$

(iv) Consequently, using $\hat{k}_n(z) = I_s + z\hat{C}_n(I - z\hat{A}_n)^{-1}\hat{K}_n$, we obtain

$$\|\hat{k}(z) - \hat{k}_n(z)\|_\infty \leq \|\hat{k}(z) - \hat{k}_{n1}(z)\|_\infty + \|\hat{C}_n(I - z\hat{A}_n)^{-1}\|_\infty \|\hat{K}_n - \hat{K}_1\|.$$

Proof (i) Reference [4] shows that $C_n = C_1, \Omega_n = \Omega + C_2C_2', A_n = A_{11}, K_n = (M_1 - A_{11}C_1')\Omega_n^{-1}$ where $M_1 = K_1\Omega + [I_n, 0]AC'$. Thus $\|\Omega_n - \Omega\| = \|C_2\|^2$. Therefore

$$K_n\Omega_n - K_1\Omega = A_{1,2}C_2' \Rightarrow (K_n - K_1)\Omega_n = A_{1,2}C_2' - K_1(\Omega_n - \Omega).$$

Since $\|\Omega^{-1}\| \leq M$, $\|K_1\| \leq \sqrt{n}$, $\|A_{1,2}\| \leq \sqrt{n}$ and $\Omega_n \geq \Omega$, it thus follows that

$$\begin{aligned} \|K_n - K_1\| &= \|[K_1(\Omega - \Omega_n) + A_{1,2}C_2']\Omega_n^{-1}\| \leq \|K_1C_2\|\|\Omega^{-1}\|\|C_2\| + \|C_2\|\|A_{1,2}\|\|\Omega^{-1}\| \\ &= \|\Omega^{-1}\|\|C_2\|(\|K_1C_2\| + \|A_{1,2}\|) \leq \|\Omega^{-1}\|\|C_2\|(\|C_2\|\sqrt{n} + \sqrt{n}). \end{aligned}$$

It is straightforward to show that this also holds for the estimates as only orthogonality relations are used here.

(ii) Consider the estimation of the state as $\hat{x}_t = \hat{V}'(\hat{\Gamma}_p^-)^{-1/2}Y_{t,p}^-$ which implies that $\langle \hat{x}_t, \hat{x}_t \rangle = I_{ps}$. According to Lemma A.4 $\|\hat{\Gamma}_f^+ - \Gamma_f^+\|_{Fr} \to 0$ if $pR_T(\tilde{d}_+) \to 0$. Then $\|\hat{\Gamma}_f^+\|_2 \leq \|\hat{\Gamma}_f^+ - \Gamma_f^+\|_2 + \|\Gamma_f^+\|_2 = O(p^{2\tilde{d}_+})$. Furthermore,

$$\hat{C}_2 = \hat{\beta}_{f,p}\hat{V}_2 = [I_s, 0](\hat{\Gamma}_f^+)^{1/2}\hat{U}\hat{\Sigma}\hat{V}'\hat{V}[0, I]' = [I_s, 0](\hat{\Gamma}_f^+)^{1/2}\hat{U}_2\hat{\Sigma}_2,$$

showing that the two norm of the jth column of $\hat{C}_2$ is of order $O(p^{\tilde{d}_+}\hat{\sigma}_{n+j})$.

(iii) and (iv) follow from straightforward computations. □

References

1. Bauer, D.: Estimating linear dynamical systems using subspace methods. Econ. Theory **21**, 181–211 (2005)
2. Bauer, D., Wagner, M.: A canonical form for unit root processes in the state space framework. Econ. Theory **6**, 1313–1349 (2012)
3. Chan, N.H., Palma, W.: State space modeling of long-memory processes. Ann. Stat. **26**, 719–740 (1998)
4. Dahlen, A., Scherrer, W.: The relation of the CCA subspace method to a balanced reduction of an autoregressive model. J. Econ. **118**, 293–312 (2004)
5. Galbraith, J.W., Zinde-Walsh, V.: Autoregression-based estimators for ARFIMA models. In: CIRANO Working Papers 2001–11 (2001)
6. Hannan, E.J., Deistler, M.: The Statistical Theory of Linear Systems. Wiley, New York (1988)
7. Hosking, J.R.M.: Fractional differencing. Biometrika **68**(1), 165–176 (1981)
8. Hosking, J.R.M.: Asymptotic distributions of the sample mean, autocovariances, and autocorrelations of long-memory time series. J. Econ. **73**(1), 261–284 (1996)
9. Larimore, W.E.: System identification, reduced order filters and modeling via canonical variate analysis. In: Proceedings of the 1983 American Control Conference, Piscataway, NJ, pp. 445–451 (1983)
10. Lewis, R., Reinsel, G.: Prediction of multivariate time series by autoregressive model fitting. J. Multivar. Anal. **16**, 393–411 (1985)
11. Palma, W., Bondon, P.: On the eigenstructure of generalized fractional processes. Stat. Probab. Lett. **65**, 93–101 (2003)
12. Poskitt, D.S.: Autoregressive approximation in nonstandard situations: the fractionally integrated and non-invertible case. Ann. Inst. Stat. Math. **59**, 697–725 (2006)
13. Sela, R.J., Hurvich, C.M.: Computationally efficient methods for two multivariate fractionally integrated models. J. Time Ser. Anal. **30**(6), 631–651 (2009)

Robust Forecasting of Multiple Yield Curves

Christoph Gerhart, Eva Lütkebohmert and Marc Weber

Abstract In this paper, we develop robust methods for forecasting term structures of interest rates. We implement a deep long short-term memory (LSTM) neural network based on keras. Our input data is based on the bootstrapped bid, mid and ask multiple (tenor-dependent) yield curves reflecting different risk categories over the period 2005–2018. We use the bid-ask spreads as an additional input factor modelling the market depth. Since there is only a limited amount of data available, there is a lack of a sufficiently large training data set. We cope with that difficulty by generating data based on fitted time series models in order to enlarge the training data. Furthermore, we apply support vector machines to predict trends in the term structures. For this approach, we include different market variables to investigate the relationship of these quantities to future yields.

Keywords Forecasting of yield curves · Multiple term structures · Machine learning · Neural networks · Support vector machines

Introduction

The term structure of interest rates (or yield curve), that describes the interest rate as a function of maturity, represents an important tool for derivative pricing, risk

C. Gerhart · E. Lütkebohmert
Department of Quantitative Finance, Institute for Economic Research, University of Freiburg,
Platz der Alten Synagoge 1, KG II, 79098 Freiburg im Breisgau, Germany
e-mail: christoph.gerhart@finance.uni-freiburg.de
URL: https://www.finance.uni-freiburg.de

E. Lütkebohmert (✉)
Freiburg Institute for Advanced Studies (FRIAS), University of Freiburg, Albertstraße 19,
79104 Freiburg im Breisgau, Germany
e-mail: eva.luetkebohmert@finance.uni-freiburg.de
URL: https://www.finance.uni-freiburg.de

M. Weber
Department for Mathematical Stochastics, Mathematical Institute, University of Freiburg,
Ernst-Zermelo-Straße 1, 79104 Freiburg im Breisgau, Germany
e-mail: marc.weber92@hotmail.com

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_13

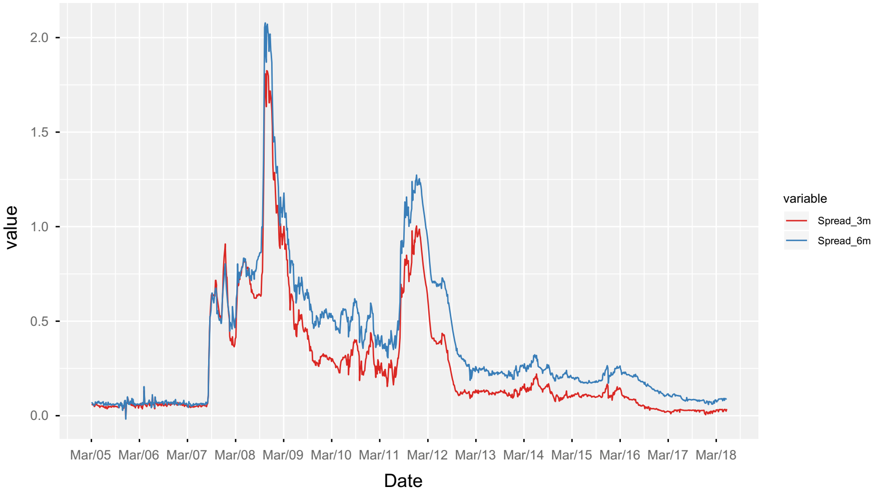
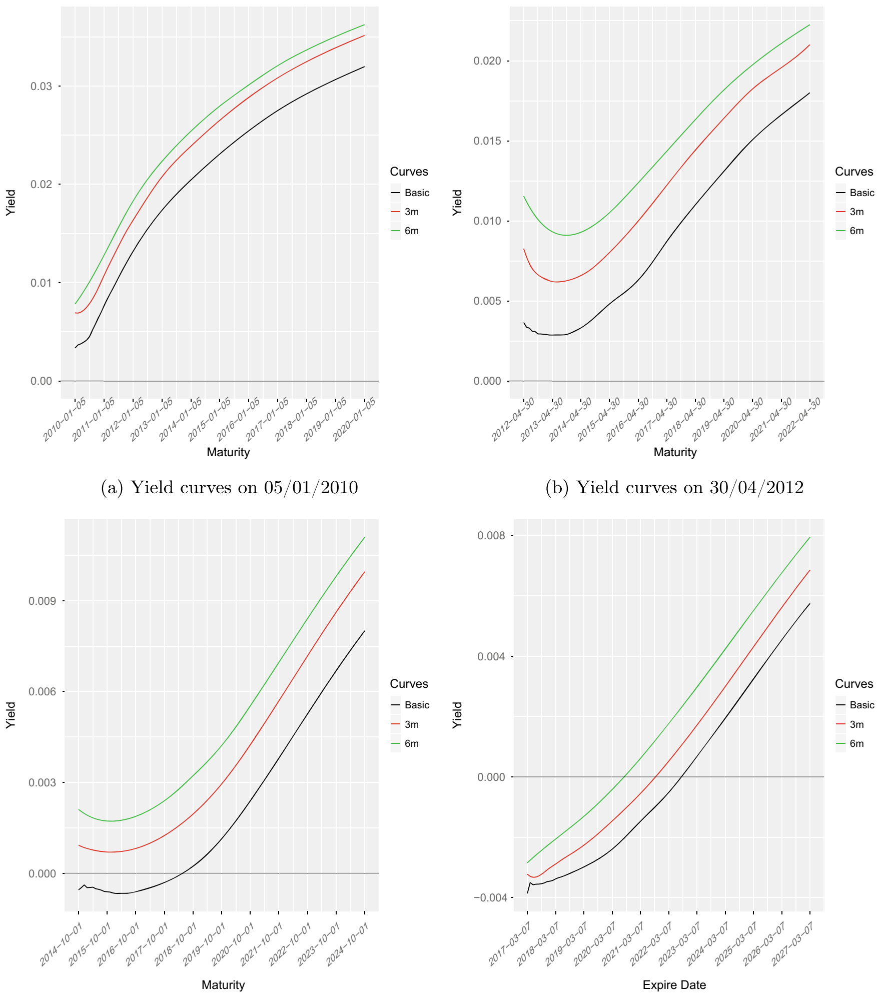


Fig. 1 EURIBOR-OIS spreads. The figure shows the spread between 3-month EURIBOR rates and OIS rates (red curve), respectively, between 6-month EURIBOR and OIS rates (blue curve) over the time period 2005–2018

management and monetary policy, and hence, has been intensively studied in the literature. In the last years, following the financial crisis 2007/2008, however, there has been a major change in interest rate markets. While interest rates of the same maturity showed certain consistencies before the crisis, giving rise to a single term structure, this no longer holds in post-crisis markets. Instead, e.g., rates on swaps with the same maturity can differ quite substantially depending on the tenor of the underlying reference rate. A famous example is the spread between interbank rates of a specific tenor (say 3 months) and the corresponding rates on Overnight Indexed Swaps (OIS) with the same maturity. As can be seen in Fig. 1, the spread was almost negligible before the crisis but then started to diverge substantially, especially after the default of Lehman Brothers. In particular, these spreads also stayed at high levels after the crisis.

Thus, in the post-crisis setting the term structure of interest rates becomes tenor-dependent reflecting different risk categories such as, for instance, different levels of credit and liquidity risk. Figure 2 shows such multiple yield curves at different points in time. We want to point out here that yield curves of different tenors are not just parallel shifts of the discount (or basic) curve since spreads over the discount curve can be quite different at the short- and at the long-end of the term structure (see, e.g., panel (b) in Fig. 2) and are fluctuating over time. Moreover, the figure indicates that yield curves are monotonically increasing in tenor length.

In this paper, we take these new characteristics of interest rate markets into account and develop robust methods for forecasting of multiple (tenor-dependent) yield curves based on neural networks. Our input data consists of bootstrapped bid, mid and ask yields over the time period 2005–2018. More specifically, we consider



(a) Yield curves on 05/01/2010

(b) Yield curves on 30/04/2012

(c) Yield curves on 01/10/2014

(d) Yield curves on 07/03/2017

Fig. 2 Tenor-dependent term structures of interest rates at different points in time

the time series of historical discount, three-month and six-month yield curves for maturities up to ten years (as shown for specific dates in Fig. 2). We implement a deep long short-term memory (LSTM) neutral network based on keras to predict future yields. Our results show extremely accurate predictions of 1-day, 1-month, 3-month and 6-month ahead yields across all maturities and curves. In particular, our results point out the importance of cross-tenor dependencies for predicting future yields, a feature that is naturally missing in existing single-curve approaches. Moreover, we show that using bid-ask spreads as additional inputs, reflecting a measure of market depth, increases the accuracy of yield curve predictions. Additionally, we apply support vector machines (SVMs) to predict trends in the term structure over a time horizon of up to 15 weeks. Here, our results for the SVM classifier show a very accurate and extremely robust performance compared to various classification benchmark methods.

Standard forecasting methods based, e.g., on time series models insinuate some specific distributional behavior and set the corresponding parameters during the training process. In contrast to such an approach, the methodology suggested in this paper can be considered as much more robust as it does not impose any distributional assumptions. Moreover, if a major change in the evolution of the time series occurs, e.g. a level shift due to some policy changes, standard models usually fail to adapt quickly to the new environment. Our approach, however, adapts quickly to such scenarios. In this sense, we consider our neural network based forecasting method as a very robust and highly efficient approach to forecast future yields.

The paper is structured as follows. In section "Data" we describe our data set. The forecasting methodology and results based on neural networks is discussed in section "Forecasting of Multiple Yield Curves via Neural Networks". Here, we also elaborate on how we dealt with the problem of limited available data. In section "Trend Forecasting via Support Vector Machines" we discuss trend forecasting based on support vector machines. The final section concludes.

Data

Our data set consists of daily bootstrapped multiple yield curves over the period 2005–2018. More specifically, we consider historical discount, three-month and six-month curves for maturities up to ten years. The discount curves have been bootstrapped from market data on overnight indexed swap (OIS) rates for maturities ranging from one week to 10 years.[1] For the construction of the risky (tenor-dependent) yield curves, market quotes of deposit rates, rates from forward rate agreements (FRA) and swap rates were used. More explicitly, the short-end is constructed from deposit rates for contracts where the tenor agrees with the maturity, i.e., 3 month deposits for the 3 month curve etc. Besides, FRAs for periods starting in one month ending in 4 months are used for the construction of the 3 month curve while the 6 month curves are constructed from market data on FRAs for time periods [1m, 7m] and [3m, 9m]. Furthermore, the medium to long-term rates are derived from market data on tenor-dependent interest rate swaps with maturities ranging from 6 months up to 10 years. For each term structure, we derived the bid, mid and ask curves from the corresponding quotes of the bootstrapping instruments. We use European market data provided by Bloomberg for the time period from September, 2005, until May, 2018, on a daily basis. The (tenor-dependent) term structures have been constructed by an

[1] The European market publishes the swap rate of an OIS at every business date for maturities ranging from 1 week to 60 years. The floating leg is indexed on the EONIA rate and the payments are based on annual frequency.

exact fit bootstrapping methodology as explained in [12]. Results for specific dates are displayed in Fig. 2. Additionally, the term structures were fitted to the parametric Nelson Siegel model introduced in [19]. Below, we forecast yields over different time horizons, and we work with the convention that one week consists of 5 trading days and one months of 21 trading days.

Besides, we include different macroeconomic variables in our data set in order to improve the forecasting performance of our methodology. Here, we use prime interest rates which are available at quarter annual frequency from the official website of the European Central Bank. In addition, we include quarter annual GDP data and monthly inflation rates which we extracted from Bloomberg. The time period of all macroeconomic variables ranges from September 2005 to March 2018.

Forecasting of Multiple Yield Curves via Neural Networks

In order to predict future yields of various maturities and tenors and over different forecasting horizons, we implemented a neural network which we present in the following subsection. Thereafter, we discuss how we dealt with the problem of limited available data, present our forecasting results and elaborate on alternative approaches.

Neural Network Design and Methodology

The neural network is based on the `keras` package with `tensorflow` backend. The input layer consists of the mid, bid and ask yields of a specific maturity, the corresponding bid-ask spreads as well as the basis spreads calculated as the difference between the mid values of the risky tenor-dependent yields and the risk-free discount yields for the specified maturity. The output consists of the yields themselves. As we follow a multiple curve approach, this results in a rather large input and output dimension. We use the eminent relationship between the risky and the discount curve in order to fine tune the training process of the neural network. Furthermore, the bid-ask spread is included in order to teach the network about market depth.

We ran a hyperparameter optimization code using the `R` package `flags` which searches for the best hyperparameter combination on a discrete grit. Afterwards, we also implemented the whole code in `Python` and used the hyperparameter optimization procedure based on the `hyperas` package, which is more advanced. For a deeper insight on grid searches versus random searches, we refer to [3]. The network architecture based on the hyperparameter optimization consists of three layers and two dropout layers in between. The first layer is an LSTM layer with 128 neurons. The next layers are given by a second LSTM layer with 32 neurons and a third dense layer. The drop out rates are set to 0.195327 and 0.056875 while the learning rate is chosen as $8.777485 \cdot 10^{-4}$ and is allowed to decay over time.

We chose 3 neurons in the output layer since we analyse only one maturity at a time for three different curves (discount, three-month and six-month curve) and the last layer is supposed to directly correspond to the number of outputs. For a deeper insight on Long-Short-Term-Memory neural networks, we refer the reader to [13] and emphasize the ability to store and forget important, respectively redundant, information of those layers.

The LSTM layer in keras needs a 3-dimensional input shape. It is, therefore, necessary to reshape the given dataset in tensor form. The first dimension corresponds to the number of historical dates used as a training, respectively test, data set. The last dimension agrees with the number of input, respectively output parameters. In our case, this amounts to 14 parameters—one fixed maturity times three curves and three different values (mid, ask, bid) already yields 9 dimensions. When adding three bid-ask spreads and 2 basis spreads, one ends up with 14 input parameters. The second dimension specifies how much data is used to be fitted to one output unit. Here, we chose chunks of 300 data points meaning that the last 300 values are used to find a connection to the next value of interest. For monthly forecasts, this would, e.g., be the 322nd value.

Hence, the input shape in the LSTM layer needs to be of the form `c(300,14)` in our case. As explained above, a three dimensional array needs to be passed to the network. As one can see in the keras documentation, the three dimensions coincide with `(batch_size, sequence_length, features)`. The first dimension does not need to be passed along. It only defines how long the training process is going to be. The reshaping procedure of the two dimensional data is accomplished via a loop. Starting with the matrix consisting of the first 300×14 data points, one overwrites the first entry in the first dimension of a dummy tensor. In the next step, one starts at the second value and proceeds to the 301st value. The procedure is repeated over the whole training data set. The output data then consists of one-day-ahead forecasts. If more than one forecast shall be constructed with each 300 data point chunk, then the output needs to be reshaped correspondingly.

As activation function in the first layers we chose the tangent hyperbolic as the sigmoid function produced larger errors. In the output layer, we use the linear activation function. Usually a relu function is chosen in the last layer. However, that does not correspond well with our chosen normalizing procedure as we transformed the data set into one which values range from -1 to 1. Therefore, a relu function would just cut off roughly half of the values and set them to 0. A linear activation is the natural choice in that context.

Based on the above specified architecture of the network, we predict yields of different maturities and tenors. As in [12] we use the root-mean-squared-error as loss function.[2] The optimizer chosen in our model is the Adam optimizer [15]. Furthermore, we chose a validation split of 0.2 and a batch size of 32. The calculation is done via 100 epochs.

[2] We did not use a pre-implemented loss of keras as most of these functions are applicable only for categorization problems, and hence, fail to be of any value for our application.

Dealing with Limited Data

As there is only a limited data set available, we face the problem that running a small amount of epochs yields bad results but choosing a large amount of epochs often results in over-fitting. We cope with that problem by generating new data based on the historical yields via the following procedure. We use the first 500 historical data points of the training set on the 9 yields (bid, mid and ask yields for three tenors and fixed maturity) and fit an ARIMA(1, 0, 1) model to these time series. Based on the time series model, we then predict intervals for the next ten values and save the interval boarders. Then, we proceed with the points eleven up to 510 to predict the next ten intervals and so on. We end up with 1000 confidence intervals in which our yields are likely to lie in. In this way, we can easily generate new paths of the nine time series available by randomly choosing values in these intervals. Now, one has an arbitrary large training data set to which we can fit the neural network. We still have to cope with the risk of over-fitting, though, since the generated paths are highly dependent on the true evolution. Enlargement procedures are very common coping with meager data sets. Usually, small geometric transformations are used to do so. See, e.g., [6] who significantly improved their performance on the MNIST data set due to enlargement by adding translations, rotations and line thickness changes on the underlying pictures. Obviously, a linear transformation such as that one is not applicable to time series models. Hence, the procedure we chose seems most appropriate.

Results

The neural network performs very well with highly accurate predictions of future yields across all tenors and maturities over various forecasting horizons. In Tables 1, 2, 3 and 4 we list the results of that approach for 1-day, 1-month, 3-month and 6-month ahead predictions. The first row in each table shows the RMSEs of the predicted yields for the discount, 3-months and 6-months curve for maturities in 1 year and 3 years, respectively, over the specified forecasting horizons for the neural network (NN). The second row lists the corresponding RMSE for the random walk (RW) approach. RMSEs of the best performing approach are displayed in bold face. Rows 3 and 4 show the correlation between the predicted values and the actual values which indicates the robustness of a forecasting method.

Our results show that the neural network produces extremely accurate predictions of future yields across all maturities and curves and for all forecasting horizons. In particular, for longer term predictions as 3- or 6-month ahead forecasts, our results clearly outperform the benchmark random walk approach which has been proven to be hard to beat in the related literature. For 1-month ahead predictions, the neural network still beats the random walk procedure for yields with longer maturities. For very short-term predictions such as 1-day ahead forecasts, the random walk approach

Table 1 The table shows the six-month ahead forecasting results of the neural network and the random walk approach as well as the correlation of the forecasts to the true values

Method	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
NN	**0.002106980**	**0.002221245**	**0.002320099**	**0.002087555**	**0.002096353**	**0.002189963**
RW	0.002455532	0.002646123	0.002898431	0.002351593	0.002598185	0.002784662
Correlation NN	0.873266651	0.879881245	0.880366621	0.879653312	0.874569823	0.890000451
Correlation RW	0.851333642	0.861110326	0.871005612	0.862343540	0.864706349	0.876448767

Table 2 The table shows the three-month ahead forecasting results of the neural network and the random walk approach as well as the correlation of the forecasts to the true values

Method	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
NN	**0.001256644**	**0.001198885**	**0.001400062**	**0.001075632**	**0.001101326**	**0.001178669**
RW	0.001344568	0.001399984	0.001506844	0.001266431	0.001395539	0.001470158
Correlation NN	0.938855486	0.93713645	0.93941100	0.94000325	0.93972566	0.93910442
Correlation RW	0.93225647	0.93089991	0.93222241	0.93772375	0.93596972	0.93643050

Table 3 The table shows the one-month ahead forecasting results of the neural network and the random walk approach as well as the correlation of the forecasts to the true values

Method	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
NN	0.000745689	0.000678933	**0.000580112**	**0.000889645**	**0.000898752**	**0.000911564**
RW	**0.000587419**	**0.000582809**	0.000583411	0.000913224	0.000955568	0.000990582
Correlation NN	0.841000726	0.866559841	0.880076524	0.902354446	0.920068596	0.919167822
Correlation RW	0.838753128	0.866618380	0.877812199	0.894634916	0.908987169	0.909525220

performs better than the neural network. However, it should be mentioned here that the RMSEs of both methods are extremely small for all maturities and curves in this case, so that both approaches actually produce very precise forecasts of next day yields. The correlation values verify the robustness of our forecasting methodology.

We also tested several other forecasting approaches that we want to discuss here without stating the results explicitly in order to keep the exposition compact. These alternative methods include a version where yields of various maturities are turned over to the network as input. However, that approach failed to perform well. An explanation for this observation is the high dimensionality of the input and output data in proportion to the number of historical dates in that case. As it is explained in [11] a large number of input dimensions corresponds rather poorly with small data

Table 4 The table shows the one-day ahead forecasting results of the neural network and the random walk approach as well as the correlation of the forecasts to the true values

Method	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
NN	0.0003088099	**0.00017444**	0.000386521	0.000333298	0.000376663	0.000361211
RW	**0.000215763**	0.000225456	**0.000215041**	**0.000215763**	**0.000225456**	**0.000215041**
Correlation NN	0.988765231	0.990015202	0.991645551	0.989822135	0.990041124	0.990117699
Correlation RW	0.991047061	0.993272731	0.994688241	0.992894395	0.993695834	0.994688241

sets. The latter limits the number of training runs such that the neural network is not able to learn which data has to be considered important and which redundant in view of every single output dimension. At this point we want to emphasis that even though a larger input dimension appears to be troublesome, the inclusion of the bid and ask quotes really improves the performance of the neural network. The information which lies in the spread obviously corresponds to the evolution of the time series which is well detected by the artificial intelligence.

Besides, we implemented an alternative network based on a parameterized plug-in model. For a given training set, we extract the Nelson–Siegel yield curve parameters β_0, β_1 and β_2 and assume these to follow a certain time series model. As input we provide the three parameters for every curve and mid, bid and ask values which results in a total of 27 input parameters. After training the network, we build forecasts for the Nelson–Siegel parameters that are then used to compute future yields via the Nelson–Siegel model. Our results showed that this did not yield any improvements over the approach that is directly based on the yields. However, as documented in [12] the dynamic Nelson Siegel approach of [8] where parameters are modeled as a time series process is in general not able to outperform the random walk benchmark. Using neural networks, one only observes slight improvements compared to that approach, however, these do not suffice to yield desirable results. The inaccuracies of using the Nelson Siegel model pass along.

Furthermore, we have included different macroeconomic variables such as the prime interest rate, inflation rates and GDP growth rates, in the input layer as these have been shown to improve predictions in standard time series based approaches (compare e.g. [1, 7, 10, 16, 18]). In our neural network, this, however, produces less precise forecasts. An explanation for this might be the seldom updates of those values. For example, GDP is updated only quarter annually while our yields are updated on daily basis. Thus, between two updating time points, we have fixed the macroeconomic variables to the value of the last update. In this way, the macroeconomic variables represent an extra dimension in the network but only provide additional value for forecasting at the updating time points. The inclusion of such macroeconomic variables may, however, improve the model in a way which is not observable when looking at the loss function as defined above. When fixing the macroeconomic variables to the value of the last update, the increments in these variables are

Table 5 The table shows the RMSE of the one-month ahead forecasting results of the neural network including macroeconomic variables relative to the RMSEs of the benchmark neural network at the updating dates of those values. The column "Frequency" specifies whether one takes monthly or quarter annually updates to extract the results

Method	Frequency	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
Macro variables included	Monthly	0.95	0.93	0.92	0.91	0.96	0.92
Macro variables included	Quarter annually	0.89	0.90	0.87	0.85	0.92	0.90

obviously equal to zero, when no update has occurred. To overcome the problem that the redundant information during most of the time points just prevents the network to train quickly, one needs to adapt the activation function in a manner, that the additional layers are not being activated when their increments are equal to zero. That is easily accomplished by using an indicator function as a multiplicative factor to the original activation function. Such a network seems to produce roughly the same results as the original ones, since the training process is seldomly interfered by the new architecture. However, when comparing the results at the days when the macroeconomic variables are updated, a major improvement is observable. This is documented in Tables 5, 6 and 7 corresponding to the forecasting horizons of one-month, three-month and six-month, respectively. Here, we extracted exactly those dates, at which an update of macroeconomic variables occurred. Since the GDP rates as well as the prime interest rate are only updated quarter annually whereas the inflation rates are updated monthly, we studied both updating frequencies separately. That is, in the first row corresponding to monthly frequency, only inflation rates are included while in the second row, where updates are quarter annually, all three macroeconomic variables are taken into account. Results are reported relative to the RMSEs in the benchmark NN without macroeconomic variables. As one can easily see, the adjusted neural network performs better than the benchmark NN at those dates and these improvements tend to become more prevalent the larger the forecasting horizon is. An interpretation of these results is, that major changes in interest rate markets take place, when macroeconomic quantities are being updated. Those changes particularly affect the remote future since markets only adapt slowly. Our conclusion is, that including macroeconomic variables is an efficient way to improve the forecasting method and to make forecasts more robust since considerable changes—perhaps even so far unforeseeable jumps—are easier to detect.

Stacked Generalization

Another way to make the model more robust is to simply combine a variety of different models. This procedure is commonly known in the literature as stacking and was first introduced by [22]. For further details, see [11]. The main idea is to train

Table 6 The table shows the RMSE of the three-month ahead forecasting results of the neural network including macroeconomic variables relative to the RMSEs of the benchmark neural network at the updating dates of those values. The column "Frequency" specifies whether one takes monthly or quarter annually updates to extract the results

Method	Frequency	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
Macro variables included	Monthly	0.91	0.89	0.90	0.89	0.93	0.92
Macro variables included	Quarter annually	0.84	0.80	0.81	0.83	0.87	0.81

Table 7 The table shows the RMSE of the six-month ahead forecasting results of the neural network including macroeconomic variables relative to the RMSEs of the benchmark neural network at the updating dates of those values. The column "Frequency" specifies whether one takes monthly or quarter annually updates to extract the results

Method	Frequency	Discount-1y	3m-1y	6m-1y	Discount-3y	3m-3y	6m-3y
Macro variables included	Monthly	0.84	0.80	0.78	0.80	0.80	0.81
Macro variables included	Quarter annually	0.80	0.76	0.77	0.79	0.77	0.80

a variety of different models and use a meta-model, which combines the outputs of the basic model in order to build the overall forecasts. If one has managed to fine tune each of the corresponding models carefully, the stacked model usually profits from the wide range on which it is build. Furthermore, the meta-learner is most commonly chosen to be relatively easy [21]. In our case, we used a weighted linear combination on the meta level and different trained neural networks as basic models. The network we presented in detail as well as two plug-in neural networks—the first based on the Nelson–Siegel model, the second based on principal component analysis—were part of the stacked generalization. Besides, we also added a simpler version of the first neural network, which means, we reduced the number of neurons in the first layer to 64 and used dense layers instead of LSTM layers.

The stacked generalization failed to outperform the presented approach. Our intuition is that the basic models are too similar in order for the overall model to significantly profit from the different models that are incorporated. Usually, when using stacking one chooses different model classes which have many different characteristics. However, we solely used neural networks so far. Including different model classes in order to build a wider ranged stacked generalization based on machine learning tools is left for future research. In this way, we anticipate that our current results may be further improved and may become even more robust, since a variety of models is less susceptible to unforeseen changes, which require updating.

Trend Forecasting via Support Vector Machines

In this section, we investigate the relationship between the different movements of the term structures and various market variables by using Support Vector Machines (SVMs). This type of machine learning classifier can be applied to forecast the up and down trends of the levels as well as the slopes for the multiple term structures of interest rates. For a detailed treatment of the subject of SVMs, we refer to [4, 5, 9]. Furthermore, an introductory discussion of SVMs is given by [2] and a more rigorous study is stated by [20]. SVMs have been successfully applied for trend predictions in financial markets. For instance, [14] analyze the accuracy of predictions of stock market movements with SVMs by comparing their performance with classical forecasting classification models. Their SVM-based models outperform the other classification methods in their empirical analysis. In the following, we apply support vector machines to predict level and slope trends in term structures of interest rates.

Model Input Selection

We consider the bootstrapped discount, three-month and six-month yield curves for the period of September 2005 to May 2016. For each term structure we derived the bid, mid and ask curves up to a maturity of 10 years. The term structures are fitted to the parametric model of [19]. Consequently, we obtain a time series of estimated parameters of level $\beta_{0,t}^{q,k}$, slope $\beta_{1,t}^{q,k}$, curvature $\beta_{2,t}^{q,k}$ and exponentially decaying rate $\lambda_t^{q,k}$ of the loadings where t represents the date, $q \in \{\text{bid}, \text{mid}, \text{ask}\}$ and $k \in \{\text{d}, 3\text{m}, 6\text{m}\}$. From the weekly changes of the estimated level and slope parameters, we derive the categorical variables indicating the up or down movement of the yield curve shape parameters. As explanatory variables we use the fitted level and slope parameters of each curve, the changes in the level and slope of the preceding period, and the Euro STOXX 50. Moreover, we include the ratio between the bid-ask spread and the mid bond prices for the time-to-maturity of 10 years as a measure for the long-term liquidity in the market. All the raw data is provided by Bloomberg and corresponds to a weekly frequency (last business day in the week). The prediction models can then be expressed in form of

$$\text{level}_t = F(\Lambda_{t-1}^l) \quad \text{and} \quad \text{slope}_t = G(\Gamma_{t-1}^s)$$

where

$$\text{level}_t = \begin{pmatrix} \text{level}_t^{\text{d,ask}} \\ \text{level}_t^{\text{d,mid}} \\ \text{level}_t^{\text{d,bid}} \\ \text{level}_t^{\text{3m,ask}} \\ \text{level}_t^{\text{3m,mid}} \\ \text{level}_t^{\text{3m,bid}} \\ \text{level}_t^{\text{6m,ask}} \\ \text{level}_t^{\text{6m,mid}} \\ \text{level}_t^{\text{6m,bid}} \end{pmatrix}, \quad \text{slope}_t = \begin{pmatrix} \text{slope}_t^{\text{d,ask}} \\ \text{slope}_t^{\text{d,mid}} \\ \text{slope}_t^{\text{d,bid}} \\ \text{slope}_t^{\text{3m,ask}} \\ \text{slope}_t^{\text{3m,mid}} \\ \text{slope}_t^{\text{3m,bid}} \\ \text{slope}_t^{\text{6m,ask}} \\ \text{slope}_t^{\text{6m,mid}} \\ \text{slope}_t^{\text{6m,bid}} \end{pmatrix} \quad \varLambda_{t-1}^{l} = \begin{pmatrix} \varLambda_{t-1}^{\text{d,ask}} \\ \varLambda_{t-1}^{\text{d,mid}} \\ \varLambda_{t-1}^{\text{d,bid}} \\ \varLambda_{t-1}^{\text{3m,ask}} \\ \varLambda_{t-1}^{\text{3m,mid}} \\ \varLambda_{t-1}^{\text{3m,bid}} \\ \varLambda_{t-1}^{\text{6m,ask}} \\ \varLambda_{t-1}^{\text{6m,mid}} \\ \varLambda_{t-1}^{\text{6m,bid}} \end{pmatrix}, \quad \varGamma_{t-1}^{s} = \begin{pmatrix} \varGamma_{t-1}^{\text{d,ask}} \\ \varGamma_{t-1}^{\text{d,mid}} \\ \varGamma_{t-1}^{\text{d,bid}} \\ \varGamma_{t-1}^{\text{3m,ask}} \\ \varGamma_{t-1}^{\text{3m,mid}} \\ \varGamma_{t-1}^{\text{3m,bid}} \\ \varGamma_{t-1}^{\text{6m,ask}} \\ \varGamma_{t-1}^{\text{6m,mid}} \\ \varGamma_{t-1}^{\text{6m,bid}} \end{pmatrix}$$

with

$$\varLambda_{t-1}^{k,q} = \begin{pmatrix} \beta_{0,t-1}^{k,q} \\ \beta_{0,t-1}^{k,q} - \beta_{0,t-2}^{k,q} \\ \frac{B^{\text{bid},k}(t-1,10) - B^{\text{ask},k}(t-1,10)}{B^{\text{mid},k}(t-1,10)} \\ S_{t-1}^{\text{SXX50}} \end{pmatrix}^{\top}, \qquad \varGamma_{t-1}^{k,q} = \begin{pmatrix} \beta_{1,t-1}^{k,q} \\ \beta_{1,t-1}^{k,q} - \beta_{1,t-2}^{k,q} \\ \frac{B^{\text{bid},k}(t-1,10) - B^{\text{ask},k}(t-1,10)}{B^{\text{mid},k}(t-1,10)} \\ S_{t-1}^{\text{SXX50}} \end{pmatrix}^{\top}.$$

As mentioned above, $\text{level}_t^{k,q}$ and $\text{slope}_t^{k,q}$ are categorical variables indicating up and down movements for curve k and type of quote q.

Model Design

The input data consists of weekly quotes from September 2005 to May 2016. As training set we choose the first 150 weekly model quantities ranging from 2nd September 2005 to 12th February 2016. The test set contains 15 weeks belonging to the period 19th February 2016 to 27th May 2016. The implemented SVM for the level predictions is based on bound-constraint formulation of the classification. It is equipped with the Gaussian Radial Basis kernel function where the kernel parameter is set to $\sigma = 0.01965$. The constant C of the regularization term in the Lagrange formulation, known as cost of constraints violation, is set to be $C = 3$. Furthermore, a 3-fold cross validation on the training data is performed to assess the quality of the model performance. This type of classifiers also supports class-probabilities output. For the forecasting of the slope movements we use a similar design of the SVM with parameters $\sigma = 0.06555$, $C = 25$ and 3-fold cross validation. The parameters are obtained by using a hyperparameter optimization procedure.

Forecasting Results

Typically, the evaluation of the performance of classifiers is stated in form of accuracy measures or their comparison to other classification benchmark methods. A

Table 8 Comparison of model performance with positive class *up-movement* as well a positive class *down-movement* for level trends

Method	+–class	Accuracy	Error rate	Kappa	Sensitivity	Specificity	Precision	F-score
SVM	Up	**0.6593**	**0.3407**	**0.3164**	0.7246	**0.5909**	**0.6494**	0.6849
LDA	Up	0.6519	0.3481	0.2971	**0.8551**	0.4394	0.6146	**0.7152**
QDA	Up	0.6519	0.3481	0.2999	0.7681	0.5303	0.6310	0.6928
RW	Up	0.4254	0.5746	−0.1498	0.4412	0.4091	0.4348	0.4380
SVM	Down	**0.6593**	**0.3407**	**0.3164**	**0.5909**	0.7246	0.6724	**0.6290**
LDA	Down	0.6519	0.3481	0.2971	0.4394	**0.8551**	**0.7436**	0.5524
QDA	Down	0.6519	0.3481	0.2999	0.5303	0.7681	0.6863	0.5983
RW	Down	0.4254	0.5746	−0.1498	0.4091	0.4412	0.4154	0.4122

Table 9 Comparison of model performance with positive class *up-movement* as well a positive class *down-movement* for slope trends

Method	+–class	Accuracy	Error rate	Kappa	Sensitivity	Specificity	Precision	F-score
SVM	Up	**0.6000**	**0.4000**	**0.2059**	0.7077	**0.5000**	**0.5679**	0.6301
LDA	Up	0.5852	0.4148	0.1783	0.7231	0.4571	0.5529	0.6267
QDA	Up	0.5630	0.4370	0.1412	**0.8154**	0.3286	0.5300	**0.6424**
RW	Up	0.4104	0.5896	-0.1807	0.3846	0.4348	0.3906	0.3876
SVM	Down	**0.6000**	**0.4000**	**0.2059**	**0.5000**	0.7077	0.6481	**0.5645**
LDA	Down	0.5852	0.4148	0.1783	0.4571	0.7231	0.6400	0.5333
QDA	Down	0.5630	0.4370	0.1412	0.3286	**0.8154**	**0.6571**	0.4381
RW	Down	0.4104	0.5896	-0.1807	0.4348	0.3846	0.4286	0.4317

comprehensive overview on useful and common accuracy measures based on the confusion matrix can be found in [11, 17].

Table 8 shows the performance of our SVM approach for the prediction of up- and down movements in the level of interest rates and compares these with the corresponding values for the benchmark random walk model (RW), linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA). Table 9 illustrates the corresponding results for up- and down movements in the slope of the yield curves. The results indicate that our SVM classifier performs extremely robust across various classification benchmark methods and clearly outperforms them in most cases.

Conclusion

In this paper, we have applied methods based on neural networks to forecast interest rates of different maturities and tenors. Our results show extremely accurate predictions, in particular for longer term yields. Besides, we included bid-ask spreads as a measure of market depths in the input vector. The neural network is able to exploit this information and our results show a significant improvement of the network's performance. Further, we included different macroeconomic quantities in our analysis and showed, that at the updating dates of those values, the forecasting performance was substantially increased. This documents that our implemented neural network reflects a robust forecasting procedure, which, on the one hand, performs quite well over time and on the other hand is able to exploit also rarely updated market information to detect major changes in yields.

In addition, we have applied SVMs to forecast trends in the yield curve shape parameters. Compared to other classification methods, our SVM approach performs very well and robust. While the linear discriminant analysis also yields good results for level trends, it performs less well for slope predictions. In contrast, the quadratic discriminant analysis works well for slope trends but less good for level trends. The suggested SVM approach, however, is very stable and provides good trend predictions across the different yield curve movements.

References

1. Ang, A., Piazzesi, M.: A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables. J. Monet. Econ. **50**, 745–787 (2003)
2. Bennett, K.P., Campbell, C.: Support vector machines: hype or hallelujah? ACM SIGKDD Explor. Newsl. **2**, 1–13 (2000)
3. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**, 281–305 (2012)
4. Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. **2**, 121–167 (1998)
5. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**, 273–297 (1995)
6. Decoste, D., Schölkopf, B.: Training invariant support vector machines. Mach. Learn. **46**, 161–190 (2002)
7. DePooter, M., Ravazzolo, F., van Dijk, D.: Term structure forecasting using macro factors and forecast combination. Working Paper 2010/01, Norges Bank (2009)
8. Diebold, F., Li, C.: Forecasting the term structure of government bond yields. J. Econ. **130**, 337–364 (2006)
9. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. Adv. Comput. Math. **13**, 1–50 (2000)
10. Exterkate, P., Van Dijk, D., Heij, C., Groenen, P.J.: Forecasting the yield curve in a data-rich environment using the factor-augmented Nelson–Siegel Model. J. Forecast. **32**(3), 193–214 (2013)
11. Friedman, J., Hastie, T., Tibshirani, R.: The Elements of Statistical Learning, vol. 2. Springer (2017)
12. Gerhart, C., Lütkebohmert, E.: Empirical analysis and forecasting of multiple yield curves (2019). Available on https://ssrn.com/abstract=3311998
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**, 1735–1780 (1997)
14. Huang, W., Nakamori, Y., Wang, S.Y.: Forecasting stock market movement direction with support vector machine. Comput. Oper. Res. **32**, 2513–2522 (2005)
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (2014)
16. Koopman, S.J., van der Wel, M.: Forecasting the US term structure of interest rates using a macroeconomic smooth dynamic factor model. Int. J. Forecast. **29**(4), 676–694 (2013)

17. Lantz, B.: Machine Learning with R, 2nd ed. Packt Publishing Ltd. (2015)
18. Moench, E.: Term structure surprises: the predictive content of curvature, level, and slope. J. Appl. Econ. **27**(4), 574–602 (2012)
19. Nelson, C., Siegel, A.: Parsimoneous modeling of yield curves. J. Bus. **60**, 473–489 (1987)
20. Steinwart, I., Christmann, A.: Support Vector Machines. Springer Science & Business Media (2008)
21. Witten, I., Frank, E., Hall, M.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann Publishers Inc. (2011)
22. Wolpert, D.: Stacked generalization. Neural Netw. **5**, 241–259 (1992)

The Changing Shape of Sovereign Default Intensities

Yusho Kagraoka and Zakaria Moussa

Abstract The term structure of sovereign default intensities evolves over time along with rising/declining levels and steeping/flatting of the slope; a hump shape may exist in the default intensity curve, and the location of the hump changes. Thus, the default intensity model should have the flexibility to capture most of the variations in the term structure of the default intensities. The dynamic Nelson–Siegel (DNS) model with a time-varying decay parameter is appropriate to generate such default intensity curves. The paper studies the default intensities estimated from credit default swap (CDS) spreads by the DNS model with a time-varying decay parameter. Empirical studies were conducted on the German and U.S. CDS markets. The model parameters were successfully estimated using the Kalman filter. It is found that the decay parameters change over time and the magnitude of the decay parameter is positively related to the level of default intensities.

Keywords Sovereign credit default swap · Sovereign default intensities · The Nelson–Siegel model · State space model · Kalman filter

Introduction

In the aftermath of the 2007–2008 financial crisis and the European sovereign debt crisis that started in 2008, sovereign credit default swap (CDS) spreads surged dramatically. Not only the levels of CDS spreads but also the shape of the term structure of CDS spreads changed dramatically during both these periods. The term structure of CDS spreads reflects the default probabilities of a reference entity and the timing of default. Thus, studying the shape of the term structure of CDS spreads and their evolution is important for forecasting CDS spreads and controlling credit risk.

Y. Kagraoka (✉)
Musashi University, Toyotama-Kami 1-26-1, Nerima-Ku, Tokyo 176-8534, Japan
e-mail: kagraoka@cc.musashi.ac.jp; kagraoka@gmail.com

Z. Moussa
Université de Nantes, IAE Nantes—Économie & Management, Chemin la Censive du Tertre BP 52231, 44322 Nantes, Cedex 3, France
e-mail: Zakaria.Moussa@univ-nantes.fr

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_14

Nelson and Siegel [18] employed a set of simple functions to describe the shape of the term structure of interest rates. Diebold and Li [5] applied the Nelson–Siegel (NS) model to forecast the term structure of interest rates. Diebold et al. [6] investigated the term structure of interest rates using the NS model accompanied by macroeconomic variables and found that the macroeconomic factors improved forecasting of the term structure of interest rates. Many variations were developed to improve the NS model. For instance, Koopman et al. [12] extended the NS model by allowing the time-varying decay parameter and incorporating a GARCH structure into the error terms. Christensen et al. [3] and Coroneo et al. [4] constructed the arbitrage-free NS model. Levant and Ma [17] and Zhu and Rahman [22] extended the NS model to include Markov switching. The NS model has been applied not only to government bond yields but also to other term structures. Yu and Zivot [21] and Krishnan et al. [13] applied the NS model to corporate bond yields. Shaw et al. [20] adopted a straightforward approach and applied the NS model to the European sovereign CDS spreads.

Credit default swap spreads depend on the default intensities and risk-free rates because risk-free rates are inevitable to obtain the present value of the fixed and floating legs of the CDS. The CDS spread is set to the level at which the present value of periodic payment made by a protection buyer and the present value of a loss compensation caused by the default of a reference entity are both equal. Therefore, CDS spreads are complicated functions of the risk-free rates and default intensities. For instance, Shaw et al. [20] stated that CDS is a pure credit instrument and is isolated from the interest rate risk. Thus, the interpretation that CDS spread reflects pure credit risk is wrong. Shaw et al.'s [20] study had another disadvantage. They applied the NS model to CDS spreads. However, it is not clear what a fundamental object is. The original NS model represents the dynamics of forward rates, and its model parameters are estimated from the term structures of spot rates. Shaw et al. [20] applied a functional form for "spot rates" to the CDS spreads. Thus, in Shaw et al. [20], the counterpart to "forward rates" is not interpreted clearly. In the original NS model, the functional form for forward rates has a clear interpretation: level, slope, and curvature of a term structure of forward rates. However, Shaw et al. [20] did not provide any financial or economic interpretation for the underlying CDS spreads.

The purpose of the study is to apply the dynamic Nelson–Siegel (DNS) model with the time-varying decay parameter to the default intensities that are extracted from CDS spreads and government bond yields. The DNS factors correspond to the level, slope, and curvature of the default intensities, and the time-varying decay parameter can accommodate various time evolutions of default intensity curve. The model parameter is estimated using the Kalman filter. The empirical studies of the DNS model are conducted on the German and U.S. CDS markets.

The remaining of this chapter is organized as follows. The second section provides a theoretical background and the methodology. The third section is devoted to the presentation of the data used to illustrate the approach, the results, and a discussion. The fourth section contains a brief conclusion.

Methodology

Default Intensity and Risk-Free Rate

The valuation formula for CDS in the reduced form model is given in several textbooks, including Brigo and Mercurio [2], Duffie and Singleton [7], Lando [16], and Schönbucher [19]. This study follows Lando [14, 15] and Houweling and Vorst [9]. In their model, the default event of a reference entity is modeled by a point process. Let τ, λ_t, and $\Pr(t, T)$ denote a default time, the default intensity at time t, and the survival probability at time t up to time T, respectively. The following relationship holds for them:

$$\Pr(t, T) = \mathrm{E}\left[1_{\{\tau > T\}}\right] = \mathrm{E}_t\left[\exp\left(-\int_t^T ds\, \lambda_s\right)\right]. \qquad (1)$$

The value of the fixed leg of periodic payment is the expected present value of the periodic payments prior to the default time under a martingale measure. The value of the floating leg is the expected present value of compensation for a default under the martingale measure. The CDS spread is set to the level at which the value of the fixed leg equals that of the floating leg. Apparently CDS spreads depend not only on the default probabilities of a reference entity but also on the term structure of risk-free rates.

Investors and researchers regard government bond yields as proxies for the risk-free rates. However, nonzero spreads of the sovereign CDS imply that government bonds are not risk-free assets. In fact, government bonds should be evaluated assuming that their default probability is greater than zero. Thus, government bond yields depend on the risk-free rates and the default intensities of the government.

Both CDS spreads and government bond yields are a complicated function of the risk-free rates and default intensities. It is not simple to disentangle the risk-free rates and the default intensities from the CDS spreads and government bond yields. A procedure to decompose government bond yields to the risk-free rates and credit spreads was proposed by Kagraoka and Moussa [11] and Kagraoka [10]. The procedure is essentially subtracting the credit risk spread implied in the sovereign CDS spreads from the bond yields to extract the risk-free rates under the assumption that the risk-free rates and default intensities are independent. This study employs this method to obtain the default intensities.

State Space Models for Default Intensities

Nelson and Siegel [18] proposed a set of parsimonious functions to express a yield curve. This model has been used to express the term structure of interest rates by many central banks worldwide as reviewed by the Bank for International Settlements [1]. Diebold and Li [5] is the first paper that advocates the NS methodology to forecast the time evolutions of the term structures of interest rates. In the NS model, time-t forward rate maturing at $t + \tau$ is expressed as

$$f_t(\tau) = \beta_{1,t} + \beta_{2,t} e^{-\lambda^{NS}\tau} + \beta_{3,t} \lambda^{NS} e^{-\lambda^{NS}\tau}. \tag{2}$$

This parametrisation of the forward rate can reproduce various shapes of the term structure of forward rates. Changes in the parameters generate variations of the term structure of forward rates. The NS model has four parameters; the level parameter, $\beta_{1,t}$, represents a long-term interest rate and changes in the level parameter causing parallel shifts in the term structure of forward rates; the slope parameter, $\beta_{2,t}$, dictates a gradient of the term structure of forward rates and its movements induce steepening or flattening of the term structure of forward rates. The curvature parameter, $\beta_{3,t}$, represents a hump shape of the term structure of forward rates and its variations produce changes in the degree of hump in the term structure of forward rates. The decay parameter, λ^{NS}, governs both the speed of the exponential decay rate of the slope factor and the location of the hump in the curvature factor. Spot rate,

$$r_t(\tau) = \frac{1}{\tau} \int_t^{t+\tau} ds\, f_t(s), \tag{3}$$

is an average of the forward rates, and it is expressed as

$$r_t(\tau) = \beta_{1,t} + \beta_{2,t} \left(\frac{1 - e^{-\lambda^{NS}\tau}}{\lambda^{NS}\tau} \right) + \beta_{3,t} \left(\frac{1 - e^{-\lambda^{NS}\tau}}{\lambda^{NS}\tau} - e^{-\lambda^{NS}\tau} \right) \tag{4}$$

in the NS model. All previous empirical studies estimate the NS parameters from the term structure of spot rates by using Eq. (4).

In the context of CDS valuation, forward rates correspond to default intensities. Therefore, spot rates correspond to the average default intensities,

$$y_t(\tau) = \frac{1}{\tau} \int_t^{t+\tau} ds\, \lambda_s. \tag{5}$$

This study empirically examines the average default intensity by applying the DNS model.

Diebold and Li [5] proposed to fix to a priori value to the decay parameter and formulated the dynamics of the term structure of interest rates by using a vector autoregressive model. Diebold et al. [6] employed the state-space model to describe the dynamics of the term structure of interest rates expressed by the NS model. The evolution of the term structure of interest rates is generated by the transition equation,

$$\beta_{t+1} - \mu^{\beta} = T^{\beta}\left(\beta_t - \mu^{\beta}\right) + R_t^{\beta}\eta_t^{\beta}, \tag{6}$$

where β_t is a 3×1 unobservable state vector, T^{β} is a 3×3 matrix, the error term η_t^{β} is a 3×1 column vector following $\eta_t^{\beta} \sim N\left(0, \Sigma_{\eta}^{\beta}\right)$, and $R_t^{\beta} = I$. The vector of the observable variables is denoted by $y_t(\tau) = \left(y_t(\tau_1), y_t(\tau_2), \cdots, y_t\left(\tau_p\right)\right)'$, and the measurement equation is written as

$$y_t(\tau) = \Lambda\beta_t + \varepsilon_t^{\beta} \tag{7}$$

where

$$\Lambda = \begin{pmatrix} 1 & \frac{1-e^{-\lambda^{NS}\tau_1}}{\lambda^{NS}\tau_1} & \frac{1-e^{-\lambda^{NS}\tau_1}}{\lambda^{NS}\tau_1} - e^{-\lambda^{NS}\tau_1} \\ 1 & \frac{1-e^{-\lambda^{NS}\tau_2}}{\lambda^{NS}\tau_2} & \frac{1-e^{-\lambda^{NS}\tau_2}}{\lambda^{NS}\tau_2} - e^{-\lambda^{NS}\tau_2} \\ \vdots & \vdots & \vdots \\ 1 & \frac{1-e^{-\lambda^{NS}\tau_p}}{\lambda^{NS}\tau_p} & \frac{1-e^{-\lambda^{NS}\tau_p}}{\lambda^{NS}\tau_p} - e^{-\lambda^{NS}\tau_p} \end{pmatrix}, \tag{8}$$

and $\varepsilon_t^{\beta} \sim N\left(0, \Sigma_{\varepsilon}^{\beta}\right)$. It is assumed that the white noise transition disturbance, η_t^{β}, and the measurement disturbances, ε_t^{β}, are orthogonal to each other. The initial state is assumed to be orthogonal to the transition and measurement disturbances; $\mathrm{E}\left[\beta_1\eta_t^{\beta'}\right] = 0$ and $\mathrm{E}\left[\beta_1\varepsilon_t^{\beta'}\right] = 0$. The non-diagonal elements of Σ_{η}^{β} allow for the shocks to the term structure to be correlated. It is further assumed that $\Sigma_{\varepsilon}^{\beta}$ is an orthogonal matrix.

Koopman et al. [12] developed the DNS model with the time-varying decay parameter, λ_t^{NS}. This study follows their methodology. Logarithm of the decay parameter is included in the fourth element of the state space to ensure positivity of it,

$$\alpha_t = \begin{pmatrix} \beta_t \\ \ln\lambda_t^{NS} \end{pmatrix}. \tag{9}$$

This model is not linear with respect to $\ln\lambda_t^{NS}$, and this property makes it difficult to use a Kalman filter technique for model estimation. Koopman et al. [12] linearized this model as follows:

$$\alpha_{t+1} - \mu^{\alpha} = T^{\alpha}(\alpha_t - \mu^{\alpha}) + R_t^{\alpha}\eta_t^{\alpha}, \tag{10}$$

$$y_t(\tau) = Z\alpha_t + \left(\Lambda\left(a_{t|t-1}\right) - Za_{t|t-1}\right) + \varepsilon_t^{\alpha}, \tag{11}$$

$$Z_t = \left(\Lambda_1(\tau)\ \Lambda_2(\tau)\ \Lambda_3(\tau)\ \lambda_t^{NS}\sum_{j=1}^{3}\dot{\Lambda}\left(a_{j,t|t-1}\right)a_{j,t|t-1}\right), \tag{12}$$

and

$$\dot{\Lambda}(\alpha_t) = \begin{pmatrix} \frac{\partial \Lambda_1(\tau)}{\partial \lambda_t^{NS}} & \frac{\partial \Lambda_2(\tau)}{\partial \lambda_t^{NS}} & \frac{\partial \Lambda_3(\tau)}{\partial \lambda_t^{NS}} \end{pmatrix} = \begin{pmatrix} 0 & \frac{\partial \Lambda_2(\tau)}{\partial \lambda_t^{NS}} & \frac{\partial \Lambda_3(\tau)}{\partial \lambda_t^{NS}} \end{pmatrix} \tag{13}$$

where α_t is a 4×1 unobservable state vector, T^α is a 4×4 matrix, and the error term η_t^α is a 4×1 column vector with the coefficient matrix $R_t^\alpha = I$. This state-space model is estimated by maximizing the log-likelihood function, as explained by Durbin and Koopman [8].

Empirical Analysis

Data

Empirical studies were conducted on the German and U.S. CDS markets. Weekly data were collected from 8 October 2008 to 27 December 2017 to estimate the NS model. Datastream provides CDS spread data for Germany in USD and for the U.S. in EUR, maturing at 6 months and 1, 2, 3, 4, 5, 7, and 10 years. The spot rates of the AAA-rated European government bonds maturing from 3 months to 10 years, with 1-month increments, are obtained from the European Central Bank. The Federal Reserve Board gives the spot rates for the U.S. Treasuries from 1 year to 10 years, with 1-year increments.

The risk-free rates and default intensities are estimated following Kagraoka and Moussa [11] and Kagraoka [10]. Kagraoka [10] advocates employing CDS spreads in a foreign currency to estimate the default intensities. Thus, CDS spreads for Germany in USD and for the U.S. in EUR are employed to estimate default intensities. Spot rates in EUR and USD are depicted in Fig. 1 where spot rates are expressed in percent. In addition, CDS spreads for Germany in USD and for the U.S. in EUR are depicted in Fig. 1 where CDS spreads are expressed in percent. The estimated risk-free rates and (average) default intensities for Germany and for the U.S. are illustrated in Fig. 2 where all the values are expressed in percent. Summary statistics of the average default intensities for Germany and for the U.S. are given in Table 1.

Parameter Estimation

The NS model parameters are estimated from the weekly data of the average default intensities by maximizing likelihood. The average default intensities maturing at 6 months and 1, 2, 3, 4, 5, 7, and 10 years are regarded as observable by matching the CDS maturities.

The CDS spreads and the average default densities of Germany and the U.S. were volatile between 2008 and 2013 and the shapes of their term structure of the average default intensities changed drastically. For Germany, the term structure of the average default intensities was very low and flat at October 2008. Later, the level

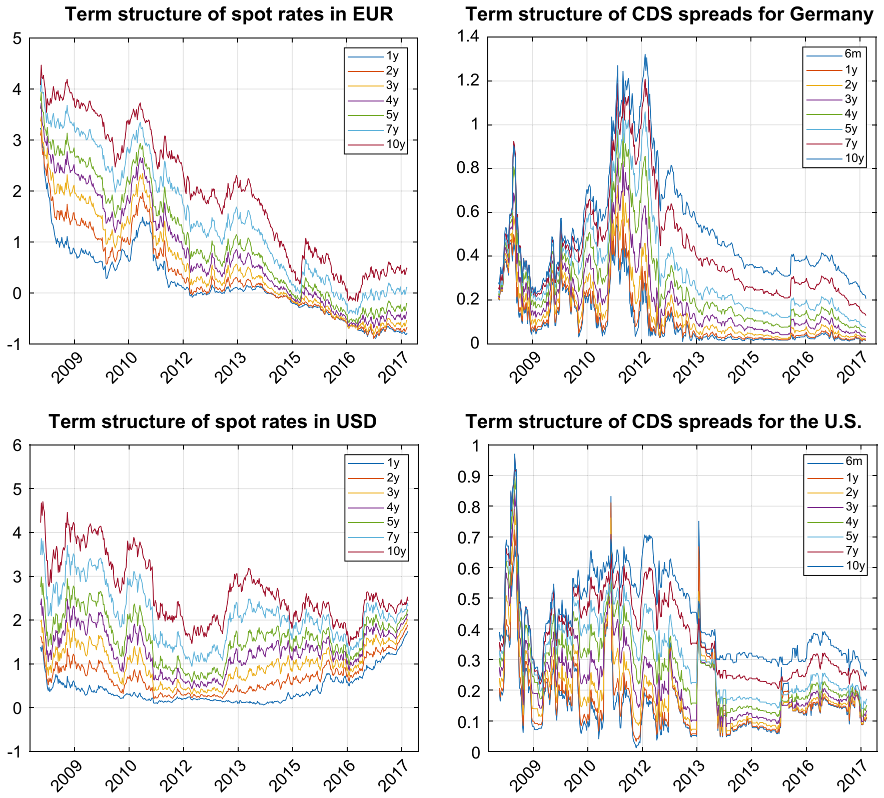


Fig. 1 Term structure of spot rates in EUR (spot rates of the AAA-rated European government bonds) (upper left) and that in USD (U.S. Treasury spot rates) (lower left). Term structure of CDS spreads for Germany in USD (upper right) and for the U.S. in EUR (lower right)

of term structure of the average default intensities surged abruptly and had a hump at around 5 years. From 2013, the term structure of the average default intensities resumed to flat at a low level. For the U.S., the term structure of the average default intensities was very volatile from 2008 to 2014 and it took a convex or concave shape. Therefore, the NS model with time-varying decay parameter is appropriate to model the term structure of the average default intensities for both Germany and the U.S.

The estimated parameters of the DNS model with time-varying decay parameters for Germany and the U.S. are reported in Table 2. The diagonal elements of $\Sigma_{\varepsilon}^{\alpha}$ are expressed by taking their logarithm. Trajectories of the state variables for Germany and the U.S. are presented in Fig. 3. This figure shows the time decay parameters for Germany and the U.S. vary with time.

First, the estimated parameters of default intensities for Germany are discussed. Level, L_t, is likely to remain the same because $T_{1,1} = 0.997$ and $T_{1,j}$ $(j = 2, 3, 4)$ are close to zero. Likewise, the slope, S_t, is likely to remain unchanged because $T_{2,2} = 1.050$ and $T_{2,j}$ $(j = 1, 3, 4)$ are close to zero. The curvature, C_t, is mainly affected

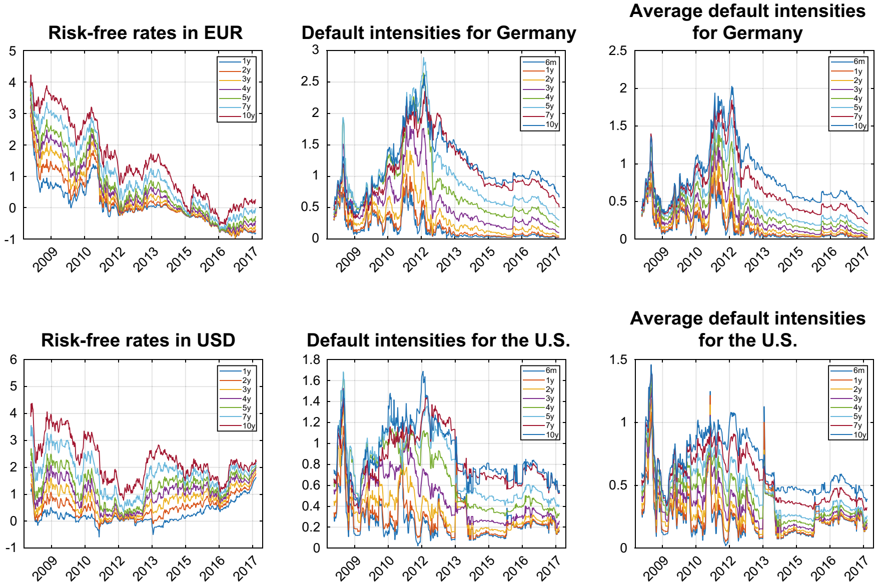


Fig. 2 Term structure of risk-free rates in EUR (upper left) and USD (lower left). Term structure of default intensities for Germany (upper middle) and for the U.S. (lower middle). Term structure of average default intensities for Germany (upper right) and for the U.S. (lower right)

by itself because $T_{3,3} = 0.980$, and negatively affected by level and slope because $T_{3,1} = -0.211$ and $T_{3,2} = -0.172$. The logarithm of the time decay parameter, $\ln \lambda_t$, is mainly affected by itself because $T_{4,4} = 0.960$, and it is affected by all the factors because $T_{4,1} = 0.154$, $T_{4,2} = 0.144$, and $T_{4,3} = 0.035$. The trajectories of the level, slope, curvature, and decay parameters are intriguing. The level correlates negatively to the slope and curvature. The decay parameter varies over time; it rises when the default intensities become higher and declines when the default intensities decrease. During the financial market turmoil, the decay parameter peaked to 0.139 on 15 April 2009. It took around 0.02 after mid-2012. The correlations between the state variables are given in the left panel in Table 3. The negative correlation between the decay parameter and the level is apparent. The correlations between the decay parameter and the slope or curvature are positive. The correlation between the decay parameter and the 5-year average default intensity is 0.650. Thus, the higher the average default intensity, the greater is the decay parameter.

Next, the estimated parameters of default intensities for the U.S. are discussed. Level is affected by itself and slope. Slope is affected by itself as well as the level and curvature. Curvature is affected by level, slope, and curvature. These parameters produce upward or downward sloping curves of default intensities for the U.S. The logarithm of the time decay parameter is mainly affected by itself because $T_{4,4} = 1.008$, and by the slope ($T_{4,2} = -0.127$) and curvature ($T_{4,3} = 0.060$). The trajectories of the level, slope, curvature, and time decay parameters are intriguing.

Table 1 Summary statistics for average default intensities for Germany and for the U.S. For each maturity mean, standard deviation (sd), minimum, maximum, and three autocorrelation coefficients at 1 week (ρ(1)), 4 weeks (ρ(4)), and 13 weeks (ρ(13)) are reported

Average default intensity for Germany

	Mean	sd	Min	Max	ρ(1)	ρ(4)	ρ(13)
6 m	0.152	0.174	0.015	0.815	0.954	0.836	0.528
1y	0.176	0.187	0.020	0.898	0.961	0.854	0.556
2y	0.233	0.221	0.032	1.085	0.972	0.882	0.605
3y	0.310	0.262	0.049	1.276	0.978	0.902	0.657
4y	0.412	0.311	0.076	1.468	0.983	0.920	0.718
5y	0.518	0.356	0.113	1.635	0.985	0.931	0.758
7y	0.670	0.376	0.200	1.844	0.986	0.937	0.787
10y	0.816	0.389	0.325	2.027	0.986	0.941	0.809

Average default intensity for the U.S

	Mean	sd	Min	Max	ρ(1)	ρ(4)	ρ(13)
6 m	0.245	0.173	0.018	1.247	0.900	0.620	0.119
1y	0.260	0.172	0.049	1.215	0.915	0.662	0.181
2y	0.297	0.176	0.092	1.185	0.942	0.748	0.331
3y	0.347	0.188	0.122	1.278	0.960	0.816	0.476
4y	0.412	0.203	0.171	1.364	0.971	0.862	0.582
5y	0.478	0.216	0.210	1.431	0.977	0.887	0.641
7y	0.574	0.214	0.302	1.454	0.978	0.890	0.654
10y	0.667	0.218	0.351	1.460	0.977	0.892	0.669

The level negatively correlates to the slope and curvature as in the case of Germany. The decay parameter varies over time; it rises when the default intensities become higher and drops when the default intensities lower. In the financial market turmoil, it peaked at 0.085 at 15 April 2009. It was below 0.03 since 2013. The correlations between the state variables are given in the right panel in Table 3. The correlation signs between the decay parameter and the level, slope or curvature are opposite to that for Germany. The correlation between the decay parameter and the 5-year average default intensity is 0.405. Thus, the higher the average default intensity, the greater the decay parameter becomes.

Conclusion

This study applies the dynamic Nelson–Siegel (DNS) model with the time-varying decay parameter to the average default intensities. The DNS factors correspond to the level, slope, and curvature of the default intensities, and the empirical results provide a

Table 2 Estimated parameters of the Nelson–Siegel model with the time-varying decay parameter for average default intensity for Germany (upper panel) and for the U.S. (lower panel). Standard errors appear in parentheses

Germany

	T					Σ_η					
	L_t	S_t	C_t	$\ln(\lambda_t)$		L_t	S_t	C_t	$\ln(\lambda_t)$		μ
L_t	0.997	– 0.011	– 0.007	0.011	L_t	0.010				L_t	0.799
	(0.055)	(0.056)	(0.018)	(0.018)		(0.001)					(1.786)
S_t	0.042	1.050	0.024	– 0.026	S_t	– 0.008	0.010			S_t	– 0.385
	(0.066)	(0.079)	(0.027)	(0.032)		(0.001)	(0.002)				(2.026)
C_t	– 0.211	– 0.172	0.980	0.020	C_t	– 0.008	0.010	0.013		C_t	– 2.181
	(0.074)	(0.093)	(0.029)	(0.038)		(0.001)	(0.002)	(0.002)			(2.849)
$\ln(\lambda_t)$	0.154	0.144	0.035	0.960	$\ln(\lambda_t)$	– 0.003	0.004	0.003	0.003	$\ln(\lambda_t)$	– 3.212
	(0.037)	(0.047)	(0.015)	(0.017)		(0.001)	(0.001)	(0.001)	(0.000)		(0.151)

Σ_ε

6 m	1y	2y	3y	4y	5y	7y	10y
−13.426	−7.804	−12.908	−8.018	−12.223	−7.107	−6.171	−6.889
(3.789)	(0.201)	(1.200)	(0.470)	(1.259)	(0.404)	(0.122)	(0.089)

(continued)

Table 2 (continued)

The U.S.

	T					Σ_η					
	L_t	S_t	C_t	$\ln(\lambda_t)$		L_t	S_t	C_t	$\ln(\lambda_t)$		μ
L_t	0.878	−0.064	−0.011	0.010	L_t	0.002				L_t	1.078
	(0.020)	(0.022)	(0.008)	(0.006)		(0.000)					(0.154)
S_t	0.056	0.923	0.039	−0.005	S_t	−0.002	0.010			S_t	−0.861
	(0.051)	(0.029)	(0.017)	(0.024)		(0.001)	(0.001)				(0.213)
C_t	0.217	0.376	0.895	−0.033	C_t	−0.001	−0.004	0.021		C_t	−1.060
	(0.073)	(0.064)	(0.019)	(0.036)		(0.001)	(0.001)	(0.003)			(0.413)
$\ln(\lambda_t)$	−0.032	−0.127	0.060	1.008	$\ln(\lambda_t)$	−0.001	0.003	−0.008	0.006	$\ln(\lambda_t)$	−3.953
	(0.033)	(0.038)	(0.010)	(0.016)		(0.001)	(0.001)	(0.001)	(0.001)		(0.086)

Σ_ε

6 m	1y	2y	3y	4y	5y	7y	10y
−7.929	−11.144	−9.260	−8.868	−10.530	−8.089	−7.842	−8.242
(0.253)	(2.208)	(0.229)	(0.739)	(0.690)	(0.609)	(0.106)	(0.197)

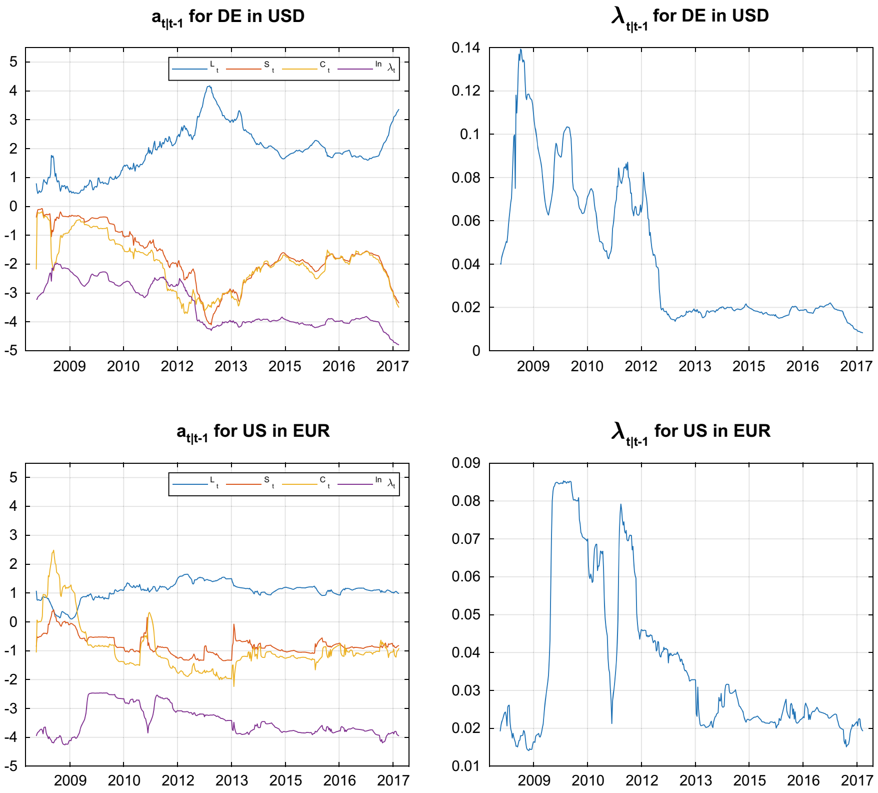


Fig. 3 Time series of the state variables for Germany (upper left) and that for the U.S. (lower left) are drawn. Time series of the time-varying decay parameter, λ_t^{NS}, for Germany (upper right) and that for the U.S. (lower right) are depicted

Table 3 Correlation between the state variables for Germany (left) and that for the U.S. (right)

Germany					The U.S.				
	L_t	S_t	C_t	$\ln(\lambda_t)$		L_t	S_t	C_t	$\ln(\lambda_t)$
L_t	1.000				L_t	1.000			
S_t	−0.980	1.000			S_t	−0.888	1.000		
C_t	−0.937	0.919	1.000		C_t	−0.875	0.866	1.000	
$\ln(\lambda_t)$	−0.664	0.767	0.544	1.000	$\ln(\lambda_t)$	0.328	−0.274	−0.402	1.000

clear interpretation of the estimated parameters. The DNS model with a time-varying decay parameter accommodates the time-varying features of the term structure of the average default intensities and their variations. Empirical studies for the German and U.S. CDS markets are conducted and the Kalman filter is used to estimate the model parameters. The empirical results show that the decay parameter is time-varying and closely relates to the level, slope and curvature of the average default intensities. The higher the average default intensity, the greater is the magnitude of the decay parameter. Future studies will investigate forecasting of the average default intensity by the DNS model.

Acknowledgements This work was supported by JSPS KAKENHI Grant Numbers 18K01707.

References

1. Bank for International Settlements: Zero-coupon yield curves: technical documentation, BIS Papers No 25 (2005)
2. Brigo, D., Mercurio, F.: Interest Rate Models—Theory and Practice: With Smile, Inflation and Credit, 2nd edn. Springer, Berlin, Heidelberg (2006)
3. Christensen, J.H.E., Diebold, F.X., Rudebusch, G.D.: The affine arbitrage-free class of Nelson-Siegel term structure models. J. Econ. **164**(1), 4–20 (2011)
4. Coroneo, L., Nyholm, K., Koleva, R.V.: How arbitrage-free is the Nelson-Siegel model? J. Empir. Financ. **18**(3), 393–407 (2011)
5. Diebold, F.X., Li, C.: Forecasting the term structure of government bond yields. J. Econ. **130**(2), 337–364 (2006)
6. Diebold, F.X., Rudebusch, G.D., Aruoba, S.B.: The macroeconomy and the yield curve: a dynamic latent factor approach. J. Econ. 131(1), 309–338 (2006)
7. Duffie, D., Singleton, K.J.: Credit Risk: Pricing, Measurement, and Management. Princeton University Press (2003)
8. Durbin, J., Koopman, S.J.: Time Series Analysis by State Space Methods, 2nd edn. Oxford University Press (2012)
9. Houweling, P., Vorst, T.: Pricing default swaps: empirical evidence. J. Int. Money Financ. **24**(8), 1200–1225 (2005)
10. Kagraoka, Y.: Are the risk-free interest rates correlated with sovereign default intensities? J. Fixed Income **28**(4), 91–103 (2019)
11. Kagraoka, Y., Moussa, Z.: Estimation of the term structure of CDS-adjusted risk-free interest rates. J. Fixed Income **24**(2), 29–44 (2014)
12. Koopman, S.J., Mallee, M.I.P., van der Wel, M.: Analyzing the term structure of interest rates using the dynamic Nelson-Siegel model with time-varying parameters. J. Bus. Econ. Stat. **28**(3), 329–343 (2010)
13. Krishnan, C.N.V., Ritchken, P.H., Thomson, J.B.: Predicting credit spreads. J. Financ. Intermediation **19**(4), 529–563 (2010)
14. Lando, D.: Three essays on contingent claims pricing, Ph.D. dissertation, Cornell University (1994)
15. Lando, D.: On cox processes and credit risky securities. Rev. Deriv. Res. **2**, 99–120 (1998)
16. Lando, D.: Credit Risk Modeling: Theory and Applications. Princeton University Press (2004)
17. Levant, J., Ma, J.: A dynamic Nelson-Siegel yield curve model with Markov switching. Econ. Model. **67**, 73–87 (2017)
18. Nelson, C.R., Siegel, A.F.: Parsimonious modeling of yield curves. J. Bus. **60**(4), 473–489 (1987)

19. Schönbucher, P.J.: Credit Derivatives Pricing Models: Models, Pricing and Implementation. Wiley Ltd. (2003)
20. Shaw, F., Murphy, F., O'Brien, F.: The forecasting efficiency of the dynamic Nelson Siegel model on credit default swaps. Res. Int. Bus. Financ. **30**, 348–368 (2014)
21. Yu, W.-C., Zivot, E.: Forecasting the term structures of treasury and corporate yields using dynamic Nelson-Siegel models. Int. J. Forecast. **27**(2), 579–591 (2011)
22. Zhu, X., Rahman, S.: A regime-switching Nelson-Siegel term structure model of the macroeconomy. J. Macroecon. **44**, 1–17 (2015)

Permutation Entropy as the Measure of Globalization Process

Janusz Miśkiewicz

Abstract The application of permutation entropy as the globalization measure is discussed within this paper. In order to verify the stated hypothesis the following time series were investigated: CO_2 emission, CPI, employment rate and GDP and the countries were grouped such that the groups differ in globalization level. The received results support the thesis that the objective globalization measure can be based on the permutation entropy.

Keywords Globalization · Permutation entropy · Time series analysis

Introduction

Globalization is a process that reduces the distance between societies in various aspects. This process was initiated by inventions facilitating transport, such as railways, cars, aeroplanes and communication—the telegraph, the telephone, the radio, the television and the Internet. This process has transformed the evolution of societies from a local phenomenon to a global phenomenon by strengthening long-range interaction and has introduced the development of societies to new paths. At present, communication means allow for easy and intense contacts between social groups. This process significantly influences not only culture but also, particularly, economy and politics. One of the most spectacular globalization processes in the twentieth century was the creation of the European Community, which at the level of political and legal regulation facilitates and improves the functioning of its members, and influence various aspects of the life of EC citizens. The globalization is also observed in the financial market. The illustration of the complex relationship on the contemporary stocks markets might be the complicated structure of share ownership [1]. The globalization of financial markets leads to complex functional relationship

J. Miśkiewicz (✉)
Institute of Theoretical Physics, University of Wrocław, pl. M. Borna 9, 50-204 Wrocław, Poland
e-mail: janusz.miskiewicz@ift.uni.wroc.pl

Department of Physics and Biophysics, Wrocaw University of Environmental and Life Sciences, ul. Norwida 25, 50-375 Wrocław, Poland

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_15

among stocks markets [2–4]. It affects also social integration [5]. So the process is broadly observed.

In fact, researches on globalization have a long history [6]. A huge effort was made in finding a proper description of the process. Many indexes have been introduced, [7, 8] e.g. Global Connectedness Index, Global Entrepreneurship Index, Global Food Security Index, Global Innovation Index, Global Liveability Ranking, Global Peace Index, Global Slavery Index, Global Terrorism Index, Global Web Index, Human Development Index, Human Poverty Index, Index of Economic Freedom, Internationalization Index, KOF Index of Globalization, Maastricht Globalization Index, Networked Readiness Index, OECD Better Life Index, Energy Globalization Index and many others. The common feature of mentioned indexes is that they are based on a designated parameter, which is considered as the characteristics of the globalization. In the present study, a different approach is proposed. The globalization measure is constructed, such that it would be possible to investigate various aspects of globalization by the same measure.

Globalization Measure

In the twenty-first century, the flow of information is the most dominating factor determining globalization. It is very probable that this tendency will continue in future. Thus, the measures related to the analysis of information seem to be the most promising. Another factor increasing globalization are the legal and political decisions. These regulations are primarily in the field of economy, international cooperation at the bilateral or multilateral level (e.g. the European Union). The ongoing process leads to the unification of the conditions in which companies operate. The situation resembles the information theory where imposed rules influence the complexity of a signal. Taking into account the described factors, it seems that the proper measure of globalization should be entropy, in particular, the information entropy, which characterizes the degree of noise/determinism in given data sets. The main difficulty in the direct use of Shannon information entropy (Eq. 1) is the limitation of the available data.

$$S = -\sum_i p_i ln(p_i) \tag{1}$$

Estimation of S requires knowledge of probabilities or occurrence of some quantities or even the probability distribution function of analysed parameters. In the case of the macroeconomy, it is hardly possible. For example, Gross Domestic Product (GDP) is usually presented as annually time series, in consequence, the typical, developed economy is described by a series of 50–100 data points. On the other hand in a long time span, one has to face the stationarity problem of such a time series. One of the possible solutions of the mentioned difficulties is the application of entropy related parameter, i.e. Theil index [9, 10].

In the present paper the approach based on permutation entropy is proposed [11, 12]. The main difference between Shannon entropy and the Permutation Entropy (PE) is that the latter one is a symbolic entropy focused on patterns rather than on probability distribution function, which makes it useful in an analysis of short time series.

For the convenience of the reader, the short description of PE is recalled here [2]: At each time s of the analysed time series $X = x_t, t = 1, \ldots, N$, a vector of D-th subsequent values is taken.

$$(x_s, x_{s+1}, \ldots, x_{s+D-1}). \tag{2}$$

This vector is sorted in an ascending order and a permutation pattern π is created.

$$\pi = (r_0 r_1 \ldots r_{D-1}) \tag{3}$$

which fulfils

$$x_{s+r_0} \leq x_{s+r_1} \leq \ldots x_{s+r_{D-2}} \leq x_{s+r_{D-1}} \tag{4}$$

Finally the permutation entropy is defined as follows:

$$PE = -\sum_{i=1}^{D!} \pi_i \ln \pi_i \tag{5}$$

In the case of globalization analysis PE should be followed by cross-correlation analysis. Within the present study, the Manhattan distance is used and it is defined as follows. Denoting the time series as A and B and their elements as a_i and b_i, respectively, the Manhattan distance between A and B in the interval $i \in (t, t+T)$ is defined as follows:

$$MD(A, B)_{t,T} = \sum_{i=t}^{t+T} |a_i - b_i|. \tag{6}$$

The proposed method of globalization measurement consists of the following two steps:

- first the time series are converted into PE time series and
- second the cross-correlation among them are analysed.

Since the globalization is rather a process than a state, therefore the final outcome should describe the evolution of a system. In order to fulfil these requirements the moving time window technique is used. In fact, in the proposed method two different time windows are required. The first one is needed to transform analysed time series into the entropy time series and the second window to measure the cross-correlation among them. At each step, the time windows are moved by one data item (the actual shift length depends on the probing frequency). Finally, the series of distance matrices are obtained. The last step is finding characteristic of the distance matrix. In the present study, the mean distance of each distance matrix is calculated and presented as the final result.

Data

The proposed method was verified on the GDP, CPI (consumer price index), employment rate and CO_2 emission time series. The macroeconomy data were obtained from the OECD database, while the CO_2 emission time series were downloaded from the European Commission web page [13]. In some cases the time series were covering only recent years, such series were rejected. The most comprehensive database was the GDP which consisted of yearly GDP time series of 93 countries covering the period from 1961 till 2016. CPI time series were monthly data of 25 countries from January 1970 to April 2015. The employment rate covered quarterly data of 26 countries from the beginning of 2001 till the end of 2017. The CO_2 emission annually time series covered dates 1970–2015 for 126 countries. The times series were converted into the relative increment time series Eq. (7).

$$R(t) = \frac{p(t) - p(t-1)}{p(t-1)} \tag{7}$$

Results

The proposed globalization measure depends on three parameters: the permutation order D, the permutation time window T and the distance–time window T_m. From the statistics quality point of view the longer the time window the more accurate is the result, but in applications, there are two main limitations which have to be taken into consideration. First of all the analysed time series are of limited length. The second factor is the problem of data stationarity, particularly if long time series are considered. On the other hand, the number of permutation patterns increases as factorial, so the estimation of their probabilities would require very long permutation time window. Therefore, in the present analysis, the permutation order was chosen from 3 up to 5 data points. The performed analysis showed that for the reasonable time window (not exceeding 12 years) the best choice is the permutation order of 3 or 4 and finally the results for $D = 4$ are discussed within this paper.

Since the aim of the research was to verify the proposed measure of the globalization the time series were chosen in such a way that in one group the globalization is expected and in the second group the globalization should not be seen or is much weaker. The first group consisted of the time series of GDP, CPI employment rate of European countries and CO_2 emission of all countries except the North American region. Europe is a particularly interesting case since the unification processes have

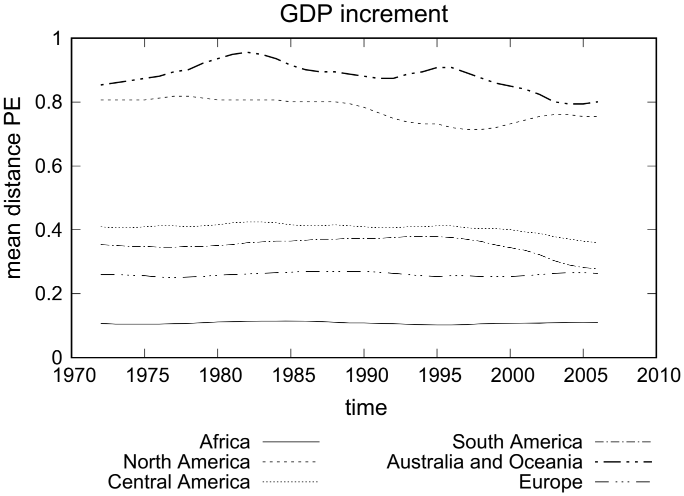


Fig. 1 Permutation entropy mean distance evolution of the GDP yearly increments. Analysis parameters: $D = 4y$, $T = 12y$, $T_m = 10y$

a long history and since the 50s of the twentieth century, European countries are increasing unification level (a short description of EU history can be found, e.g. on Britannica web page [14]. The comparative group where the time series of GDP, CPI, the employment rate of non-European countries and CO_2 emission rate of the North American countries.

The results are presented in Figs. 1, 2 and 3. The proposed globalization measure is based on the permutation entropy. Considering the general idea of the globalization which is the unification of regulations among regions and countries one can expect that the entropy distance between those of the high globalization level should be lower. In the opposite case—if the globalization is low then the entropy distance is expected to be high. This expectation perfectly agrees with the general outcome of the performed analysis as in Figs. 1, 2 and 3.

The most evident observation of the mean distance of the permutation entropy GDP increments evolution (Fig. 1) is that the appropriate lines present natural ordering from the highest: Australia and Oceania, North America, Central America, South America, Europe and Africa. The ordering corresponds to the globalization level of those regions. The top line presents the evolution of the mean permutation entropy distance among Australia and Oceania countries taking the value in the interval 0.85–0.95, while for European countries the average distance is about 0.25, so much lower, which can be related to the much higher level of integration among European countries. Whereas, Australia and Oceania region is, in fact, one huge country and many small island countries, naturally separated and with the lack of significant progress of the mutual relationship development [15, 16]. In this situation, the low level of economies globalization is not surprising. The second line is the North American

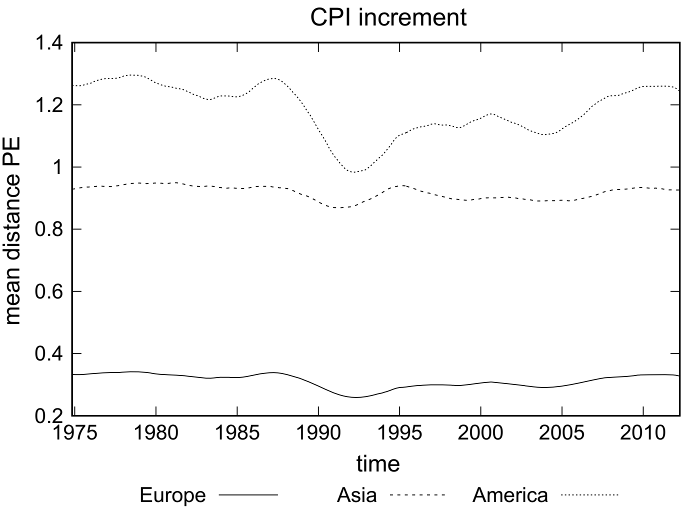


Fig. 2 Permutation entropy mean distance evolution of the CPI monthly increments. Analysis parameters: $D = 4m$, $T = 48m$, $T_m = 48m$

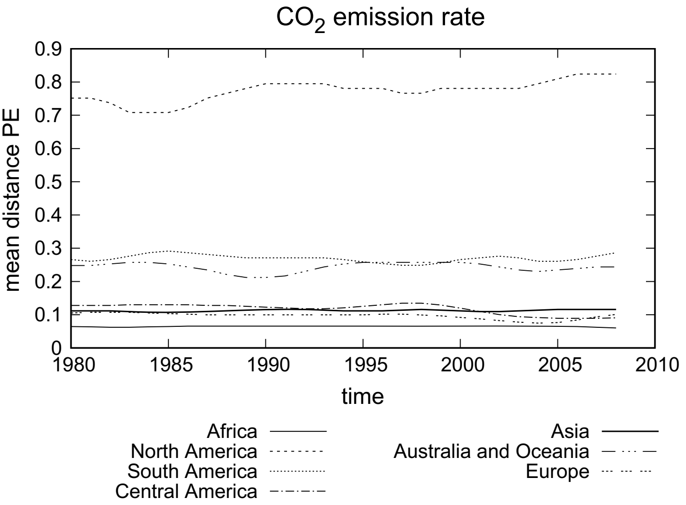


Fig. 3 Permutation entropy mean distance evolution of the CO_2 emission rate yearly increments. Analysis parameters: $D = 4y$; $T = 12y$, $T_m = 4y$

group. This is also the case of significantly different countries. The USA and Canada are so developed that are really independent. Mexico, the last country in this group is in contrast to the previous one much poorer. In results, the North American group consists of countries which are not following the same economic development pattern. This agrees with the high value of the mean distance of permutation entropy.

The very manifesting feature of the graph in Fig. 1 is the gap between the first top lines and the others showing the significant difference between the permutation entropy time series which coincides with differences in economic development patterns. The gap between the top group and the following one is on average of 0.4 and is significant considering the evolution of the PE mean distance. It is noticeable that the lowest line of the second group is the European PE mean distance. Taking into account that the European countries for the last 60 years are working on integration, the high level of similarities is not surprising. The lowest line presents the evolution of the mean PE of the African region. Although, it might be surprising the African countries are one of the poorest, suffering various economic, social, political and other problems and as a side effect, one can observe high similarities in development pattern of this region being the effect of similar difficulties rather than cooperation.

In the CPI time series analysis, only three groups are investigated due to the limitation on the data. Once again the main globalization factor governing the price index is the formation of the European Community. Indeed the line representing the distances among European countries are separated from two others by the gap of the size 0.6. The highest PE mean distances are observed for the American region related to the huge differences among countries.

The last set of analysed data—the CO_2 emission rate is related to one of the serious threads of contemporary world [17–20]. Since the CO_2 is considered as one of the leading factors of global warming [20–22]. Therefore a lot effort was made to reduce the CO_2 emission [23–25]. Unfortunately, the reduction is expensive and despite the general agreement, there are countries which did not accept the Kyoto agreement. The results obtained by proposed globalization measure are in fact spectacular. Starting from the top line of Fig. 3, which presents the evolution of the PE mean distance for the North American countries. Considering the fact that the USA and Canada are not signatories to the Kyoto agreement and these are highly industrialized countries, (with very high CO_2 emission) the huge gap between this group and the next one is absolutely understandable. This result seems to be the most spectacular. The situation is serious because since 2000 the mean PE distance has been increasing in this region. So, not much, or very little being done in joint action in the CO_2 emission reduction. The second group of lines represents Australia and Oceania and the South American countries. Although they have signed the Kyoto agreement, these are countries with serious economic problems and particularly South American countries are putting much effort in the economy/industry grow. Therefore, a significant CO_2 reduction cannot be expected. The very interesting case is the Europe region because the European Community strongly introduces restrictions on CO_2 emissions. This effort is well seen in years 1995–2005.

Another quite impressive result is the comparison of the averaged distance between European and non-European countries for the employment time series (Fig. 4). The averaged Manhattan distance of the permutation entropy for European countries is on the level 0.1 while for non-European takes the value of 0.35, so three times greater. Besides that, within the evolution, one can distinguish the periods 2009–2012, when new members entered EC which resulted in further unification of the employment market and decrease of permutation entropy distance.

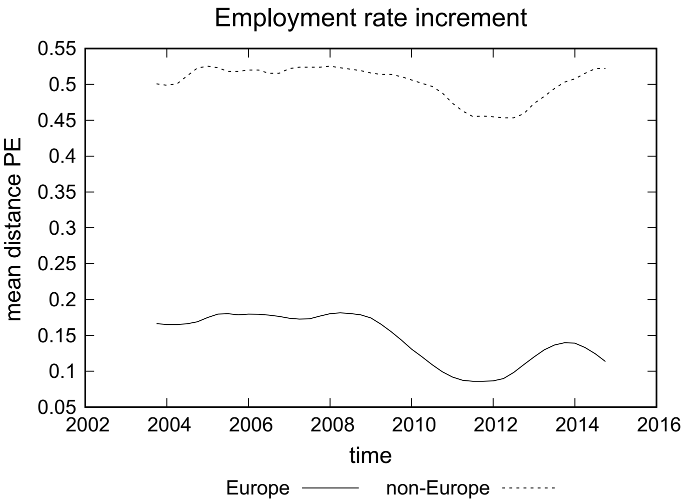


Fig. 4 Permutation entropy mean distance evolution of the employment rate quaterly increments. Analysis parameters: $D = 4q$, $T = 16q$, $T_m = 8q$

Conclusions

Summarizing, within the study it was verified if the entropy measure could be used as a globalization measure. The proposed method was verified on several groups of data series. The obtained results agree with the stated hypothesis.

Indifference to standard globalization measures pointed out in the introduction, it is focused on general time series feature rather than on a particular parameter, such that the proposed measure is universal. The same measure can be applied to various aspects of human activity giving the opportunity to analyse the globalization process more objectively.

References

1. Cerqueti, R., Rotundo, G., Ausloos, M., Cerqueti, R., Rotundo, G., Ausloos, M.: Investigating the configurations in cross-shareholding: a joint copula-entropy approach. Entropy **20**(2), 134 (2018). https://doi.org/10.3390/e20020134
2. Andersen, J.V., Nowak, A., Rotundo, G., Parrott, L., Martinez, S.: Price-quakes shaking the world's stock exchanges. PLOS ONE **6**(11), e26472 (2011). https://doi.org/10.1371/journal.pone.0026472
3. Bellenzier, L., Vitting Andersen, J., Rotundo, G.: Contagion in the world's stock exchanges seen as a set of coupled oscillators. Econ. Model. **59**, 224–236 (2016). https://doi.org/10.1016/j.econmod.2016.07.002
4. Kim, H.S., Min, H.G., McDonald, J.A.: Returns, correlations, and volatilities in equity markets: evidence from six OECD countries during the US financial crisis. Econ. Model. **59**, 9–22 (2016).

https://doi.org/10.1016/j.econmod.2016.06.016

5. The problem of comunity Cerqueti, R., Correani, L., Garofalo, G.: Economic interactions and social tolerance: a dynamic perspective. Econ. Lett. **120**(3), 458–463 (2013). https://doi.org/10.1016/j.econlet.2013.05.032
6. Gills, B.K., Thompson, W.: Globalization and Global History. Routledge (2012)
7. Samimi, P., Jenatabadi, H.S.: Globalization and economic growth: empirical evidence on the role of complementarities. PLoS One **9**(4) (2014). https://doi.org/10.1371/journal.pone.0087824
8. Samimi, P., Lim, G.C., Buang, A.A.: Globalization measurement: notes on common globalization indexes (7), 20 (2011)
9. Miśkiewicz, J.: Globalization—Entropy unification through the Theil index. Physica A **387**(26), 6595–6604 (2008)
10. Miśkiewicz, J., Ausloos, M.: Has the world economy reached its globalization limit? Physica A **389**(4), 797–806 (2010)
11. Bandt, C., Pompe, B.: Permutation entropy: a natural complexity measure for time series. Phys. Rev. Lett. **88**(17), 174102 (2002)
12. Zanin, M., Zunino, L., Rosso, O.A., Papo, D.: Permutation entropy and its main biomedical and econophysics applications: a review. Entropy **14**(8), 1553–1577 (2012). https://doi.org/10.3390/e14081553
13. EDGAR—GHG (CO_2, CH_4, N_2o, F-gases) emission time series 1990-2012 per region/country-European Commission. http://edgar.jrc.ec.europa.eu/overview.php?v=CO2ts1990-2015
14. https://www.britannica.com/topic/European-Union
15. Kelly, R.M., Bayes, J.H., Hawkesworth, M.E., Young, B.: Gender, Globalization, & Democratization. Rowman & Littlefield Publishers (2001)
16. Sprague, J.: Globalization and Transnational Capitalism in Asia and Oceania. Routledge (2015)
17. Canadell, J.G., Le Quéré, C., Raupach, M.R., Field, C.B., Buitenhuis, E.T., Ciais, P., Conway, T.J., Gillett, N.P., Houghton, R., Marland, G.: Contributions to accelerating atmospheric co2 growth from economic activity, carbon intensity, and efficiency of natural sinks. Proc. Natl Acad. Sci. **104**(47), 18866–18870 (2007)
18. Cox, P.M., Betts, R.A., Jones, C.D., Spall, S.A., Totterdell, I.J.: Acceleration of global warming due to carbon-cycle feedbacks in a coupled climate model. Nature **408**(6809), 184 (2000)
19. Eamus, D., Jarvis, P.G.: The direct effects of increase in the global atmospheric co2 concentration on natural and commercial temperate trees and forests. In: Advances in Ecological Research, vol. 19, pp. 1–55. Elsevier (1989)
20. Shakun, J.D., Clark, P.U., He, F., Marcott, S.A., Mix, A.C., Liu, Z., Otto-Bliesner, B., Schmittner, A., Bard, E.: Global warming preceded by increasing carbon dioxide concentrations during the last deglaciation. Nature **484**(7392), 49 (2012)
21. Crutzen, P.J., Mosier, A.R., Smith, K.A., Winiwarter, W.: N 2 o release from agro-biofuel production negates global warming reduction by replacing fossil fuels. In: Crutzen, P.J. (Eds.) A Pioneer on Atmospheric Chemistry and Climate Change in the Anthropocene, pp. 227–238. Springer (2016)
22. Norby, R.J., Luo, Y.: Evaluating ecosystem responses to rising atmospheric co2 and global warming in a multi-factor world. New Phytol. **162**(2), 281–293 (2004)
23. Copeland, B.R., Taylor, M.S.: Free trade and global warming: a trade theory view of the kyoto protocol. J. Environ. Econ. Manag. **49**(2), 205–234 (2005)
24. Jacobson, M.Z.: Control of fossil-fuel particulate black carbon and organic matter, possibly the most effective method of slowing global warming. J. Geophys. Res. Atmos. **107**(D19), ACH–16 (2002)
25. Jacobson, M.Z.: Review of solutions to global warming, air pollution, and energy security. Energy Environ. Sci. **2**(2), 148–173 (2009)

Time Series Analysis in Earth Sciences

Forecasting Subtidal Water Levels and Currents in Estuaries: Assessment of Management Scenarios

M. Á. Reyes Merlo, R. Siles-Ajamil and M. Díez-Minguito

Abstract Floods are one of the most harmful extreme events that occur in estuaries, which are induced by tides and freshwater discharges. The Guadalquivir River Estuary (SW Spain) has experienced multiple flooding events in recent decades. High-resolution time series from a long-term monitoring campaign were analyzed to assess the impacts of subtidal water levels and currents in this estuary. An autoregressive approach, which is considered as the nonstationarity of the freshwater discharge, indicated that water levels were well described by the linear superposition of levels induced by tides and freshwater discharge, whereas a nonlinear relationship between both tides and freshwater discharge reproduced the subtidal currents better. The obtained relationships were used to assess, on a medium-term basis, the effects of the expected 15% reduction in freshwater discharges and the planned 23% deepening of the navigation channel. The comparison among the results from four different scenarios based on these modifications and the present conditions of the estuary revealed that (1) subtidal levels will decrease when the freshwater input is reduced and (2) subtidal levels will be more harmful during extreme events due to channel deepening.

Keywords Guadalquivir estuary · Flood risk · Tides · River discharge · Management

Introduction

Estuaries are transition environments between the river and the sea whose dynamics are influenced by tides, waves, and fluvial processes [1], among others. These environments host ecosystems for a wide range of life forms are the economic foundation of many coastal nations and are the waterway between land and open sea.

M. Á. Reyes Merlo · R. Siles-Ajamil (✉) · M. Díez-Minguito
Andalusian Institute for Earth System Research (IISTA), University of Granada, Avda. del Mediterráneo, s/n, 18006 Granada, Spain
e-mail: reyess@ciccp.es

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*, Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_16

In the near future, human involvement and thus, social conflicts are expected to increase in estuaries. This activity includes increased demands on freshwater use, reclamation of marshes, use of soils, and increase in water depths to allow the navigation of larger ships. Consequently, reductions in freshwater inputs and increased flooding risks are expected in estuaries.

A prototypical estuary for environmental and human conflicts is the Guadalquivir River estuary (GRE) [2]. The GRE, which is in the southwest of the Iberian Peninsula and flows into the Gulf of Cádiz, is normally subjected to low river discharges. The estuary comprises the last 110 km of the Guadalquivir River, where the first 90 km are navigable. The river has a total length of 680 km and a drainage basin of 63,822 km^2. Between the mid-nineteenth and twentieth centuries, the most important interventions, particularly at the margins, were executed in the GRE. The river was extensively dammed, marshes were occupied, and the course of the estuary was drastically shortened and channelized. Currently, 1.7 million people inhabit the estuary and its surroundings with the people spread over 90 settlements. In the southern part of Spain, where the management is extremely harsh due to the interaction of numerous conflicting stakeholders, the GRE is one of the most important socioeconomic areas. Intensive farming and paddy cultivation, commercial navigation, fisheries and aquaculture, salterns, the Doana National Park, and urban development and associated infrastructures, with an extensive road network and irrigation channels, are the main uses and elements in the estuary.

The recent plan of the Port Authority of Seville to deepen the channel triggered social alarm and created the need for a comprehensive assessment of its potential consequences on physical and ecological dynamics [2, 3]. The amplification in the tidal range is one of the consequences to the deepening ad narrowing of the channel (e.g., as has occurred in other estuaries such as those of the Delaware and Ems rivers [4, 5]). This shift in the tidal motion will lead to a relative increase in high water and a decrease in low water levels and with respects to the former, may improve the navigation conditions. However, this shift may enhance flood risks, lead to a lowering of the groundwater table, increase the suspended sediment matter concentration, and increase salinity intrusion farther upstream (e.g., [6–9]). As a consequence of losing intertidal areas, ongoing deepening may favor flood dominance, while the river's flushing capacity decreases [10]. An increase in tidal currents yields an increase in the suspended sediment concentrations, which in turn reduce the effective hydraulic drag, further increasing the tidal range [11]. This nonlinear feedback between the flow and suspended sediment concentrations cannot be captured with common linear models [12].

Accordingly, the present study addresses the main estuarine processes with the aims to forecast the effects of the expected decrease in the freshwater discharge and the deepening of the navigation channel on the subtidal water levels and currents. Two major estuarine forcings are considered for the simulations: tidal motion and freshwater discharge. Tidal motion is reproduced fairly well with deterministic tidal propagation models. Regarding the freshwater discharges, this work applies a nonstationary statistical approach to represent its stochastic nature. The predictions are performed with common regression models by means of Monte Carlo simulations inspired in the previous works of [8, 13]. The following four specific scenarios are considered: (S1) reference scenario in which freshwater discharge and water depth remain unchanged; (S2) decrease in freshwater inputs, with unchanged water depths; (S3) increase in water depth as a consequence of dredging, whereas the freshwater discharge regime remains unchanged; and (S4) decrease in freshwater discharge and increase of water depth.

The nonstationary statistical approach to account for the river discharge is described in section "River Discharge: Nonstationary Modeling". Discharge and tides were used to predict subtidal water levels and currents for different management scenarios by means of Monte Carlo simulations. Both the scenarios and the implemented regression models are explained in section "Regression Models & Management Scenarios Simulations", which includes a short-term analysis for subtidal elevations as a consequence of a real-case flooding situation. The main attainments of this work are summarized in section "Final Remarks and Conclusions".

River Discharge: Nonstationary Modeling

This characterization aimed to obtain the marginal distribution that best represented the river discharge behavior in the Guadalquivir estuary. This distribution was used to simulate a long-term series of the variable, including its noticeable seasonality. After at stationary analysis (section "Stationary Modeling"), the nonstationary approach developed in [14] was applied (section "Nonstationary Modeling").

Stationary Modeling

Usual and mixture models for the marginal distribution of the river discharge are implemented. The usual distributions were the exponential (EX), lognormal (LN), Weibull (WB), and gamma (GM). Mixture models are intended to incorporate the central and extreme populations into a single model. In these models, the central regime is a truncated distribution, where the upper and lower tails are represented by means of generalized Pareto distribution (GPD). The resulting distributions for analysis are the LN-GPD and the WB-GPD [15]. The parameters of the mixture models are estimated by maximum likelihood. The Kolmogorov–Smirnov test, at a 5% significance level assessed the goodness of fit. To avoid numerical inconsistencies with null discharge (4% of the data) and considering the precision of the measurements (10^{-2}), the time series is uniformly completed with values between 5×10^{-3} and 1×10^{-2} m^3/s.

Figure 1 (left panel) shows the empirical, as well as the modeled cumulative distribution function (cdf) of the river discharge. According to the test, the best fit corresponded to the LN distribution. Figure 1 (right panel) shows the cdf of the LN-GPD and WB-GPD, and the LN-GPD was the better distribution. Comparing the

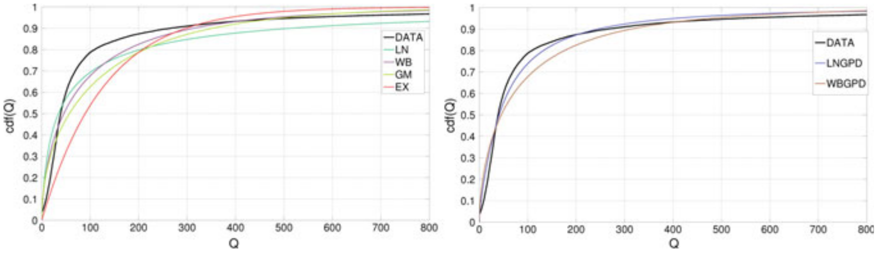


Fig. 1 Cumulative distribution function (cdf) for the river discharge with the usual models (left panel) and the mixture models (right panel) described in section "Stationary Modeling"

best distributions of usual and mixture model, the LN and LN-GPD, respectively, the latter model improved the fit, especially in the tails. Nevertheless, none of the models could capture the trend for low freshwater discharges. The poor fit with usual models and the strong seasonality observed in river discharge justified the use of nonstationary distributions.

Nonstationary Modeling

The nonstationary distribution that best fit the data was the LN-GPD-NE, which included the seasonality in the parameters of the distribution using a Fourier time series. For the discharges in the Guadalquivir, the model with the minimum Bayesian information criterion had an order of approximation for the Fourier series in the parameters (μ_{LN}, σ_{LN}, ξ_2), of (4, 2 ,2). The pdf and cdf considering the seasonality in the parameters of the distribution were also modeled too (Fig. 2). The improvement in the fit with respect to a stationary distribution was noticeable.

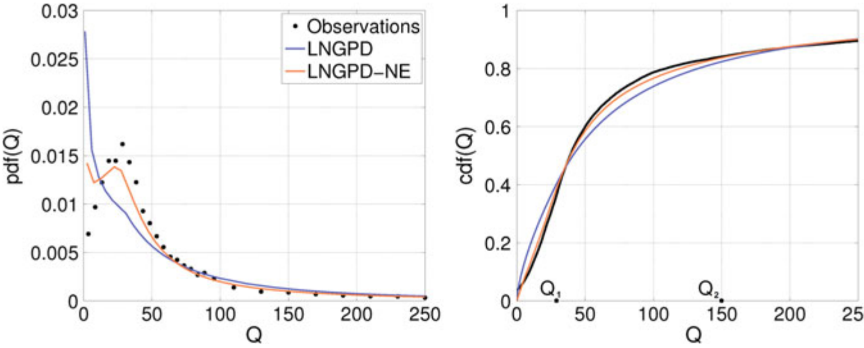


Fig. 2 Pdf (left panel) and cdf (right panel) with the LN-GPD and LN-GPD-NE distributions. Labels Q_1 and Q_2 are for discharge values that correspond to ~33% and ~80% quantiles, respectively

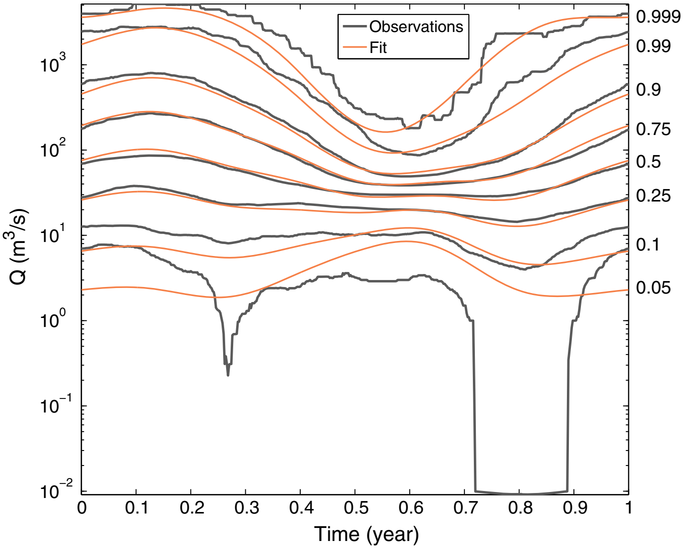


Fig. 3 Iso-probability quantiles for non-exceeding probability (right numbers) equal to 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.99, and 0.999 for empirical (black) data and the LGN-GPD-NE (orange) model

Figure 3 shows the quantiles that corresponded to the empirical accumulated probability values and those obtained when the LGN-GPD-NE was modeled. A moving window of one month was used to obtain the empirical quantiles. The lower most part of the tail was not well reproduced for quantiles ≤0.1 (~10 m^3/s). The reason was that the system was highly regulated, and further considerations should be performed to overcome this situation. Nevertheless, the other quantiles were well reproduced. Thus, this model was selected when performing the simulations in the other sections.

To gain insight into the dynamic effect of different river flow values on subtidal water levels, currents, and into the salinity distribution, two river flows Q_1 and Q_2 that corresponded, to ~33% and ~80% quantiles (right panel of Fig. 3) were selected. For these flows, by means of a one-dimensional model for tidal propagation and salinity distribution [9], the subtidal levels and currents and the subtidal distribution of salinity were determined. As shown in Fig. 4, the higher freshwater flow, Q_2, induced higher currents and residual levels at all locations along the main channel, especially upstream. It is noteworthy that residual levels exceeded 2 m at the head of the estuary. River flows and saline intrusion followed an inverse relationship. The discharge of $Q_2 = 150$ m^3/s almost completely moved the salt intrusion (defined as the kilometric point with 2 psu) to the mouth. The more usual flow of $Q_2 = 30$ m^3/s maintained the intrusion at approximately 70 km.

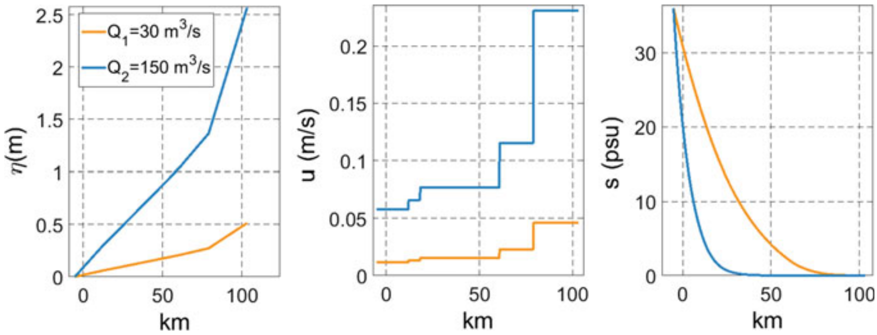


Fig. 4 Subtidal water levels (left panel), currents (central panel), and salinity distributions (right panel) for the discharge values $Q_1 = 30$ m^3/s (orange curves) and $Q_2 = 150$ m^3/s (blue curves) along the main channel from the estuary mouth (km 0)

Regression Models & Management Scenarios Simulations

Regression Models

Three regression models were applied and compared. The regression models corresponded to those detailed in the following: [16] (linear superposition of river discharge Q_d and tidal range H) [17] (accounts for nonlinear interaction between Q_d and H) and mixed partial combination of both).

A regression analysis was carried out for the Guadalquivir estuary. Values were averaged subtidally for a time step of 25 h. Only the currents projected along the channel were fitted. In [16], a linear combination of tidal contribution (tidal range) and water level variation due to river discharge was used.

$$\begin{aligned} \overline{\eta}_G &= s^G_{\overline{\eta},1} H + s^G_{\overline{\eta},2} Q_d + s^G_{\overline{\eta},3} \ , \\ \overline{u}_G &= s^G_{\overline{u},1} H + s^G_{\overline{u},2} Q_d + s^G_{\overline{u},3} \ . \end{aligned} \tag{1}$$

This regression was applied to all the tidal gauges and current meters installed in the Guadalquivir using the local tidal range and the Q_d realized from the Alcalá del Río Dam. Water level and current observations were recorded between 2008 and 2011 by eight tidal gauges, β_i, and six current meters, α_i, located at different stations along the main channel [3]. In [17], the following regression (nonlinear scaling) was used:

$$\begin{aligned} \overline{\eta}_K &= s^K_{\overline{\eta},1} H_0^2 Q_d^{-4/3} + s^K_{\overline{\eta},2} Q_d^{2/3} + s^K_{\overline{\eta},3} \ , \\ \overline{u}_K &= s^K_{\overline{u},1} H_0^2 Q_d^{-4/3} + s^K_{\overline{u},2} Q_d^{2/3} + s^K_{\overline{u},3} \ , \end{aligned} \tag{2}$$

where $s^G_{\overline{\eta},k}$ and $s^K_{\overline{\eta},k}$ in Eqs. 1 and 2 are the fitted coefficients, and H_0 is the tidal range (station β_0). The variables are referred in the same time step.

In these expressions, the exponents correspond to the cited authors. The following expressions, designed as mixed, were proposed to combine both models without indiscriminately increasing the number of parameters:

$$\begin{aligned} \overline{\eta}_M &= s^M_{\overline{\eta},1} H + s^M_{\overline{\eta},2} Q_d + s^M_{\overline{\eta},3} + s^M_{\overline{\eta},4} H^{s^M_{\overline{\eta},5}} Q_d^{s^M_{\overline{\eta},6}} \ , \\ \overline{u}_M &= s^M_{\overline{u},1} H + s^M_{\overline{u},2} Q_d + s^M_{\overline{u},3} + s^M_{\overline{u},4} H^{s^M_{\overline{u},5}} Q_d^{s^M_{\overline{u},6}} \ . \end{aligned} \tag{3}$$

This mixture model was intended to find consistent exponents in the nonlinear term that best represented the GRE. The model performance S_k and correlation coefficient R were the fitted parameters. Figure 5 shows the observed and the predicted subtidal elevations for instruments β_S, β_5, β_1, and β_0. The figure also shows the observed and the predicted subtidal currents for instruments α_5 and α_0.

Overall, the models reproduced the best signal at the landward-most stations. The fit coefficients with Godin and Kukulka formulation were the same order of magnitude among instruments, e.g., $s^G_{\overline{\eta},1}$ for is $\sim 10^{-1}$. This consistency was not observed in the mixed model. The results suggested that despite the improvement in model performance with the mixed model, the understanding of the dynamics from a global point of view was lost. For the sake of simplicity (parsimony principle), the interpretations were performed with the first two models.

The best fit for the elevations was achieved with the model in [16] and mixed models. Focusing on the former and according to the coefficients, with low discharges, the dynamic behavior of the subtidal elevation in the GRE was mostly controlled by the astronomical tidal range. With discharges of approximately around 400 m^3/s and higher, the situation changed. In this case, the results are in agreement with [10], where the subtidal elevations are linearly related to the freshwater discharge. The fit of [17] seemed to be not adequate with small discharges. Since no linear term occurred with the tidal range, the subtidal behavior for the elevations could not be reproduced: important spring–neap variations were dumped by the term $H_0^2/Q^{3/4}$. In addition, note that the model considered the tidal range at the mouth, not locally, which also affected the goodness of fit.

The pdf and cdf for instruments β_6 and β_0 are depicted in Fig. 6. When plotting the probability density and cumulative distribution functions with the observations and the models as shown in Fig. 6, we found that the upper tails were better reproduced than the lower tails. This fact must be considered when evaluating the results of our simulations, for example, if we intend to assess the navigability of the channel.

In an attempt to improve the fit for the low regime, the Markov Chain technique was considered in the regression analysis with [16, 17] and the mixed formulation. The values of the new coefficients, which are not included, presented the same features as those explained before, with the historic or pasttime term the most important. The calibration improved, but the validation was almost the same (values and figure not included).

Although the performance was slightly lower than that without the Markov Chain ($\sim 10^{-2}$), in the validation, the overall performance was similar. The proposed regression models did not significantly improve the fit. The high subtidal elevations and currents time step (25 h), which possibly attenuated or included the past effect, could

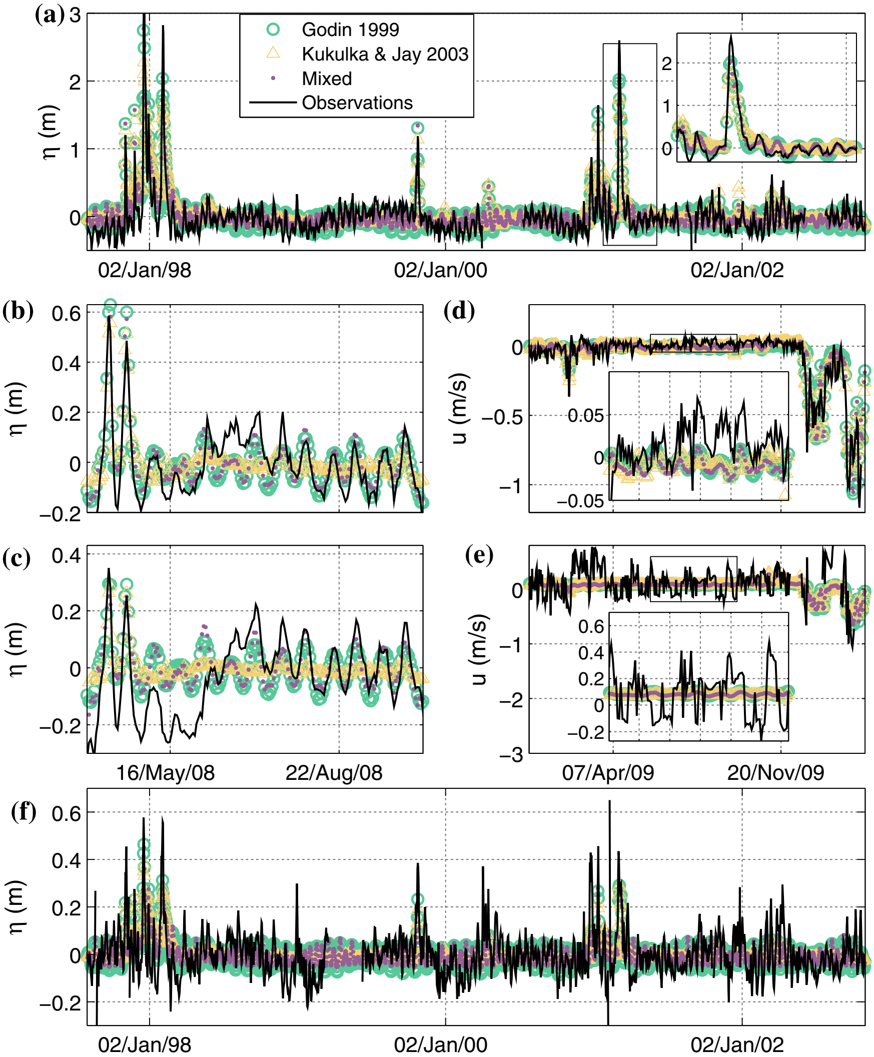


Fig. 5 Panels **a** and **f**: Calibration and validation at β_S and β_0, respectively, of [16] (green circles), [17] (orange triangles) and mixed model (purple dots) regression formulae against subtidal water level obtained from observations (black curve). Panels **b** and **c**: Fit of these models for water levels at β_5 and β_1, respectively. Panels **d** and **e**: Fit of these models for subtidal currents at α_5 and α_0, respectively. Insets zoom into rectangle areas. Symbols are indications of greater than the 95% error confidence interval

explain this fact. In addition, the results were improved by changing the way to consider the inertia or the variables of the models, such as including the wind and other lateral effects as a secondary circulation. Regardless, from this point, the Markov Chain was not considered with subtidal elevations and currents.

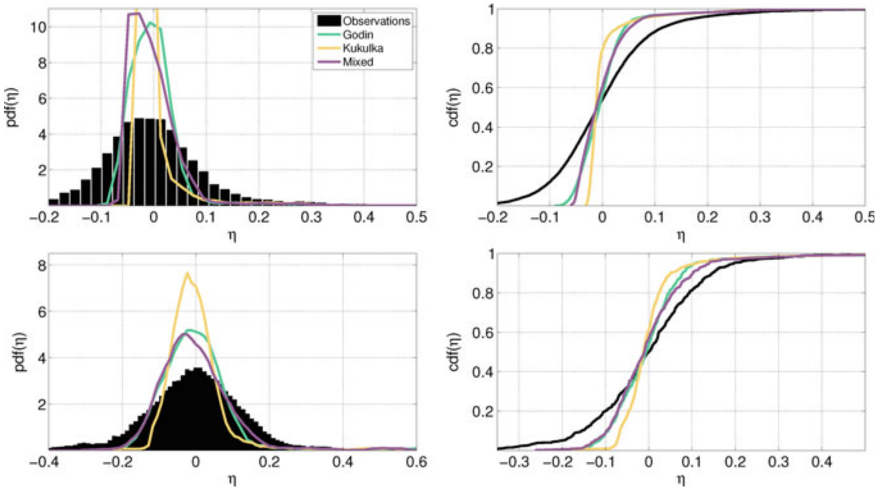


Fig. 6 Pdf and cdf for subtidal elevations at β_6 (first row) and β_0 (second row) for the observed data (black color),[16] (green line), [17] (orange line) and mixed model (purple line)

Management Scenario Simulations

The proposed regression models were used to assess the effects of different scenarios in the subtidal water levels, from short- to mid- temporal scales. The mid-term simulations considered four different scenarios, related to the reduction in the freshwater discharge and the increase in the water depth after a dredging intervention. The short-term simulation assessed the variation in the subtidal level after a peak discharge, supposing both the present state of the system and the response, to the same forcing, after the dredging intervention. The coefficients of the regression models were kept fixed during the simulations.

Mid-term Simulations Subtidal water levels depend on the freshwater discharge and the astronomical tidal range. For the different scenarios, 50 simulations were run, each with a 25 years duration. The river discharges were stochastically simulated with the Monte Carlo method using the nonstationary distribution. A simple one-dimensional tidal model [18] assessed the changes induced by the dredging intervention (rise in the water depth). In this model, friction and the other parameters were supposed to remain as in the present situation. Thus, the shift in the tidal motion was applied by changing the amplitude and phase of the semidiurnal component M_2.

The considered scenarios were as follows:

- S1: Similar discharge (Q_d) distribution with the current water depth (h), ($\sim Q_d$, ∼h).
- S2: Decrease in the freshwater discharge by 15%, according to [19]; the water depth did not change ($\downarrow Q_d$, ∼h).

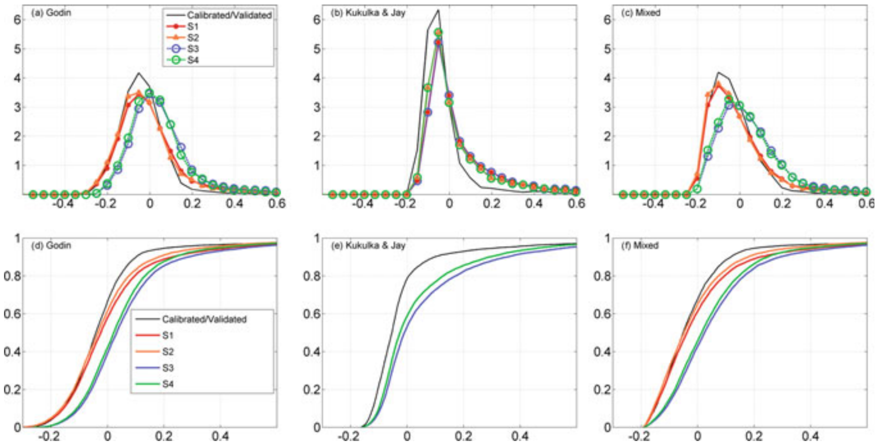


Fig. 7 Pdf (first row) and cdf (second row) for subtidal elevations at β_S for the calibration and validation period (black line) and scenarios S1 (red), S2 (orange), S3 (blue), and S4 (green) for the regression models. In [17] (second column), S1 overlaps S3, and S2 does the same with S4

- S3: Increase in the water depth, from 7 to 8.5 m, according to the Port Authority of Seville; the freshwater regime did not change ($\sim Q_d$, ↑h).
- S4: Combination of S2 and S3 (↓ Q_d,↑h).

Figure 7 depicts the pdf and cdf at β_S with the different models.

The model in [16] and the mixed model, despite different values, presented similar behavior (Fig. 7 panels a-c, d-f). As expected with these models, for the S1 (red line) the observed values (black line) at this instrument were better reproduced than those with the model in [17]. With this formulation, S1 overlapped S3, and S2 and S4 overlapped. The reason for this result was that the authors built their model with the astronomical tidal range at the mouth station. Since the one-dimensional tidal model fixed the harmonics in the mouth after the dredging, the model in [17] could only capture the variation in the freshwater regime. In all scenarios with all models except with the mixed model, subtidal water level distributions did not have marked changes in the dispersion and behavior of the tails in comparison to those of the simulation of the present state S1.

Figure 8 displays the mean values of the subtidal levels for each scenario with the regression models. Comparing the scenarios with the current simulated situation of S1, we observed that the subtidal levels decreased when the freshwater input was reduced (S2), and increased as a response to deepening the navigation channel (S3). The mixed scenario (S4) values were between those of S2 and S3. According to the proposed values for S4 and in comparison to those for S1, it was possible to locate where the subtidal elevations decreased (lower stretch) or increased (upper stretch).

Short-Term Simulations In this section, the effects of flooding were assessed on the subtidal water levels. The purpose was twofold: first, to show that the presented regression models could be used as an early warning system, since managers can

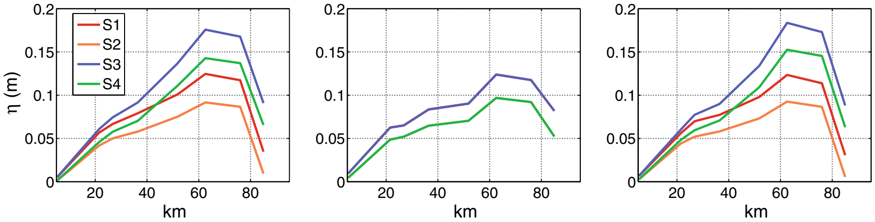


Fig. 8 Along channel representation of the mean subtidal levels with the different scenarios (S1, S2, S3, and S4) with the formulation of [16] (left panel), [17] (central panel) and mixed (right panel)

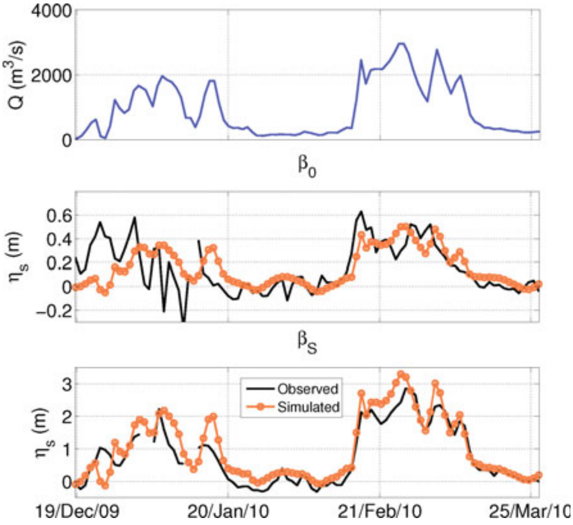


Fig. 9 River discharge (upper panel) and observed (black) and modeled (orange) subtidal elevations at β_0 (middle panel) and β_S (lower panel)

control the river discharge released by the dam and second, to evaluate the response of the estuary against a flood with or without the dredging intervention. The model in [16] was used in this section.

The first step was to test the model against a high discharge with as many instruments as possible. The high discharge corresponded to the discharges of April 2008 with a peak value of 550 m^3/s. The global coefficient of determination R^2, was 0.67, using all the stations. The second step was to explore the along-estuary response with a higher discharge. The selected period corresponded to the winter of 2009/2010, as depicted in Fig. 9. Only β_0 and β_S recorded the elevations. In this case, the fit improved, with a global R^2 of 0.81. Finally, we focused on a second period with a high discharge, from February 15 to March 13, 2010. The recorded subtidal elevations at β_S were higher than 0.4 m. The performance of the observations and the predictions showed an $R^2 = 0.87$. Thus, the regression model could fairly predict the subtidal elevations in flooding. From a management point of view and delimiting, beforehand, the value of the discharge, the model can be used to develop a plan of action against flooding in the riverine communities 25 h in advance.

We chose this latter discharge to fulfill the short-term analysis. For this analysis, the objective was the assessment of what would have happened if the depth of the navigational channel had been deeper. In this case, modifying the harmonics was used to introduce the shift, the differences in the subtidal elevations between the current situation and the increased depth scenario were plotted, and the situation worsens in terms of safety upstream, from pk 35 to the weir. Note this case was a subtidal assessment of 25 h. Most likely, the results after accounting for higher frequency motions (discharge regarded as a 'bore') would be more adverse.

Final Remarks and Conclusions

Linear and nonlinear regression models were applied to characterize subtidal water levels and currents, based on tidal motion and freshwater discharge variables.

whereas tidal motion was reproduced with deterministic models, the seasonality of the freshwater discharge in the Guadalquivir was characterized by a nonstationary mixture distribution model, with a lognormal distribution for the central regime and a generalized Pareto distribution for both upper and lower tails. Though the lower most part of the discharge distribution was not captured due to the regulation of the dam, quantiles greater than 0.1 were well reproduced.

The best fit for elevations in the Guadalquivir estuary corresponded to a linear superposition of tidal range and discharge, while the best fit for currents included nonlinear interaction. The regression models captured the elevations better than they captured the currents. The elevation signal was best reproduced at the landward-most stations. An analysis of the subtidal level distribution revealed that the upper tails were better modeled than the lower tails, which is a relevant fact when assessing the navigability in the channel. Markov Chain processes were analyzed, but not considered in the simulations, because although the performance of the calibration notably improved with the Markov process, the validation remained the same.

The obtained relationships were used to predict on a medium-term basis the effects of a freshwater discharge decrease by 15% for the next years and the deepening of the navigation channel by increasing the water depth 23%. Monte Carlo techniques, with regression models, were used for subtidal water level simulations. During the simulations, the coefficients that related the processes within the models were fixed. The shifts associated with the forcings were introduced in their distribution. A new nonstationary distribution was computed for the reduction in freshwater discharge. The shifts associated with the deepening were introduced by changing the amplitude and phase of the semidiurnal component M2 through a one-dimensional tidal model. Considering the subtidal elevation simulations, the comparison between the present conditions and the scenarios revealed that subtidal levels decreased when the freshwater input was reduced and increased as a consequence of deepening the channel. A short-term scenario was studied to assess the response of the system in the case of flooding. According to the results, the increase in the subtidal elevations was the most significant from 35 km upstream to the dam. Overall, these simulations contribute to a better understanding of the subtidal water level distributions with tidal motion and freshwater discharge.

Acknowledgements The authors acknowledge support from project PIRATES (Ref. CTM2017-89531-R): Programa Estatal de Investigación, Desarrollo e Innovación orientada a los RETOS de la sociedad.

References

1. Dalrymple, R.W., Zaitlin, B.A., Boyd, R.: Estuarine facies models: conceptual basis and stratigraphic implications: perspective. J. Sediment. Res. **62**(6), 1130–1146 (1992)
2. Ruiz, J., Polo, M. J., Dez-Minguito, M., Navarro, G., Morris, E., Huertas, E., Caballero, I., Contreras, E., Losada, M. Á.: The Guadalquivir estuary: a hot spot for environmental and human conflicts, In: Finkl, C.W., Makowski, C. (Eds.), Environmental Management and Governance, Vol. 8 of Coastal Research Library, pp. 199–232. Springer (2015)
3. Navarro, G., Gutierrez, F.J., Díez-Minguito, M., Losada, M.Á., Ruiz, J.: Temporal and spatial variability in the Guadalquivir estuary: a challenge for real-time telemetry. Ocean Dyn. **61**(6), 753–765 (2011)
4. DiLorenzo, J.L., Huang, P., Thatcher, M.L., Najarian, T.O.: Dredging impacts on delaware estuary tides. In: Estuarine and Coastal Modeling, pp. 86–104. ASCE (1993)
5. de Jonge, V.N., Schuttelaars, H.M., van Beusekom, J.E., Talke, S.A., de Swart, H.E.: The influence of channel deepening on estuarine turbidity levels and dynamics, as exemplified by the Ems estuary. Estuar. Coast. Shelf Sci. **139**, 46–59 (2014)
6. Winterwerp, J.C., Wang, Z.B.: Man-induced regime shifts in small estuaries—I: theory. Ocean Dyn. **63**(11), 1279–1292 (2013)
7. van Maren, D.S., Winterwerp, J.C., Vroom, J.: Fine sediment transport into the hyper-turbid lower Ems River: the role of channel deepening and sediment-induced drag reduction. Ocean Dyn. **65**(4), 589–605 (2015)
8. Reyes-Merlo, M.Á., Díez-Minguito, M., Ortega-Sánchez, M., Baquerizo, A., Losada, M.Á.: On the relative influence of climate forcing agents on the saline intrusion in a well-mixed estuary: medium-term Monte Carlo predictions. J. Coast. Res. **65**, 1200–1205 (2013)
9. Siles-Ajamil, R., Díez Minguito, M. and Losada, M. Á.: Tide propagation and salinity distribution response to changes in sea water depth and channel network in the Guadalquivir River Estuary: an exploratory model approach. Submitted to Ocean and Coastal Management (2019)
10. Díez-Minguito, M., Baquerizo, A., Ortega-Sánchez, M., Navarro, G., Losada, M.Á.: Tide Transformation in the Guadalquivir Estuary (SW Spain) and Process-Based Zonation. J. Geophys. Res. Oceans **117** (2012)
11. Wang, Z.B., Winterwerp, J.C., He, Q.: Interaction between suspended sediment and tidal amplification in the Guadalquivir Estuary. Ocean Dyn. **64**(10), 1487–1498 (2014)
12. Losada, M.Á., Díez-Minguito, M., Reyes-Merlo, M.Á.: Tidal-fluvial interaction in the Guadalquivir River Estuary: spatial and frequency-dependent response of currents and water levels. J. Geophys. Res. Oceans **122**(2), 847–865 (2017)
13. Guerra-Chanis, G.E., Reyes-Merlo, M.Á., Dez-Minguito, M., Valle-Levinson, A.: Saltwater intrusion in a subtropical estuary. Estuarine Coast. Shelf Sci. **217**, 28–36 (2019)
14. Solari, S., Losada, M.Á.: Non-stationary wave height climate modeling and simulation. J. Geophys. Res. Oceans **116** (2011)
15. Solari, S.: Metodologías de Simulación de Agentes Naturales y Desarrollo de Sistemas. Modelo de Verificación y Gestión de Terminales Portuarias. Aplicación al Puerto de la Bahía de Cádiz. Ph.D. thesis. University of Granada, Spain (2011)

16. Godin, G.: The propagation of tides up rivers with special considerations on the Upper Saint Lawrence River. Estuarine Coast. Shelf Sci. **48**(3), 307–324 (1999)
17. Kukulka, T., Jay, D.A.: Impacts of columbia river discharge on Salmonid habitat: 2. Changes in Shallow-Water Habitat. J. Geophys. Res. Oceans 108.3294 (C9) (2003)
18. Prandle, D., Rahman, M.: Tidal response in Estuaries. J. Phys. Oceanogr. **10**(10), 1552–1573 (1980)
19. CEDEX: Evaluación del Impacto del Cambio Climático en los Recursos Hídricos en Régimen Natural. Tech. rep. Centro de Estudios y Experimentacin de Obras Públicas, p. 22 (2011)

Spatial Distribution of Climatic Cycles in Andalusia (Southern Spain)

J. Sánchez-Morales, E. Pardo-Igúzquiza and F. J. Rodríguez-Tovar

Abstract Several climatic cycles in Andalusia (southern Spain) have been identified by using precipitation and temperature data from most of the twentieth and the early twenty-first centuries at 707 meteorological stations. Some of the cycles detected had been recognized in previous studies, such as the 3-year cycle and the 7/8-year cycle, which were the most common periodicities across the study area. Spectral analysis was used for statistical analysis. The power spectrum estimator used is the smoothed Lomb–Scargle periodogram. The results reveal very interesting spatial patterns that had not been seen before in previous climatic studies, which illustrate a combined and complex influence of the North Atlantic Oscillation and the Mediterranean Oscillation. In general, the precipitation record studied presents better results than the temperature record, which offers less clarity in assessing climatic variability in Andalusia. Nevertheless, most of the cycles identified in the precipitation record were detected in the temperature record as well.

Keywords Power spectrum · Climate · North Atlantic Oscillation · Mediterranean Oscillation · Andalusia · Spain

J. Sánchez-Morales (✉) · F. J. Rodríguez-Tovar
Universidad de Granada, Avd. Fuentenueva s/n, 18071 Granada, Spain
e-mail: josesanmor@correo.ugr.es; jose.chez.morales@gmail.com

F. J. Rodríguez-Tovar
e-mail: fjrtovar@ugr.es

E. Pardo-Igúzquiza
Instituto Geológico y Minero de España, Ríos Rosas 23, 28003 Madrid, Spain
e-mail: e.pardo@igme.es

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_17

Introduction

Andalusia (southern Spain) is a region characterized by huge climatic contrasts, i.e. the 'Sierra de Grazalema' in the southwest is the wettest place on the entire Iberian Peninsula with average rainfall of over 2000 mm/year, whereas the 'Desierto de Tabernas' in the southeast is considered the driest place in Continental Europe with less than 150 mm/year on average. The influence of both the Atlantic Ocean and the Mediterranean Sea on this area of 87,597 km^2, plus the presence of the Betic Cordillera with altitudes above 3000 m.s.l. (metres above sea level), make this region unique from a climatic point of view and very interesting for analysing the evolution of climate from past to present times, especially during the most recent periods. Thus, Andalusia could be seen as a natural laboratory for the study of climatic changes from past to present.

Climatic studies from the region of Andalusia are frequent and focus on a variety of aspects. Thus, several statistical techniques and methodologies have been used, i.e. principal component analysis [1], empirical orthogonal function [2], innovative missing values estimator [3], non-instrumental climate reconstruction [4], and gridded dataset and combined indices evolution [5], amongst many others.

The causes of climate variability in Andalusia at annual, inter-annual, decadal and multi-decadal timescales are generally associated with diverse phenomena, involving several climatic subsystems, revealing the interactions between the atmosphere and ocean [1–5]. In some cases, a cyclic pattern in climatic variability has been interpreted. The identification of climatic cycles in Andalusia by means of spectral analysis has been carried out in previous studies although the method differed according to the specific study [6, 7]. On this basis, this study examines climatic evolution during the twentieth and twenty-first centuries in Andalusia, based on the spectral analysis of data from different climatic proxies, in order to evaluate the cyclic nature of the said evolution and to interpret the processes involved.

Methodology

This study uses spectral analysis as a statistical technique to evaluate the importance of the frequencies associated with precipitation and temperature time series in Andalusia (see meteorological datasets below). The power spectrum estimator used is the smoothed Lomb–Scargle periodogram [8–10], which works directly with uneven time series, such as the annual precipitation and/or annual temperature series in the study area. The technique evaluates the statistical significance of the peaks using the Monte Carlo permutation test, as neighbouring frequencies are highly correlated, and then it adjusts statistical significance by smoothing the periodogram [10]. Linear smoothing with three terms was applied to the raw periodogram. The output consists of the Lomb–Scargle spectrum, the achieved confidence level spectrum, the mean spectrum of permutations and the phase spectrum.

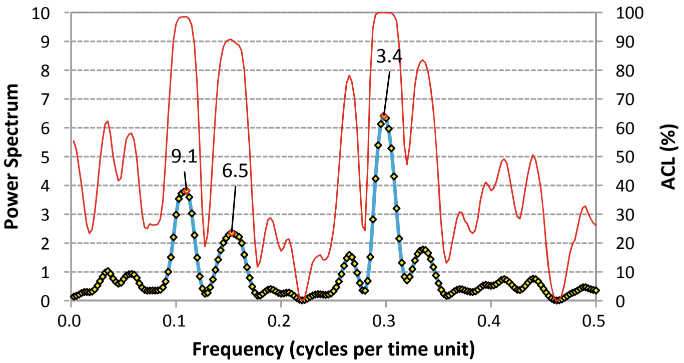


Fig. 1 Example of the power spectrum for annual precipitation at station P6289, and associated peaks of 9.1, 6.5 and 3.4 years above Achieved Confidence Level (ACL) of 90%, by using the smoothed Lomb–Scargle periodogram

The parameters required have been optimized for dealing with the annual precipitation and/or temperature time series in the study area, and for achieving the intended goal of capturing climatic cycles with a duration just above the sampling interval (i.e. biannual oscillation). Thus, 0.5 has been used as the highest frequency to evaluate, 200 as the number of frequencies in the interval, 2000 as the number of permutations, 75,654 as the random seed, 3 as the number of smoothing terms and linear smoothing was enabled. To illustrate the above process, the output of this methodology from one of the precipitation stations is presented below (Fig. 1).

Meteorological Datasets

The precipitation and temperature data were collected from two different sources and named chguadalquivir [11] and aemet [12]. The sources differ from each other with regard to the sampling interval; datasets from chguadalquivir were available monthly and covered the period from 1951 to 1987, whereas datasets from aemet were available daily and covered the period from 1901 to 2012. There were 1574 precipitation station and 526 temperature station datasets from chguadalquivir, spread across the entire study area. There were 595 precipitation station and 282 temperature station datasets from aemet, all located in the eastern half of the study area.

For a better comparison, all the datasets have been converted into annual datasets. As a first step, the daily datasets were summarized into monthly datasets; only months with a minimum of 25 days of precipitation measurements and/or a minimum of 20 days of temperature measurements were considered. Then, to convert the monthly datasets into annual datasets, only complete years and/or years that had 12 months of measurements were used. A second level of filtering was also carried out in which precipitation stations with less than a total of 20 years of records and temperature

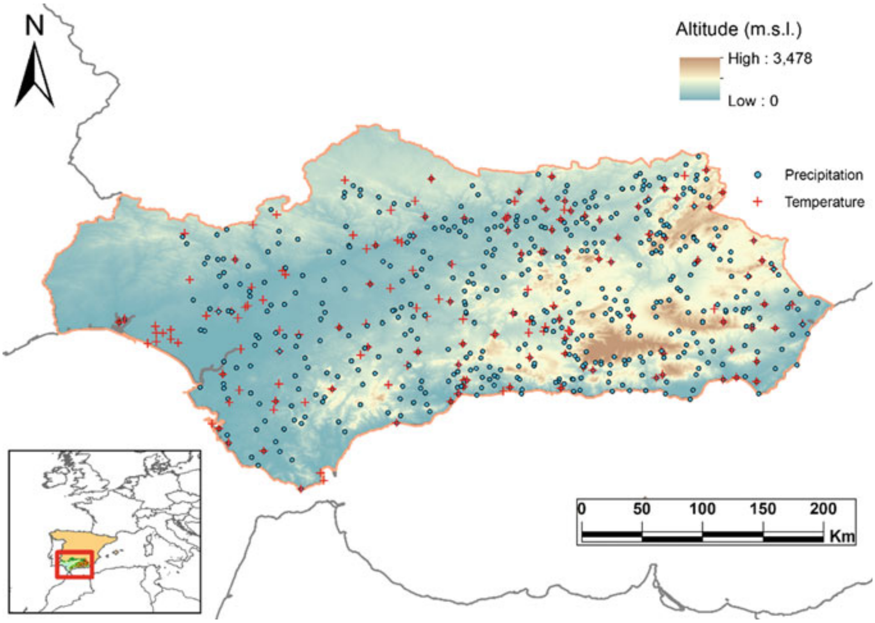


Fig. 2 Distribution of the meteorological stations selected for the spectral analysis

stations with less than a total 10 years of records were excluded from the analysis. It was not a requirement for that the total amount of years with records to be consecutive, as this spectral analysis methodology can deal with uneven series. Upon combining the two data sources, a comparison method was established to see which one had more years of information for the same station, as various stations were in both data sources. If the length of the series in years was the same in both sources, preference was given to the dataset from aemet, which was originally compiled daily.

The filtering process produced 547 precipitation stations and 160 temperature stations: a total of 707 meteorological datasets to be analysed (Fig. 2).

Results

The spectral analysis carried out on the 707 datasets detected 1751 significant peaks in the precipitation datasets (Fig. 3) and 466 significant peaks in the temperature datasets (Fig. 4), all above or equal to 90% of ACL. All the cycles detected for the same meteorological variable have been plotted in the same diagram.

The number of cycles detected is lower in the temperature record than in the precipitation record, but it is very remarkable that both temperature and precipitation variables show signals at the same frequencies. The 3-year cycle and the 7/8-year

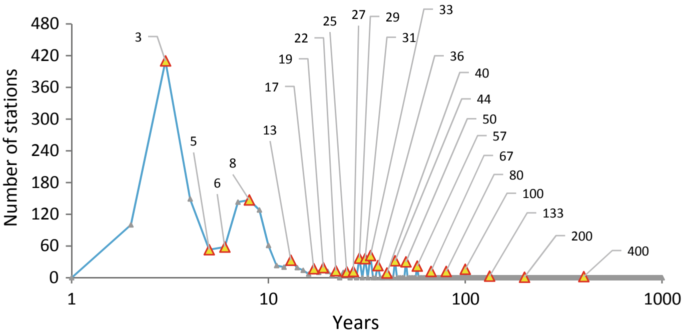


Fig. 3 Diagram containing all the precipitation cycles in the study area and the number of stations in which the cycles were detected (yellow triangles represent the value of the cycle in years)

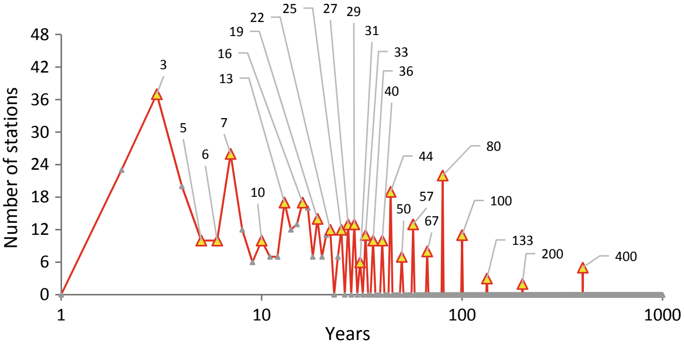


Fig. 4 Diagram containing all the temperature cycles in the study area and the number of stations in which the cycles were detected (yellow triangles represent the value of the cycle in years)

cycle were the most frequently detected, and show the highest significance. Thus, a detailed analysis of cycles at both 3 years and 7/8 years has been conducted.

All precipitation (344) and temperature (28) stations showing peaks between frequency values of 0.4 (2.5 years) and 0.29 (3.5 years) have been collected and plotted on a map to see their spatial distribution (Fig. 5). The same output has been conducted for the 7/8-year cycle by isolating all peaks between 0.154 (6.5 years) and 0.118 (8.5 years), 223 precipitation stations and 23 temperature stations (Fig. 6).

Overall (Figs. 5 and 6), the spatial distribution of temperature observed does not show any geographical predominance of one cycle over the other, as opposed to the maps showing the precipitation cycles observed, where a general trend can be inferred. In the latter, there are more 3-year cycles in the east and more 7/8-year cycles in the west. The two cycles have been combined into one map by using all the precipitation stations in which the two cycles were detected together (Fig. 7). This process was based on a preliminary selection of stations in which at least one of the two cycles must be present at each station; if both cycles were detected in the same

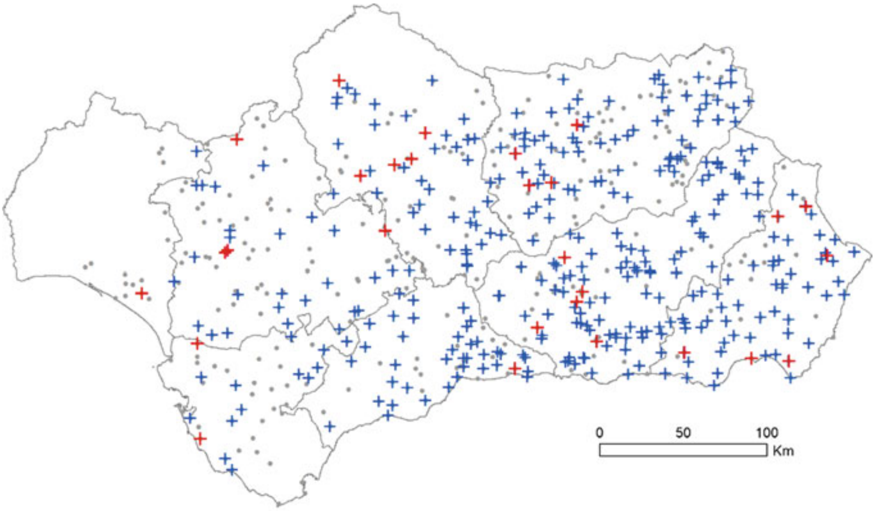


Fig. 5 Spatial distribution of all precipitation and temperature stations in which the 3-year cycle was detected (crosses in blue for precipitation and red for temperature) above 90% of ACL, and all the other stations used for the analysis (points in grey)

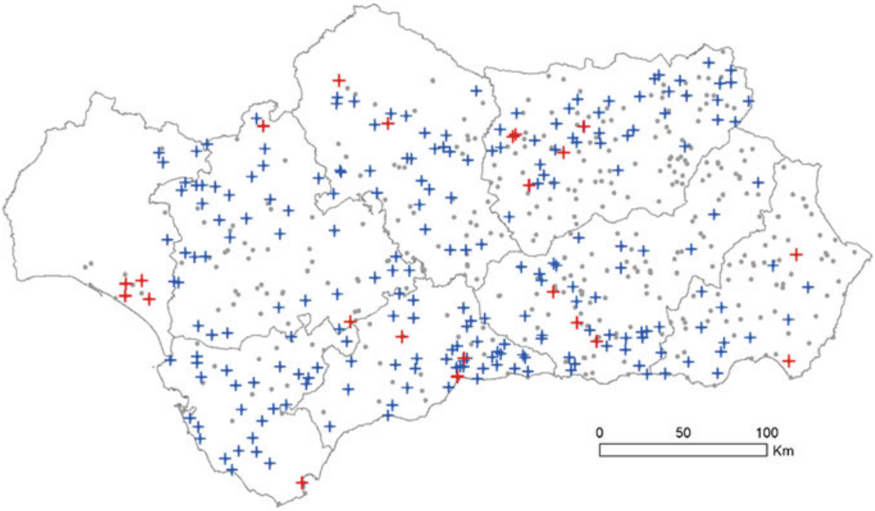


Fig. 6 Spatial distribution of all precipitation and temperature stations in which the 7/8-year cycle was detected (crosses in blue for precipitation and red for temperature) above 90% of ACL, and all the other stations used for the analysis (points in grey)

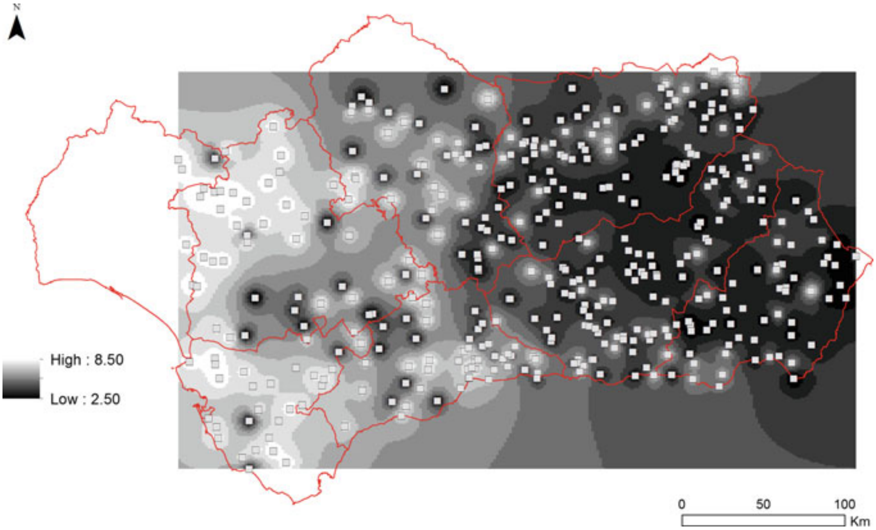


Fig. 7 Spatial interpolation on 423 precipitation stations showing the geographical influence in Andalusia on the 3-year cycle and the 7/8 year cycle. Values from 2.50 to 8.50 in years

station, the cycle with more power spectrum was the one counted for that station. Thus, 423 precipitation stations resulting from the aforementioned conditions have been analysed. About 261 precipitation stations show a predominance of the 3-year cycle and 162 precipitation stations show a predominance of the 7/8-year cycle. This process was followed by an interpolation process in which the ‘Inverse Distance Weighting’ technique was applied. The parameters were ‘2’ for the power and ‘5’ for the maximum number of neighbours.

Interpretation

The causes of climate variability in Andalusia from annual to multi-decadal timescales have generally been associated with Atlantic Ocean phenomena: large scale circulation features of Western Europe and the Atlantic Ocean [1], alternation of zonal circulation and meridional circulation in the Atlantic that shifts the Azores High [2], persistency and displacement of the Azores High [3], and changes in North Atlantic Oscillation (NAO) phases [4]. However, Eastern Andalusia is less influenced by Atlantic air masses and climate variability there is also influenced by Mediterranean Sea dynamics [5].

Previous studies applying spectral analysis to climatic datasets reveal cyclic climatic variability in the range of 2–250 years, but mainly located in the range of 2–11 years. Thus, a study on rainfall variability in Southern Spain [6] found peaks above 95% significance at 2.1, 3.5, 7–9, 16.7 and 250 years, and a previous study on hydraulic heads across the Vega de Granada aquifer [7] found a decadal cycle (peaks between 8 and 11 years) and a 3.2-year cycle, amongst others.

The power spectrum has been estimated for both North Atlantic Oscillation and Mediterranean Oscillation indexes (MOI) (Figs. 8 and 9). The NAO index is available at a daily resolution [14] from 1950 to 2017. The MOI has two versions [15], 'Algiers-Cairo' (MOAC) and 'Israel-Gibraltar' (MOIG), both from 1948 to 2016 at daily resolution. The NAO and MOIG indexes have been analysed by using annual series.

The NAO index has significant periodicities at 50, 13.7, 2.7, 2.5 and 2.1 years, and the MOIG index at 7.8, 6.5, 4.1, 3.4, 2.4 and 2.1 years, all above 90% of ACL.

A second exercise has been conducted to assess the correlation between all the monthly series from the 707 stations (547 precipitation stations and 160 tempera-

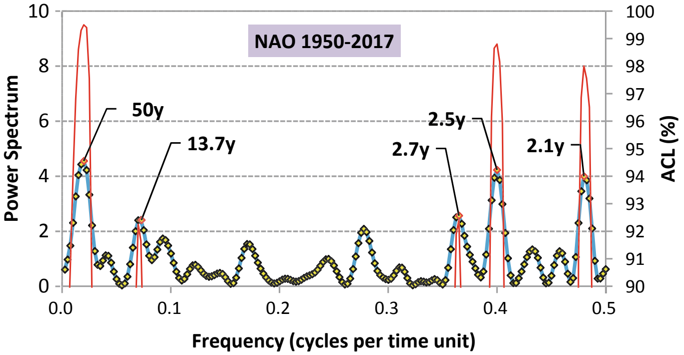


Fig. 8 Power spectrum of the NAO index using annual records derived from the original dataset, and associated peaks of 50, 13.7, 2.7, 2.5 and 2.1 years above ACL of 90%, by using the smoothed Lomb–Scargle periodogram

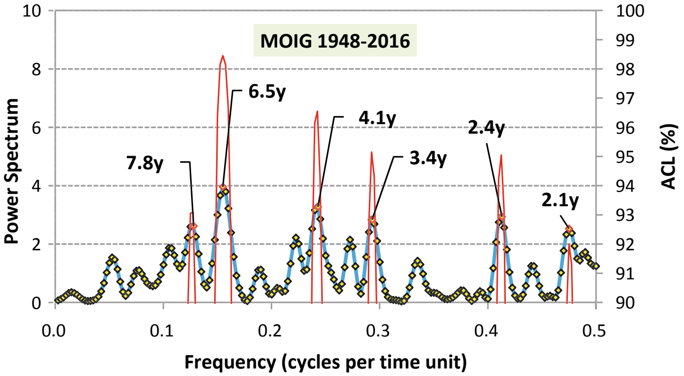


Fig. 9 Power spectrum of the MOIG index using annual records derived from the original dataset, and associated peaks of 7.8, 6.5, 4.1, 3.4, 2.4 and 2.1 years above ACL of 90%, by using the smoothed Lomb–Scargle periodogram

ture stations) and the NAO and MO indexes. The geographical influence of both oscillations has been assessed (Figs. 10, 11, 12 and 13). The NAO correlation for precipitation (Fig. 10) is low; it tends to be more negative towards the west (−0.343) and zero or slightly positive towards the east (0.067). The NAO and temperature correlation (Fig. 11) is less clear, with values between −0.214 and 0.240. The MOIG and precipitation correlation (Fig. 12) is greater in value (from −0.16 to −0.76) than that of the NAO and is always negative; it shows a pattern of being more negative towards the west (−0.753) and less negative towards the east (−0.240). The correlation between MOIG and temperature (Fig. 13) is always positive, with values between 0.386 and 0.643, and it is equally distributed across the study area.

The most significant frequencies associated with both the NAO and MOIG indexes have been combined, as have the power spectra frequencies detected above 90% of ACL (Figs. 14 and 15). Certain precipitation and temperature frequency values match some of the frequencies associated with NAO and MOIG quite well.

Particularly interesting are the precipitation cycles that correspond to the MOIG cycles of 3.4, 6.5 and 7.8 years, and to the NAO cycles of 2.1, 2.7, 14.3 and 50 years. The match between the temperature cycles recorded and the NAO and MOIG cycles is less clear than the precipitation comparison.

Other cycles detected in the study area, which were also above 90% of ACL, could be tentatively associated with other climatic phenomena (Table 1; [13] for a review).

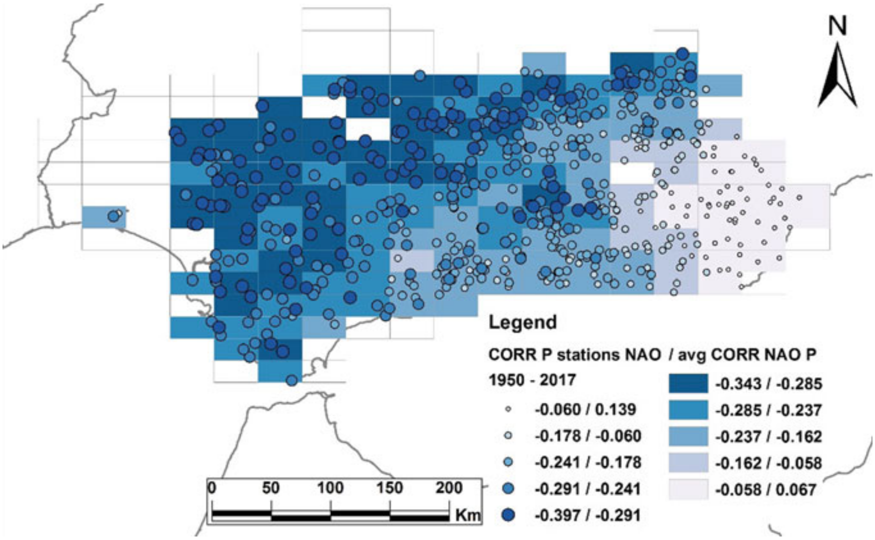


Fig. 10 Correlation coefficients between the NAO index monthly values (1950–2017) and the monthly values of the 547 precipitation stations. The rectangles represent the average value for that particular area

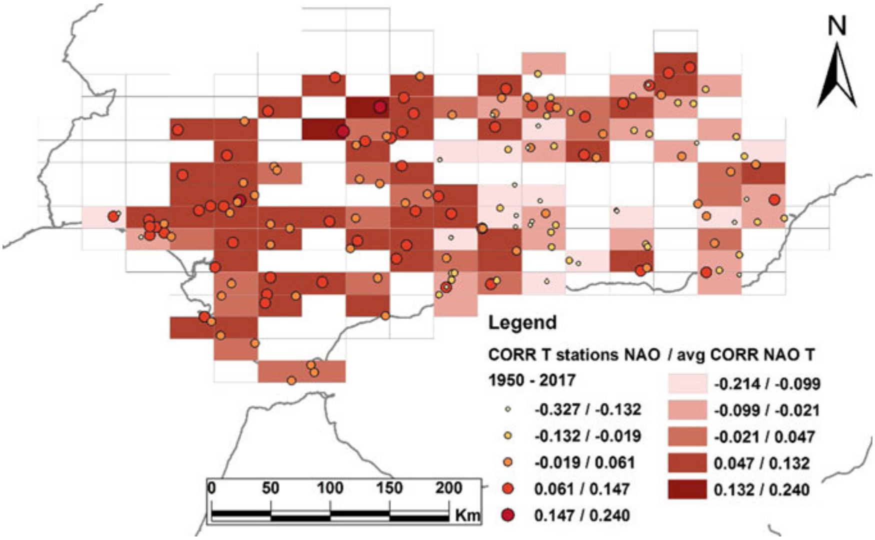


Fig. 11 Correlation coefficients between the NAO index monthly values (1950–2017) and the monthly values of the 160 temperature stations. The rectangles represent the average value for that particular area

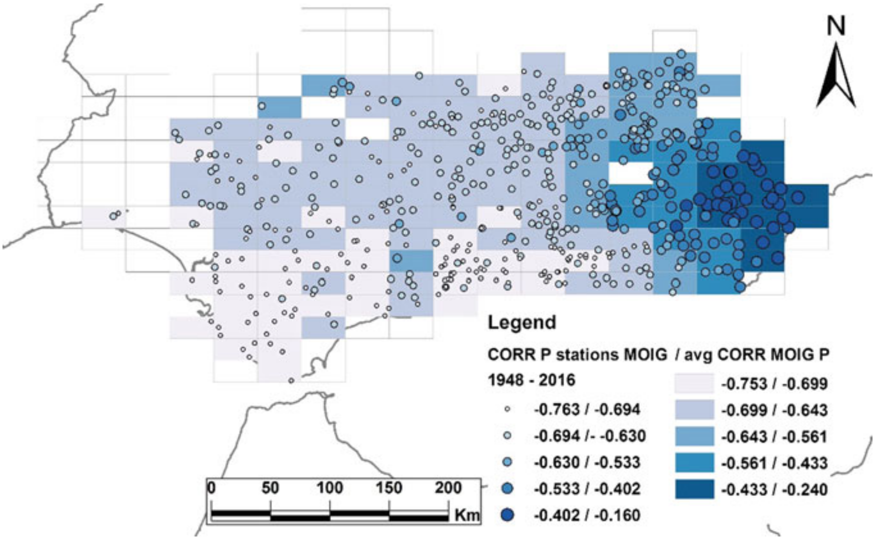


Fig. 12 Correlation coefficients between the MOIG index monthly values (1948–2016) and the monthly values of the 547 precipitation stations. The rectangles represent the average value for that particular area

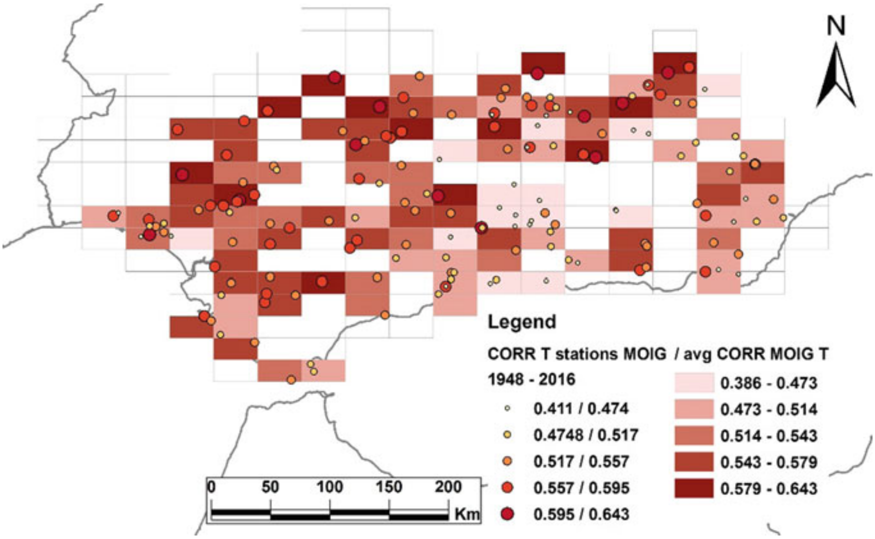


Fig. 13 Correlation coefficients between the MOIG index monthly values (1948–2016) and the monthly values of the 160 temperature stations. The rectangles represent the average value for that particular area

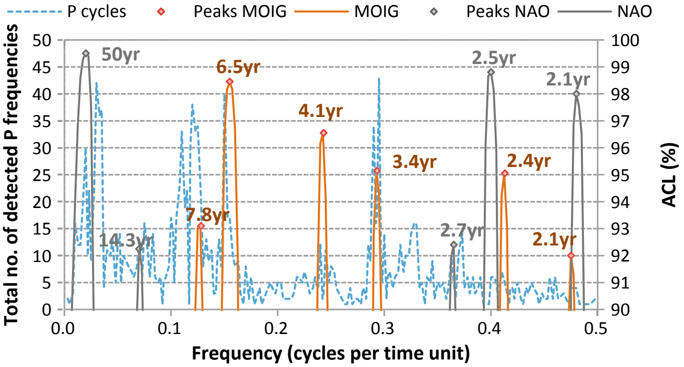


Fig. 14 Histogram of frequencies detected above ACL 90% from all precipitation stations, together with the most significant frequencies of the NAO and MOIG indexes

Conclusions

Spectral analysis of precipitation and temperature time series from a considerable number of meteorological stations (707 locations) distributed across southern Spain in the region of Andalusia has allowed for the characterization of the spatial variability of the two most frequent cycles in the study region: a 7/8-year cycle and a 3-year cycle. The length of time over which data was collected at the stations analysed was

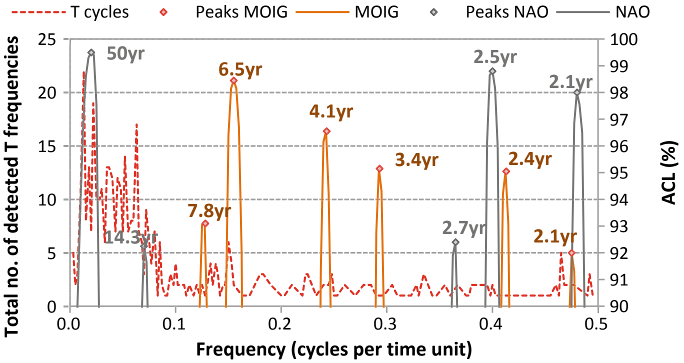


Fig. 15 Histogram of frequencies detected above ACL 90% from all temperature stations, together with the most significant frequencies of the NAO and MOIG indexes

Table 1 Tentative correlation between the other cycles detected above 90% ACL and other well known cycles, including the number of precipitation (P) and temperature (T) stations where the said cycles were detected

Cycle value in years	Cycle	P	T
10/11	Sunspot cycles of 11 years	84	17
17	Southern Oscillation Index (SOI)	17	16
19	Lunisolar cycle	19	14
20–25	Hale cycle	51	49

relatively short (in most cases less than 50 years, but always more than 20 years for precipitation and 10 years for temperature) and therefore the focus of this research has been the high-frequency cycles. Many other periodicities greater than 10 years have been found with an ACL higher than 90%, but they are less abundant. Initially, it was thought that the western part of the study area might be more influenced by the climatic activity derived from the North Atlantic Oscillation (NAO), manifesting as a cycle in the range from 7 to 8 years. On the other hand, the eastern part of the study area was hypothesized to be more influenced by the Mediterranean Oscillation (MO), represented by a 3-year cycle. However, the correlation values calculated across the study area between the meteorological variables on the one hand, and the NAO and MOIG indexes on the other, plus the spectral analysis conducted, show that the Mediterranean Oscillation may play a greater role in the region than the North Atlantic Oscillation. Thus, the 7/8-year cycle may correspond to the 6.5- and 7.8-year cycles detected in the MOIG index, and the 3-year cycle may be associated with the 3.4-year cycle detected in the MOIG index, and with the 2.7- and 2.5-year cycles detected in the NAO index. The explanation of how these two climatic phenomena interact with one another is outside the scope of this paper and more research is needed. Nevertheless, these results can help meteorologists and climatologists to better understand how weather and climate work in the southern Iberian Peninsula.

Acknowledgements This work has been supported by research project CGL2015-71510-R from the Spanish Ministry for the Economy, Industry and Competitiveness.

References

1. Esteban-Parra, M.J., Rodrigo, F.S., Castro-Díez, Y.: Spatial and temporal patterns of precipitation in Spain for the period 1880–1992. Int. J. Climatol. **18**(14), 1557–1574 (1998)
2. Rodrigo, F.S., Esteban-Parra, M.J., Pozo-Vázquez, D., Castro-Díez, Y.: A 500-year precipitation record in Southern Spain. Int. J. Climatol.: J. R. Meteorol. Soc. **19**(11), 1233–1253 (1999)
3. Ramos-Calzado, P., Gómez-Camacho, J., Pérez-Bernal, F., Pita-López, M.F.: A novel approach to precipitation series completion in climatological datasets: application to Andalusia. Int. J. Climatol. **28**(11), 1525–1534 (2008)
4. Rodrigo, F.S., Gómez-Navarro, J.J., Montávez-Gómez, J.P.: Climate variability in Andalusia (southern Spain) during the period 1701–1850 based on documentary sources: evaluation and comparison with climate model simulations. Clim. Past **8**(1), 117–133 (2012)
5. Fernández-Montes, S., Rodrigo, F.S.: Trends in surface air temperatures, precipitation and combined indices in the southeastern Iberian Peninsula (1970–2007). Climate Res. **63**(1), 43–60 (2015)
6. Rodrigo, F.S., Esteban-Parra, M.J., Pozo-Vázquez, D., Castro-Díez, Y.: Rainfall variability in southern Spain on decadal to centennial time scales. Int. J. Climatol. **20**, 721–732 (2000)
7. Luque-Espinar, J.A., Chica-Olmo, M., Pardo-Igúzquiza, E., García-Soldado, M.J.: Influence of climatological cycles on hydraulic heads across a Spanish aquifer. J. Hydrol. **354**(1–4), 33–52 (2008)
8. Lomb, N.R.: Least-squares frequency analysis of unequally spaced data. Astrophys. Space Sci. **39**, 447–462 (1976)
9. Scargle, J.D.: Studies in astronomical time series analysis. II. Statistical aspects of spectral analysis of unevenly spaced data. Astrophys. J. **263**, 835–853 (1982)
10. 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. Comput. Geosci. **49**, 207–216 (2012)
11. Confederación Hidrográfica del Guadalquivir (CHG). www.chguadalquivir.es
12. State Meteorological Agency (AEMET). http://www.aemet.es
13. Rodríguez-Tovar, F.J.: Orbital climate cycles in the fossil record: from semidiurnal to million-year biotic responses. Annu. Rev. Earth Planet. Sci. **42**, 69–102 (2014)
14. Climate Prediction Center. National Weather Center. http://www.cpc.ncep.noaa.gov/products/precip/CWlink/pna/nao.shtml
15. Climatic Research Unit, University of East Anglia. https://crudata.uea.ac.uk/cru/data/moi/

Localized Online Weather Predictions with Overnight Adaption

Michael Zauner, Michaela Killian and Martin Kozek

Abstract This paper extends the results presented in a conference paper showcasing an approach for online forecasting of ambient temperature and solar irradiation. The proposed method creates a localized prediction with an improvement over the available weather predictions ranging from 52 to 92% in ambient temperature forecast and 8–42% for solar irradiation forecast. This localized forecast can be used for improved predictions in smart homes or PV power plants for a more efficient operation. A new method for adapting the parameters of the autoregressive model with external input (ARX) for the solar irradiation over the night is proposed. This allows the model to be tuned to changing weather conditions without relying on external inputs.

Keywords Weather prediction · Local optimization · Parametric model

Introduction

With the ever-growing energy demand and the exhaustion of nonrenewable resources the efficient usage of renewable energy sources (wind, solar, tidal, and biomass) gets more important [4]. Due to the weather-dependent nature of those renewable energy sources, it is challenging to balance energy production and consumption in global electrical grids and in decentralized smart grids with smart consumers (e.g., smart homes). Therefore, it is vitally important to have accurate forecasting models for those renewable energy sources [6]. The most important factor influencing solar power production via PV (photovoltaic) systems is solar irradiation [9], followed by meteorological parameters like ambient temperature and relative humidity [4].

This work was supported by the project "intelliEE-Home" (FFG. No. 853663) in cooperation with evon GmbH.

M. Zauner (✉) · M. Killian · M. Kozek
Vienna University of Technology, Institute of Mechanics and Mechatronics,
Getreidemarkt 9, 1060 Vienna, Austria
e-mail: michael.zauner@tuwien.ac.at

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_18

Another application where solar irradiation and ambient temperature are important factors are the heating and cooling tasks of residential buildings [5]. Therefore, a model predictive controller for an HVAC (heating, ventilation, and air conditioning) system can not only provide better comfort for the residents but also save energy if accurate predictions are available.

The well-known modern numerical weather forecasting services (WFS) use discrete cells for simulating weather predictions. The initial conditions for those simulations are gathered by land-based weather stations and satellite images. This results in poor localized predictions as the forecast is valid for the whole cell. The idea of this paper is to create a localized weather prediction based on the forecasts of the WFS and the past and current local sensor data.

The WFS provides information up to several days ahead [9]. The temporal resolution of those WFS predictions is typically limited to 1 h. This resolution is usually not sufficient for the performance of most applications [3]. A common approach for short-term forecasting is based on sky imaging [2] and time series models [1, 4, 7, 9].

While predictions based on sky imaging provide good results in the range up to a few minutes, they also suffer from drawbacks: the devices are expensive, they require a lot of maintenance, and predictions are only usable when the cloud cover is not too high or too low.

The usage of localized sensors and global predictions provided by WFS allows for more accurate localized predictions and a higher temporal resolution. While many authors use nonlinear methods like Artificial Neuronal Networks (ANN) [7], Group Method of Data Handling (GMDH) [4], or Support Vector Machines (SVM) [9] for forecasting, a linear autoregressive model with exogenous input (ARX) is proposed in this paper. The advantages of using ARX models over ANN, GMDH, and SVM are that less parameters have to be optimized and the optimization can be done in real time. Bacher et al. [1] proposed a similar modeling approach in their work. The main differences to the proposed work are the usage of a diurnal component and a clear sky approximation via smoothing kernels. The method proposed in this paper includes an overnight prediction scheme to accommodate unmeasured changes in weather conditions.

With the usage of ARX models, the proposed method can learn statistical differences between local conditions (provided via the sensors) and the WFS predicted conditions.

This paper extends the contribution of Zauner et al. [10] by providing a more in-depth overview of the model order selection process, as well as providing a complete explanation on how the algorithm for generating localized forecasts works. Another addition is the comparison of the proposed models with a zero-order model, serving as reference.

The remainder of the paper is structured as follows: Section "Ambient Temperature Forecast" explains the methods and algorithms used for the ambient temperature forecast. Section "Solar Irradiation" highlights the necessary changes for the solar irradiation forecast. Section "Results" briefly explains the simulation setup as well as the results of the simulations for ambient temperature forecast and solar irradiation forecast. In the end, section "Conclusion" concludes this paper.

Ambient Temperature Forecast

The WFS is assumed to provide an ambient temperature prediction for the next 80 h. In the first, 65 h hourly prediction values are available. After that the WFS only provides predictions in 3 h intervals. An example of WFS prediction for the ambient temperature can be seen in Fig. 1.

The WFS prediction is linearly interpolated and a new time series $\vartheta_{\text{pred}}(k)$ with the uniform sampling time of $T_s = 0.25\,\text{h}$ is constructed, where $k = \{1, ..., T\}$ with T being the final time step where predictions are available.

It is assumed that a local weather station is measuring the local ambient temperature $\vartheta_{\text{amb}}(k)$ every 0.25 h. The last n measurements of the local temperature are saved in the system. At every time step, the vector

$$\boldsymbol{x}^T(k) = [\vartheta_{\text{amb}}(k-n+1), \ldots, \vartheta_{\text{amb}}(k), \vartheta_{\text{WFS}}(k), \ldots, \vartheta_{\text{WFS}}(k+m-1)] \quad (1)$$

is constructed, where $\vartheta_{\text{WFS}}(k)$ is the latest WFS prediction for the current time step and $\vartheta_{\text{amb}}(k)$ is the current measured ambient temperature. The variables $n \in \mathbb{N}^+$ and $m \in \mathbb{N}^+$ represent the order of the denominator and nominator in the resulting ARX model.

Using the weighted recursive least squares algorithm (WRLS) shown in (2a)–(2c),

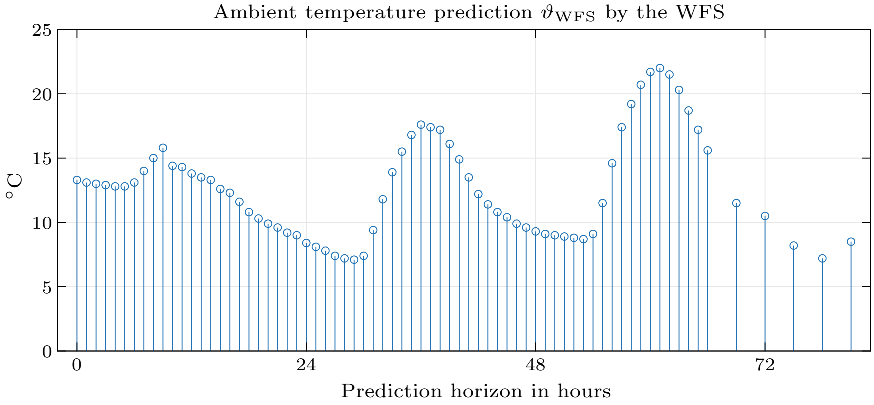


Fig. 1 Temperature prediction from the WFS. The hourly ambient temperature predictions are provided for the next 65 h and after that in 3 h intervals for an additional 15 h

$$\gamma(k) = \frac{\boldsymbol{P}(k)\boldsymbol{x}(k)}{\boldsymbol{x}^T(k)\boldsymbol{P}(k)\boldsymbol{x}(k) + \lambda}, \tag{2a}$$

$$\hat{\boldsymbol{\theta}}(k+1) = \hat{\boldsymbol{\theta}}(k) + \gamma(k)[\vartheta_{\text{amb}}(k+1) - \boldsymbol{x}^T(k)\hat{\boldsymbol{\theta}}(k)], \tag{2b}$$

$$\boldsymbol{P}(k+1) = \frac{1}{\lambda}[\boldsymbol{I} - \gamma(k)\boldsymbol{x}^T(k)]\boldsymbol{P}(k), \tag{2c}$$

with $\boldsymbol{P}(k) \in \mathbb{R}^{(n+m)\times(n+m)}$ being the parameter-covariance matrix, $\hat{\boldsymbol{\theta}}(k) \in \mathbb{R}^{(n+m)}$ representing the estimated parameter vector, and $\gamma(k) \in \mathbb{R}^{(n+m)}$ the correction vector. The scalar value $\lambda \leq 1$ represents the sensitivity of the algorithm to more recent values. Choosing a λ closer to 1 will increase the weight of past samples that are significant to the current parameter estimation. With $\lambda = 1$ the WRLS algorithm will behave like a regular recursive least squares algorithm. Furthermore, $\boldsymbol{I} \in \mathbb{R}^{(n+m)\times(n+m)}$ is defined as the unity matrix.

The initial value for the parameter-covariance matrix $\boldsymbol{P}$ is chosen as $\boldsymbol{P}(0) = \alpha\boldsymbol{I}$ where $\alpha \gg 1$. The initial value for $\hat{\boldsymbol{\theta}}(0)$ is chosen as a random $(n+m) \times 1$ vector. The future predictions for the ambient temperature denoted by $\hat{\vartheta}_{\text{amb}}(k|j+1)$ where $j+1$ represents any given future time step and k represents the current time step which are given by (3). Note that $j \geq k$ and $j < T$ must hold.

$$\hat{\vartheta}_{\text{amb}}(k|j+1) = \hat{\boldsymbol{x}}^T(k|j)\hat{\boldsymbol{\theta}}(k), \tag{3}$$

where

$$\hat{\boldsymbol{x}}^T(k|j) = [\tilde{\vartheta}_{\text{amb}}(k|j-n+1), \ldots, \tilde{\vartheta}_{\text{amb}}(k|j), \vartheta_{\text{WFS}}(j), \ldots, \vartheta_{\text{WFS}}(j+m-1)], \tag{4}$$

with $\tilde{\vartheta}_{\text{amb}}(k|i)$ being defined as

$$\tilde{\vartheta}_{\text{amb}}(k|i) = \begin{cases} \vartheta_{\text{amb}}(i) & \text{if } i \leq k \\ \hat{\vartheta}_{\text{amb}}(k|i) & \text{else.} \end{cases} \tag{5}$$

In (4), $\vartheta_{\text{WFS}}(j)$ is the most recent prediction for the time step j. Equation (5) recursively calculates predictions by (3) until only current or past measurements are needed for the formulation of $\hat{\boldsymbol{x}}^T$ in (4).

The predicted future values have to be recalculated after every new measurement since the parameter vector $\hat{\boldsymbol{\theta}}(k)$ is updated in (2b).

Algorithm 1 showcases the calculations needed at every time step k. The input $\vartheta_{\text{amb}} \in \mathbb{R}^{(n+1)\times 1}$ are the past and current measurements aligned in such a way that $\vartheta_{\text{amb}}(n)$ is the latest available measurement. The second input $\vartheta_{\text{WFS}} \in \mathbb{R}^{M\times 1}$ is the prediction, where $\vartheta_{\text{WFS}}(1)$ is the prediction for the $(k-1)$-th time step (compare to Eq. (1)). The lines 8–10 implement the WRLS algorithm while the lines 11–15 generate the future predictions over the prediction horizon N_p. The returned $\hat{\boldsymbol{\vartheta}}_{\text{amb}}$ are the predictions for future time steps.

```
Require: $\exists\, \vartheta_{\text{amb}} \in \mathbb{R}^{(n+1)\times 1}, \vartheta_{\text{WFS}} \in \mathbb{R}^{M\times 1}, \lambda \in \mathbb{R}^{1\times 1}, N_p \in \mathbb{R}^{1\times 1}$
Ensure: $M \geq N_p + m$
if $\nexists\, \hat{\boldsymbol{\theta}}, \boldsymbol{P}$ then
    $\hat{\boldsymbol{\theta}} = \text{rand}$
    $\boldsymbol{P} = \alpha * \boldsymbol{I}$
else
    {use previous $\hat{\boldsymbol{\theta}}, \boldsymbol{P}$}
end if
$\boldsymbol{x} = [\boldsymbol{\vartheta}_{\text{amb}}(0 : n-1), \boldsymbol{\vartheta}_{\text{WFS}}(1 : m)]^T$
$\gamma = \boldsymbol{P}\boldsymbol{x}/(\boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x} + \lambda)$ {Eq. (2a)}
$\hat{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} + \gamma[\vartheta_{\text{amb}}(n) - \boldsymbol{x}^T\hat{\boldsymbol{\theta}}]$ {Eq. (2b)}
$\boldsymbol{P} = [\boldsymbol{I} - \gamma\boldsymbol{x}^T]\boldsymbol{P}/\lambda$ {Eq. (2c)}
for $j = 1$ to $N_p$ do
    $\tilde{\boldsymbol{\vartheta}}_{\text{amb}} = [\boldsymbol{\vartheta}_{\text{amb}}(j : n), \hat{\boldsymbol{\vartheta}}_{\text{amb}}(\max(1, j-n) : j-1)]$
    $\boldsymbol{x} = [\tilde{\boldsymbol{\vartheta}}_{\text{amb}}, \boldsymbol{\vartheta}_{\text{WFS}}(j+1 : j+m)]^T$
    $\hat{\vartheta}_{\text{amb}}(j) = \boldsymbol{x}^T\hat{\boldsymbol{\theta}}$ {Eq. (3)}
end for
return $\hat{\boldsymbol{\vartheta}}_{\text{amb}}, \hat{\boldsymbol{\theta}}, \boldsymbol{P}$
```

Algorithm 1: WRLS prediction

Solar Irradiation

The WFS provides the hourly solar irradiation predictions for the next 43 h. Because of the diurnal and annual periodicity of the sun, the scheme presented in section "Ambient Temperature Forecast" cannot be used without modifications. The absence of measurements during the night does not allow for parameter adaption during the night. This is problematic since the weather conditions could change drastically overnight.

In a first step, the global horizontal irradiance (GHI), provided by the sensors, is transformed into the clear sky index. The clear sky index τ_{cs} is defined by

$$G = G_{\text{cs}} \cdot \tau_{\text{cs}}, \tag{6}$$

where G is the current global horizontal irradiation (in W/m^2) and G_{cs} is the clear sky global horizontal irradiation (in W/m^2). The clear sky index τ_{cs} is an indication for the transmissivity of the atmosphere. The GHI for clear sky conditions is calculated via the toolbox provided by Sandia National Laboratories [8].

As previously, the vector

$$\boldsymbol{x}^T(k) = [\tau_{\text{cs}}(k-n+1), \ldots, \tau_{\text{cs}}(k), \tau_{\text{WFS}}(k+1), \ldots, \tau_{\text{WFS}}(k+m)] \tag{7}$$

is created at every time step $k = \{1, ..., T\}$, where T is the final time step where predictions are available and $\tau_{\text{WFS}}(k+1)$ is the next clear sky index calculated with the WFS data. It is important to note that the current prediction $\tau_{\text{WFS}}(k)$ is not used, instead the next future prediction $\tau_{\mathrm{WFS}}(k+1)$ is included. This corresponds to a negative input dead time. The variables $n \in \mathbb{N}^+$ and $m \in \mathbb{N}^+$ represent again the order of the denominator and nominator in the resulting ARX model.

During the day, the WRLS algorithm shown in (2a)–(5) can be applied to compute the predictions for the clear sky index $\hat{\tau}_{\mathrm{cs}}$.

During the night, no calculations are possible due to the lack of measurements. In the morning, new initial values for $\boldsymbol{P}$ and $\hat{\boldsymbol{\theta}}$ are needed since the weather conditions could have changed significantly overnight.

To calculate the new initial values, the latest predictions from sunrise to sunset are queried before sunrise and concentrated into the vector $\boldsymbol{\tau}_{\mathrm{pred}}$. These predictions are then compared against measured solar irradiation time series of past days in a database.

```
Require: $\exists\ \tau_{\mathrm{pred},i} \in \mathbb{R}^{1\times 1}$, $\tau_{\mathrm{datab}} \in \mathbb{R}^{n_d \times 48}$
1: normalize $\tau_{\mathrm{pred}}$ to 12h day $\Rightarrow \tilde{\tau}_{\mathrm{pred}} \in \mathbb{R}^{1\times 48}$
2: for $j = 1$ to $n_d$ do
3:     $e_j = \sqrt{\frac{1}{48}\sum_i^{48}(\tilde{\tau}_{\mathrm{pred},i} - \tau_{\mathrm{datab},j,i})^2}$
4: end for
5: sort $e_j$ ascending
6: return $\tau_{\mathrm{datab},j}$ of $[e_1, e_2, ..., e_K]$
```

Algorithm 2: Find the K most similar solar days

Algorithm 2 showcases an example on how to search a database with n_d normalized entries. For a normalized solar day, the time between sunrise and sunset is defined as 12 h. Therefore, a single normalized solar day consists of 48 entries when sampled at 15 min intervals. Line 3 of the algorithm uses the Euclidean distance to calculate the similarity, but other distances could be considered too. Algorithm 2 returns the K most similar solar days in the database.

With the usage of the clear sky index and normalizing the solar days to 12 h, comparisons between the daily solar conditions can be drawn regardless of the time of the year.

Algorithm 3 describes the overnight prediction process. The prediction for the next day and the K most similar solar days from the database along with λ are the inputs. After normalizing the database entries and initializing $\boldsymbol{P}$ and $\hat{\boldsymbol{\theta}}$, the WRLS algorithm shown in (2a)–(5) is executed in line 5. In this WRLS algorithm, $\tau_{\mathrm{WFS}} = \tau_{\mathrm{pred}}$ and $\tau_{\mathrm{CS}} = \tilde{\tau}_{\mathrm{datab},j}$ according to (7). The overnight prediction algorithm then returns the new initial values for $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{P}$ for the WRLS algorithm that is active during the next day.

The choice of K should be large enough for the parameters to settle during the overnight prediction.

```
**Require:** $\exists\ \tau_{\text{pred},i} \in \mathbb{R}^{1\times 1}, \tau_{\text{datab}} \in \mathbb{R}^{K\times 48}, \lambda \in \mathbb{R}^{1\times 1}$
normalize $\tau_{\text{datab}}$ to size($\tau_{\text{pred}}$) $\Rightarrow \tilde{\tau}_{\text{datab}} \in \mathbb{R}^{K\times I}$
$\hat{\boldsymbol{\theta}}(0) = \text{rand}$
$\boldsymbol{P}(0) = \alpha * \boldsymbol{I}$
**for** $j = 1$ to $K$ **do**
    $[\hat{\boldsymbol{\theta}}(j), \boldsymbol{P}(j)] = \text{WRLS}(\tau_{\text{pred}}, \tilde{\tau}_{\text{datab},j}, \hat{\boldsymbol{\theta}}(j-1), \boldsymbol{P}(j-1), \lambda)$
**end for**
**return** $\hat{\boldsymbol{\theta}}(K), \boldsymbol{P}(K)$
```

Algorithm 3: Overnight prediction

Results

In this section, the simulation setup and the results of the proposed localized weather prediction algorithm described in sections "Ambient Temperature Forecast" and "Solar Irradiation" are presented. Both ambient temperature and solar irradiation forecasting simulations use data collected over a period of 36 days. The local weather station collects measurements for ambient temperature (in deg C) and solar irradiation (in W/m^2) with a common sampling time of 15 min. The database for the overnight prediction consists of 181 collected daily solar irradiation values from a different weather station from a different time frame. The dataset entries were already normalized to the normalized solar day with the correct sampling time of 15 min.

Ambient Temperature Forecast

The values for n and m in (1) represent the order of the ARX model. The denominator order is set by n and represents how many past measurements are used in the model. The nominator order is defined by m and corresponds to the amount of future predictions used.

Since the ARX model represents a virtual system rather than a physical system, there is no obvious correct choice for the model order. By enumerating over all reasonable combinations of n and m, and simulating each model output, two models have been selected for further analysis. The selection was not only based on the adjusted coefficient of determination (see Fig. 2) or the resulting RMSE (root mean square error) but also on the overall order of the model. Higher order models tend to give smaller RMSE values, while also having higher chances of overfitting the data. Overfitting occurs when the model does not only capture the underlying model characteristics but also the specific noise characteristics. This effect is unwanted since noise is inherently unpredictable and therefore the robustness of the model is decreasing.

Figure 2 showcases the adjusted coefficient of determination R^2_{adj} for all reasonable model orders. The first selected model ARX1 performs best at longer forecasting ranges of $N_p > 10$. The second selected model ARX2 is more reliant at shorter forecasting ranges of $N_p \leq 10$ but less accurate than ARX1 at longer ranges. The parameters of the two selected models used for the simulation can be found in Table 1.

In Fig. 3 the RMSE between various models and the WFS predictions are shown. The RMSE for the WFS is slowly increasing over the whole prediction horizon with 3 °C at $N_p = 0$. The RMSE of the ARX1 model is the lowest at prediction ranges from $10 < N_p < 288$ with 1.5 °C. This yields a forecast skill [9] of 52% over the available weather predictions. The second model ARX2 has the lowest error at the shortest ranges $N_p \leq 10 = 2.5$ h. For those shortest ranges, a forecast skill of up to 92% is achieved.

To showcase that the WFS prediction and the measured temperature are not simply offset from each other, a zero-order model was simulated too. This model, called ARX0 in Fig. 3, can be calculated by setting $\boldsymbol{x} = 1$ in (1). A model of order zero does not use any previous measurements or future predictions. It only uses a dynamically updated offset to align the current measurement to the current prediction. With the

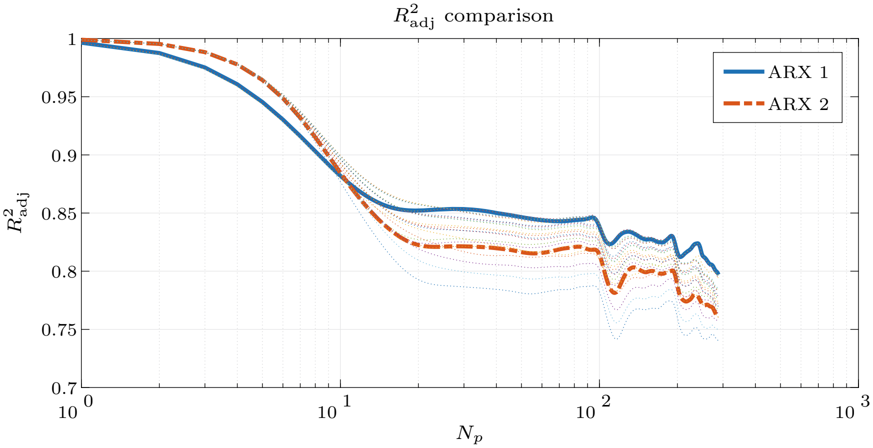


Fig. 2 Adjusted coefficient of determination R^2_{adj} over the prediction horizon N_p. Dotted lines represent omitted models. Note the logarithmic scale

Table 1 Parameters used for the ambient temperature forecast

Variable	ARX1	ARX2
N_p	288 samples = 3 days	288 samples = 3 days
n	1	2
m	2	2
λ	0.996	0.996

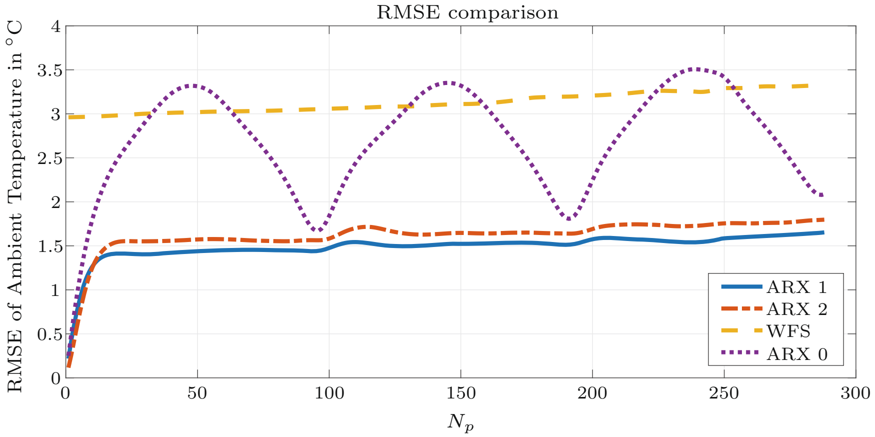


Fig. 3 RMSE between various ARX model outputs and the WFS for ambient temperature forecasting. ARX0 represents a model of order 0

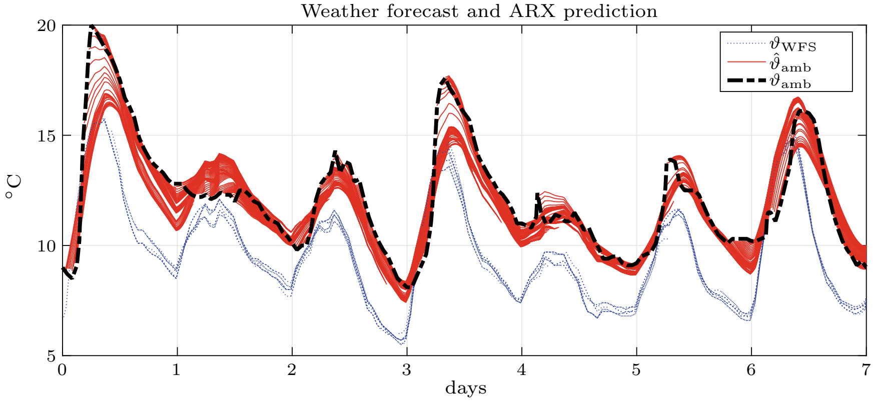


Fig. 4 Sample 7-day interval of WFS and ARX1 model output for ambient forecasting

ARX0 model output, a diurnal character can be seen. At $N_p = 96 = 1$ day, at $N_p = 192 = 2$ days, and at $N_p = 288 = 3$ days, the RMSE has a local minimum.

The ARX model output and the WFS forecast can be seen in Fig. 4. At every time step k when there is a new WFS prediction available, the new prediction ϑ_{WFS} is aligned correctly and plotted. At every time step k, the new predictions of the ARX1 model are plotted for the first 24 h. The predictions of the ARX2 model are indistinguishable by eye from the predictions of the ARX1 model at this scale, and therefore they are omitted for clarity. Note that the depicted interval requires a sufficient run-in period for the parameters to settle.

Solar Irradiation

As seen in section “Ambient Temperature Forecast”, the chosen values for n and m were evaluated by examining multiple models over a wide parameter range. The adjusted coefficient of determination of all the examined models can be seen in Fig. 5. The chosen model, called ARX3, is highlighted. This model offers the best trade-off between model order and performance. The chosen model parameters for the simulation can be found in Table 2 alongside with the parameter K which is used for the overnight prediction.

In Fig. 6, the ARX3 model output $\hat{\tau}_{\text{cs}}$ and the WFS predictions τ_{WFS} can be seen for 3 selected days. The black dash-dotted line represents the real (measured) clear sky index τ_{cs}. During the start/end of the sunny days, the value for the clear sky index τ_{cs} shows erratic changes. This is due to the small magnitudes of solar irradiation measured and also small magnitudes of clear sky solar irradiation G_{cs} which leads to

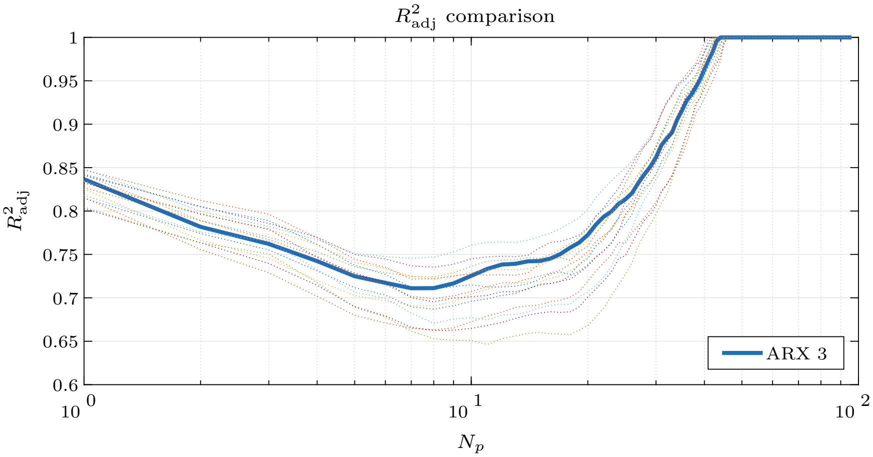


Fig. 5 Adjusted coefficient of determination R^2_{adj} over the prediction horizon N_p. Dotted lines represent omitted models. Note the logarithmic scale

Table 2 Parameters used for the solar forecast

Variable	ARX 3
N_p	Till sunset
n	1
m	3
λ	0.98
K	5

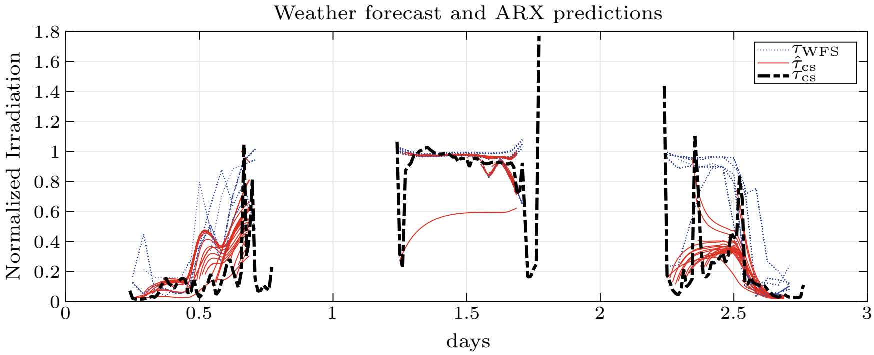


Fig. 6 Comparison of normalized WFS and ARX3 model output for solar forecasting for 3 selected days

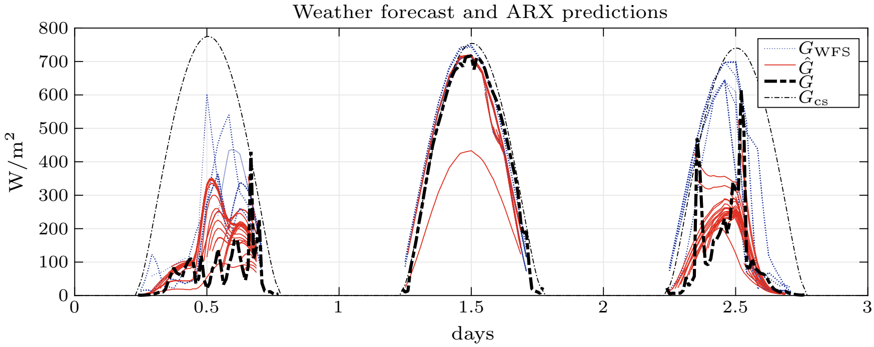


Fig. 7 Comparison of WFS and ARX3 model output for solar forecasting for 3 selected days

ill-conditioned normalization. During the day, when the clear sky index settles, the predictions are reliable. In Fig. 7, the outputs have been converted to GHI in W/m^2 and the clear sky global horizontal irradiation G_{cs} is plotted for reference.

Figure 8 gives a better overview on the accuracy of the ARX3 model compared to the predictions provided by the WFS. The development of the RMSE error over the prediction horizon is plotted for the ARX3 model output and the WFS predictions. Both predictions have an increasing RMSE over the prediction horizon with the ARX3 predictions displaying a significantly better RMSE for short-term predictions and at the end of N_p. The forecast skill over the WFS model is 8–42% for the solar irradiation forecast.

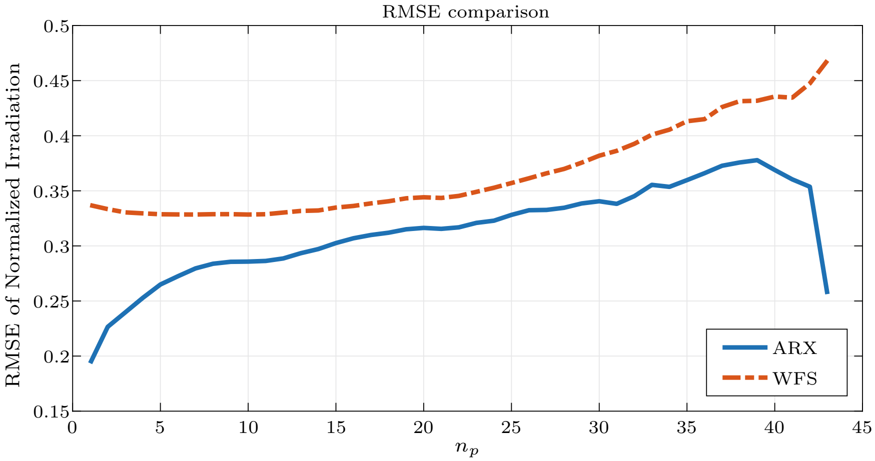


Fig. 8 RMSE between the ARX3 model output and the WFS for solar forecasting

Conclusion

Inspired by previous studies a forecasting method for ambient temperature and solar irradiation has been developed. The proposed approach localizes the numerical weather prediction provided by WFS to increase the local accuracy and reduce the forecasting errors. The overnight adaption of the ARX model parameters allows the model to accommodate to unmeasured changes and ensures a well-tuned start into the next day.

Simulation results with real data showcase the benefits of the proposed methods. The method is not computational intensive and can easily be run online on low-cost CPUs, for example, in home automation systems. Therefore, the localized forecasts could provide better predictions for smart home controllers and therefore increase the comfort, save money, and energy.

The results further showcase that the model order for forecasting the ambient temperature is dependent on the forecasting range. If the focus on the forecast is a short-term prediction, a higher order model can be considered. On the other hand, if the focus lies in forecasting longer ranges a simpler and more robust model should be considered. A possible solution to achieve a good prediction for various ranges would be to calculate multiple models in parallel and then blend their outputs together in a fuzzy manner.

References

1. Bacher, P., Madsen, H., Nielsen, H.A.: Online short-term solar power forecasting. Solar Energy **83**(10), 1772–1783 (2009)
2. Chow, C.W., Urquhart, B., Lave, M., Dominguez, A., Kleissl, J., Shields, J., Washom, B.: Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed. Solar Energy **85**(11), 2881–2893 (2011)
3. Fernndez-Peruchena, C.M., Gastn, M.: A simple and efficient procedure for increasing the temporal resolution of global horizontal solar irradiance series. Renew. Energy **86**, 375–383 (2016). https://doi.org/10.1016/j.renene.2015.08.004, http://www.sciencedirect.com/science/article/pii/S0960148115302044
4. Giorgi, M.D., Malvoni, M., Congedo, P.: Comparison of strategies for multi-step ahead photovoltaic power forecasting models based on hybrid group method of data handling networks and least square support vector machine. Energy **107**, 360–373 (2016). https://doi.org/10.1016/j.energy.2016.04.020, http://www.sciencedirect.com/science/article/pii/S0360544216304261
5. Killian, M., Zauner, M., Kozek, M.: Comprehensive smart home energy management system using mixed-integer quadratic-programming. Appl. Energy **222**, 662–672 (2018). https://doi.org/10.1016/j.apenergy.2018.03.179, http://www.sciencedirect.com/science/article/pii/S0306261918305282
6. Larson, D.P., Nonnenmacher, L., Coimbra, C.F.: Day-ahead forecasting of solar power output from photovoltaic plants in the American southwest. Renew. Energy **91**, 11–20 (2016). https://doi.org/10.1016/j.renene.2016.01.039, http://www.sciencedirect.com/science/article/pii/S0960148116300398
7. Marquez, R., Coimbra, C.F.: Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the NWS database. Solar Energy **85**(5), 746–756 (2011)
8. Stein, J.S.: The photovoltaic performance modeling collaborative (PVPMC). In: 2012 38th IEEE Photovoltaic Specialists Conference (2012). https://doi.org/10.1109/pvsc.2012.6318225, https://pvpmc.sandia.gov/
9. Wang, Z., Tian, C., Zhu, Q., Huang, M.: Hourly solar radiation forecasting using a volterra-least squares support vector machine model combined with signal decomposition. Energies **11**(1), 68 (2018)
10. Zauner, M., Killian, M., Kozek, M.: Localized online weather predictions with overnight adaption. In: The Proceedings of International Conference on Time Series and Forecasting, ITISE 2018, vol. 1, pp. 250–259 (2018)

Storm Characterization Using a BME Approach

Manuel Cobos, Andrea Lira-Loarca, George Christakos and Asunción Baquerizo

Abstract The storm that occurred at the Spanish coast of the Mediterranean Sea at the end of January of 2017 produced severe coastal floods. The space–time evolution of seawave heights during that event is analyzed in space and time using a combination of the spatiotemporal random field (S/TRF) theory and the Bayesian maximum entropy (BME) method. Observed were combined with hindcasted datasets from Puertos del Estado (Spain) to assess modeling accuracy and improve the analysis. The mean absolute error and root mean square error of the tenfold cross-validation technique were found to be equal to $7.90 \cdot 10^{-2}$ m and $9.59 \cdot 10^{-2}$m, respectively. The results are presented in the form of spatial maps of seawave height statistics (mean and variance) at the study domain. The mean wave height during the storm propagation is fairly well reproduced. The variance shows two regions of permanent maximum variance at the Mazarron Bay and between the Azahar coast and the north face of the Balearic islands. Some indicators were computed based on S/TRF of the mean wave height maps. The storm shape and a suitable storm determination threshold for the definition of the storm can be inferred from the results. The classification of several storms based on this methodology improves the assessment of the potential damage caused by the storm event, thus enabling the development of management strategies in coastal areas.

Keywords Storm characterization · Flood risk · Spatiotemporal random field · Bayesian maximum entropy

M. Cobos (✉) · A. Lira-Loarca · A. Baquerizo
Andalusian Institute for Earth System Research, University of Granada, Av. Del Mediterráneo S/n, 18006 Granada, Spain
e-mail: mcobosb@ugr.es

A. Lira-Loarca
e-mail: aliraloarca@ugr.es

A. Baquerizo
e-mail: abaqueri@ugr.es

G. Christakos
Geography Department, San Diego State University, San Diego, CA, USA
e-mail: gchrista@sdsu.edu

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_19

Introduction

It is widely recognized by the scientific community that climate change presents an unstable and uncertain picture of the environment [12]. It is expected that climate-driven effects will cause an increase of the mean sea level during the current century [8], and, also, an increase of the frequency and severity of extreme events like storms [13]. Therefore, climate change will have important consequences in the littoral [1, 14]. Coastal zones are areas especially vulnerable to climate change around the globe. This situation increases the vulnerability of large coastal regions and house properties, infrastructures, and industrial installations. Currently, more than 40% of the population live in the 100 km zone next to the coast, whereas 23% of the population live below the 100 m altitude. This is an issue in Spain where a high percentage of its more than 4000 km of coastline is inhabited and is very vulnerable to climate-driven hazards. Some authors predict the disappearance of approximately 40–50% of the beaches caused by an average sea level rise of 50 cm [15]. In fact, the entire Mediterranean Sea coast is a vulnerable region to climate change [10].

In view of the above considerations, this chapter presents a quantitative approach to characterize seawave height distributions during storms. The goal of this approach is twofold: (i) the characterization of a hundred different kinds of storms together with the atmospheric conditions that generate them and (ii) the detection and prevention against potential pattern of flooding.

Modern Spatiotemporal Geostatistics introduces a new perspective to space–time data modeling and prediction that involves the Bayesian maximum entropy (BME) theory [3, 4, 6]. The present work proposes a knowledge-based (KB) framework that combines spatiotemporal random field (S/TRF) modeling and BME theory. There are several applications of the BME theory in Earth Sciences, such as the mapping of particular matter pollution [20]; the prediction of granular content [2]; the uncertainty assessment of groundwater quality [17]; and the updating of digital soil organic matter maps based on historial records [19]. A recent review of the BME applications can be found in [11].

The KB framework is not commonly used in coastal and marine sciences although it seems to give very promising results for the design and/or construction of breakwaters, the management of estuaries, the location of energy extraction systems, or the development of management strategies against flood risk. The purpose of this chapter is to determine those stochastic space–time attributes of seawave heights during storms that are useful in the management of coastal areas or in flooding risk assessment. The proposed approach is applied in the characterization of a devastating storm that occurred in the Spanish Mediterranean Sea during the period January 20–24, 2017. This chapter is organized as follows. Section "Methodology" outlines the BME approach. Section "Study Area: Storms at the Mediterranean Sea" describes the study area, data sources, and previous tasks required to apply the BME approach in the specific site. Section "Results and Analysis" presents the results of the validation technique, the generated BME predictions, and the numerical indicators characterizing seawave heights. The discussion is presented at section "Discussion", followed by the study conclusions.

Methodology

The spatiotemporal random field (S/TRF) theory [3, 4] offers a powerful theoretical framework to study and analyze natural processes and phenomena that evolve in a spatial and/or space–time continuum [11]. The BME approach involves three main stages: the prior, meta-prior, and posterior stage (Fig. 1).

More specifically: at the prior stage, an adequate space/time geometry and rules of KB integration and processing are selected. The empirical relationships and/or theoretical laws governing the phenomena are expressed in terms of a general probability density function (f_G) that is maximally informative. At the meta-prior stage, the acquisition and storage of the various forms of specificatory (case-specific) data from various sources are considered, including hard (exact) and soft (uncertain) data. At posterior stage, the specificatory data are used to improve the prior pdf by means of Bayesian conditionalization. The result is a posterior pdf (f_K) that accounts for both the general KB (physical law, theoretical model, etc.) and the specificatory KB (hard and soft data).

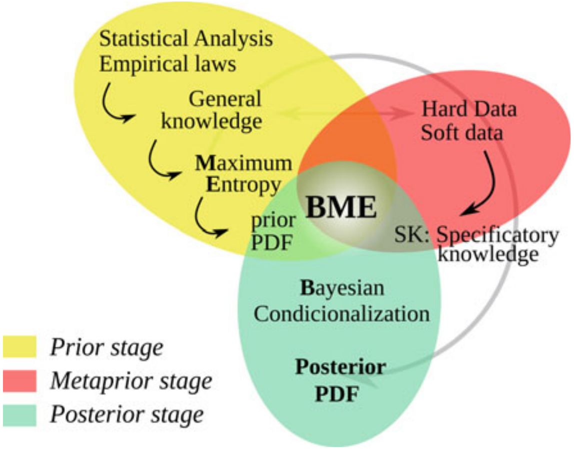


Fig. 1 Knowledge-based (KB) Bayesian maximum entropy (BME) method followed in this study

Space–Time Rules of General Knowledge

The rules of KB integration are established in terms of a set of moment equations $g_\alpha(\chi_m)$, $\alpha = 1, .., N$; g_α that represent known statistics of the S/TRF $X(p_m)$ at the space–time points p_m; N denotes the number of stochastic moments; and $\chi_m = \chi_s \cup \chi_k$ is the union of specificatory data (known) and the required predictions (unknown). A mean estimator based on the optimization of the Shannon information criterion was used [18]. This criterion reads

$$I_f(X_m) = -log f_G(\chi_m), \tag{1}$$

where χ_m represents a realization of the random variables X_m. In this chapter, $m = 1$ and the S/TRF X stands for the first spectral moment of the seawave height. Therefore, the general knowledge equations are expressed as follows:

$$\overline{h_\alpha(p_m)} = \overline{I_f(X_m)} = -\int log f_G(\chi_m) f_G(\chi_m)\, d\chi_m, \tag{2}$$

in which the form of the general KB-based pdf is assumed as given below:

$$f_G(\chi_m) = e^{\mu_0 + K}, \quad J = \sum_{\alpha=1}^{N_c} \mu_\alpha(p_m) g_\alpha(\chi_m), \tag{3}$$

with $\mu_0,\ \mu_\alpha$ is a vector of Lagrange multipliers whose coefficients are associated with $g_\alpha(\chi_m)$. Usually, $\overline{g_0} = 1$ and μ_0 account for the normalization constraint [7].

Specificatory Sources of Knowledge

At the meta-prior stage, site-specific data (exact or uncertain) are included in the analysis. The advantages were widely discussed in [5]. In situ or specificatory knowledge is provided by devices, information systems, numerical models, or experiments. In general, data cleaning, normalization, and some transformations should be performed previously in order to use it. At the end of this stage, accurate data (χ_h) and uncertain data (χ_s) coded by intervals or probability density functions with a measurement of the uncertainty should be provided.

Bayesian Conditionalization of Prior Pdf

The logical space–time rules are expressed in terms of the BME model. Hence, (f_G) (Eq. 3) is updated with the specificatory data as $P_K[x_k \leq \chi_k] = P_G[x_k \leq \chi_k \cup \chi_{data}]$, yielding

$$f_K(\chi_k) = f_G(\chi_k|\chi_{data}) = \frac{f_G(\chi_k \cup \chi_{data})}{f_G(\chi_{data})} = \frac{\int_R dF(\varepsilon, h) \int_{I(\varepsilon)} f_G(\chi_k, \chi_{data})\, d\chi_s}{\int_R dF(\varepsilon, h) \int_{I(\varepsilon)} f_G(\chi_{data})\, d\chi_s}, \tag{4}$$

where $K = G \cup S$, $I(\varepsilon)$ is the domain of χ_s and $\chi_{data} = \chi_h \cup \chi_s$.

If the general KB, G comprises the first two stochastic moments of the attribute (space–time mean and covariance or variogram functions) and the site-specific knowledge, S contains only hard data, the BME method is reduced to the Kriging method [3]. At this study, a kind of an enhanced Kriging method is considered since S-KB also contains soft data.

BME Estimation

Finally, the stochastic moments are computed at each space–time estimation point of interest p_k. As an example, the first and second centered moments (μ and σ^2) are obtained as follows:

$$\mu = \int \chi_k f_K(\chi_k)\, d\chi_k, \quad \sigma^2 = \int (\chi_k - \mu)^2 f_K(\chi_k)\, d\chi_k. \tag{5}$$

The results from Eq. (5) enable the mapping of the mean and variance of the S/TRF $X(p_k)$ at each space–time estimation point of interest p_k.

One-Point Site Indicators

A set of stochastic site indicators (SSI) was introduced in [5] based on the space–time characteristic function $I_x(p, \zeta) = 1$ if $X(p) > \zeta$ and 0, otherwise where ζ defines a threshold. On the basis of the previous definition, an SSI should be a function of the associated S/TRF X, the threshold ζ, and the space–time point p.

Table 1 shows the one-point indicators used in this study. The parameter I_1 describes the relative area of excess height that physically described the relative active area of the storm. The parameter I_2 provides the average wave height exceedance over the study area. This indicator assesses the potential impact of the storm arriving to the coast. The mean excess differential wave height is defined by I_3. It provides the maximum wave height difference with respect to the threshold. Finally, I_4 is the conditional mean excess wave height which defines the mean wave height of the storm subarea with seawave heights over the threshold.

Table 1 One-point indicators definition and description. $E[]$ is the expectation of variables between brackets

One-point indicators	Equation	Description
$I_1(\zeta)$	$E[I_x(p,\zeta)]$	Active area of storm given a threshold
$I_2(\zeta)$	$E[X(p)I_x(p,\zeta)]$	Wave height average that exceed ζ
$I_3(\zeta)$	$E[(X(p)-\zeta)I_x(p,\zeta)]$	The mean excess differential wave height
$I_4(\zeta)$	$E[X(p)\vert X(p)\geq\zeta]$	Average wave height exceeding threshold ζ over I_1

Study Area: Storms at the Mediterranean Sea

The Mediterranean Sea (MS) is a semi-closed water body connected to the Atlantic Ocean through the Strait of Gibraltar at its west limit. The main basin is divided into two principal sub-basins. Its waters bathe the coasts of more than 20 countries. The MS region lies in a transition zone between the arid climate of North Africa and the temperate and rainy climate of central Europe and it is affected by interactions between mid-latitude and tropical processes [10]. The North West MS is delimited westward by the Straight of Gibraltar (35° 58′ 18″ N–5° 29′ 09″ W) and eastward by the Tyrrhenian Sea where it has as boundaries the islands of Corse and Sardegna at longitude 10° E (Fig. 2a). Hence, northwest basin covers a water area of 650 thousand of km^2 with a mean depth of about 1500 m and a perimeter of approximately 4000 km.

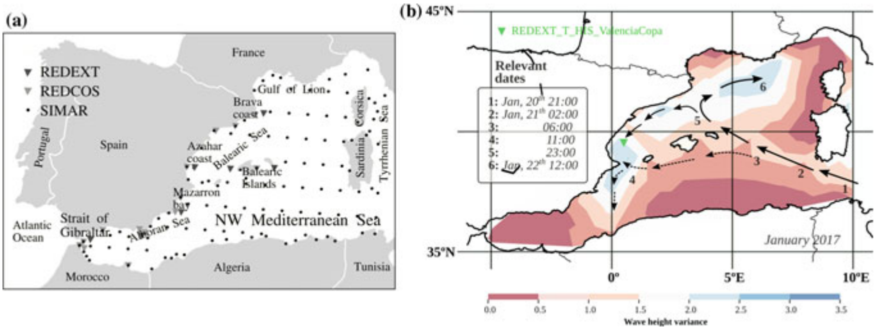


Fig. 2 Study area. **a** Western basin of the Mediterranean sea (MS) and some interest locations; **b** Wave height variance in the time period computed with available data during the storm of January 2017 and some relevant dates

Observed and Simulated Datasets

The data used in this study were obtained from Puertos del Estado, Spain (Fig. 2a). The present methodology uses data compound by observations from REDEXT and REDCOS and hindcasting analysis from SIMAR. Observations were recorded by directional SeaWatch and Triaxys devices at a sampling frequency equals to 4 Hz during sample periods of 24 and 26 min every hour. This data are processed and hourly statistical samples of spectral data are obtained. The significant spectral wave height is the S/TRF $X(p)$ of our study. The seawave properties from SIMAR were computed from reanalysis with WaveWatchIII oceanographic model feeded with the wind field obtained by HIRLAM with the atmospheric data from NCEP (National Centers for Environmental Predictions). These data are source of uncertainty and will be treated as soft data.

Figure 2b shows the temporal wave height variation in the study area during the storm. As it can be seen, the storm is fed by winds that blow from the east of the MS. Seawaves travel from southeast to northwest, growing during the storm propagation until arriving to the Balearic islands. Then, the storm is split into two branches, one of them propagates to the northwest arriving to Brava coast and the Gulf of Lion along the north face of Balearic Islands while the other branch propagates toward the southwest.

Implementation at the Northwest Basin of the Mediterranean Sea

The methodology is applied to the storm that occurred between of January 20 and 24, 2017. The severity of this storm led to several floods in the coastal area of Valencia. The following section presents the BME methodology applied to a S/TRF that models the significant spectral heights of seawaves.

Spatiotemporal Dependences No physical laws were used in this chapter because seawaves propagation over uneven bottoms cannot be reduced to analytical expressions in terms of the space–time moments. So, f_G is defined in terms of the covariance of χ_h. Seawave heights vary slightly in space and time; therefore, the physical process is assumed to be homostationary [4], and the covariance is a function of the distance h and the time-lag τ (isotropy). These hypotheses will be discussed later. Then a coupled space–time covariance model is selected given by

$$c(h,\tau) = ae^{\left(\frac{h}{b}-\frac{\tau}{c}\right)} + d, \tag{6}$$

where a (1.891), b (6.727), c (46.196), and d (0.431) are parameters that were fitted using the sequential least squares programming optimization algorithm (SLSQP) in Python; h and τ are spatial distances and temporal lags. The fitted surface is depicted in Fig. 3. The fitted covariance models adequately represent the covariance of data

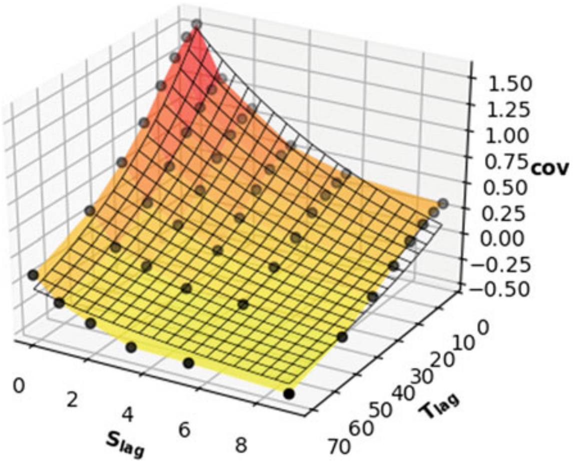


Fig. 3 Spatiotemporal covariance of seawave heights during the storm. The mesh represents the fitting Eq. (6) evaluated at pair points given by space distance and time lag

yielding mean absolute error and root mean square error values less than 0.018 and 0.021, respectively. The significance of these space–time relationships is observed in parameters b and c. The points located closest to 6.727 km and in a time lag shorter than 46.2 h present a positive covariance and will be more representative of the final value at the required location.

Site-Specific Data The specificatory knowledge is obtained from datasets. REDEXT and REDCOS are used as hard data. A total of 1440 space–time data points were distributed in 10 locations. Soft data χ_s was related with hard data χ_h with a linear regression of pair points where hindcasting and observed data coexisted. Figure 4a shows the parameters of the linear regression. It is noted that simulated data slightly overpredict the observed values of seawave heights. CI_{95} are also indicated with a dashed line. The soft dataset is represented by 14790 points distributed in 102 spatial locations with the mean and variance computed by linear regression. The error follows a Gaussian distribution, and hence χ_s can be adequately described by the mean and variance.

The whole data presented are specificatory data that we use in the BME method. Figure 4b shows the smoothing cdf of hard and soft data and the Gaussian distribution. As we can observe after the detrending, the seawave heights follow a normal distribution. The mean (μ) and median (f_{50}) are next to zero, the standard deviation is one, the skewness (γ_1) is slightly negative, and the modified kurtosis (g_2) is almost zero. The maximum error along the normalize z-axis is lower than 7%. The Gaussian behavior of data is adequate to apply the BME theory.

The BME Prediction Steps The computation of the stochastic space–time moments follows the steps that are briefly summarized below:

1. Obtain f_G, analytically or numerically (Eq. 6).
2. Select an adequate number of local hard and soft neighboring data in space–time. Usually, a minimum of 50 space–time points is required.

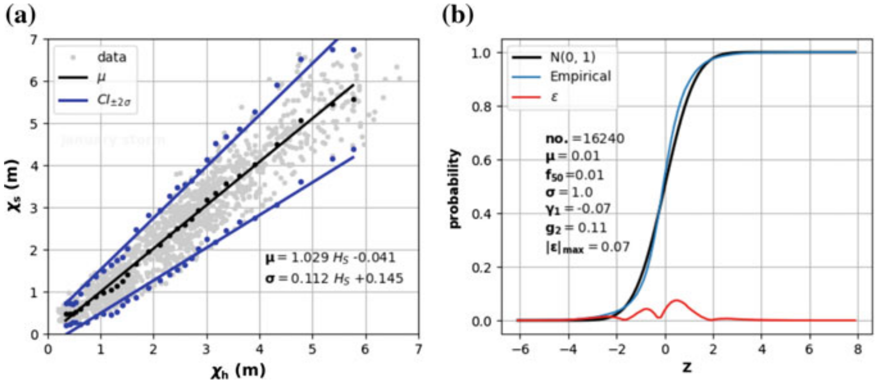


Fig. 4 Previous tasks in the specificatory data. **a** Linear regression of the sample mean (μ) and standard deviation (σ) between hard and soft data at collocated space–time locations. **b** Normalization of space–time seawave heights and some statistical

3. Compute the spatial distances and time lags between data $p_d = (p_h, p_s)$ and the estimation points of interest p_k, as well as the corresponding covariance values.
4. Calculate the local mean BME through regression (Eq. 4).
5. Subtract the mean from hard and soft data values.
6. Calculate the statistical moments (Eq. 5).

Results and Analysis

Validation with Tenfold Cross-Validation Technique

The performance of the BME method was assessed by means of a tenfold cross-validation technique. We selected randomly a tenth of the space–time hard data and removed it from the original specificatory data. Then, we used as space–time points the removed space–time points and run the BME model. We repeated this process ten times, and the results were compared with the raw data. Table 2 shows the results of two common accuracy indices widely applied in literature, the mean absolute error (MAE) and the root mean square error (RMSE). The computed values of both indices ranged between 1 and 9 hundredths which guarantee the accuracy of the methodology.

$$MAE = \frac{1}{N_c}\left(\sum_{i=1}^{N_c} |\mu - \chi_{map}|\right) \quad RMSE = \frac{1}{N_c}\sqrt{\sum_{i=1}^{N_c} (\mu - \chi_{map})^2} \tag{7}$$

Table 2 Mean absolute error and root mean square error of tenfold cross-validation application to χ_h

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10
MAE (x10^{-2})	2.96	7.90	3.08	4.15	2.79	3.23	4.22	3.12	3.64	1.31
RMSE (x10^{-2})	5.58	5.35	7.47	7.06	5.22	6.14	9.59	6.27	8.37	1.13

BME Predictions

The first two centered moments were computed following Eq. (5). The results are presented in 144 maps with the temporal evolution of the mean seawave heights (Fig. 5) and the variance of the seawave heights (Fig. 6).

As it is observed in Fig. 5, the storm is generated at the Strait of Sicilia and travels over the north basin of the Mediterranean Sea. Southeastern winds favor the growth

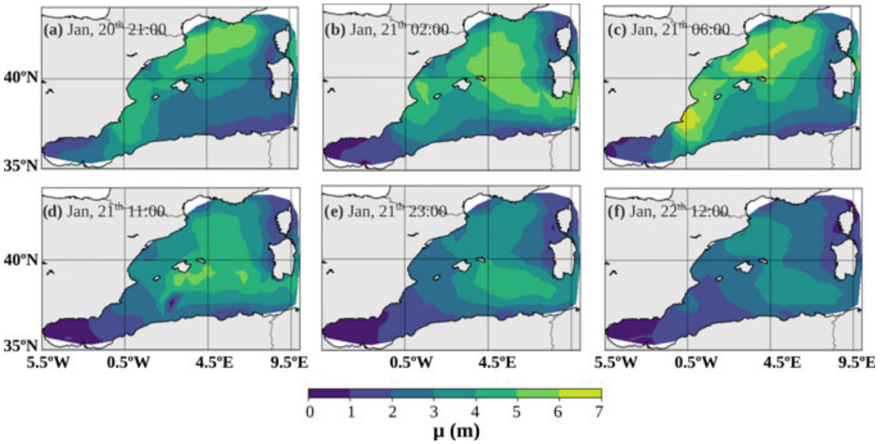


Fig. 5 Estimation of the mean wave height at the study area for several relevant dates

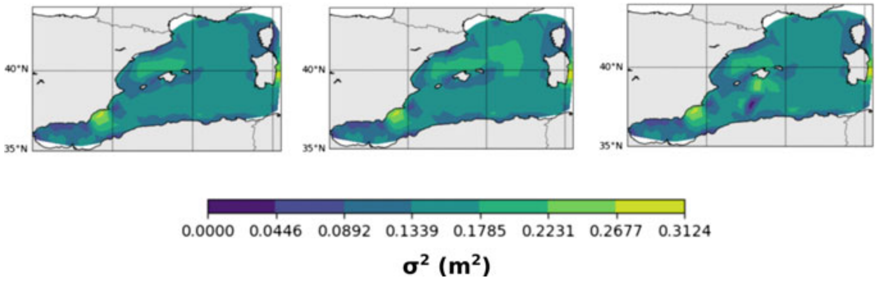


Fig. 6 Estimation of the wave height variance at the study area for several relevant dates

and propagation of waves from the south of Sardinia (January 20, 21.00). The water available distance where winds blow is maximum and persistent winds drive higher seawaves. The wave front reaches the Balearic islands, and then is divided into two fronts that propagate over the lower and upper parts of the basin (January 21, 06.00). During this stage, maximum wave heights (∼7 m) are observed. The bottom front quickly reaches the Gulf of Valencia and is dispersed downward (January 21, 11.00). The upper front propagates toward the Gulf of Lion, and then it decays too. The same pattern is observed at the video included as supplementary material where SIMAR hindcast data were used.

Conversely, the space–time variation observed in μ is not found in σ^2 (Fig. 6). An almost regular pattern of covariance is depicted. Two areas present persistent maxima, the Mazarron bay ($\sigma^2 = 0.3\,\mathrm{m}^2$) and Azahar coast ($\sigma^2 \sim 0.25\,\mathrm{m}^2$) next to the upper face of the Balearic islands. The uncertainty measured through the variance is 4–5 times greater than the rest of the study area. A reason of this deviation is that those regions have larger depths that facilitate the wave propagation.

ST Characterization of a Storm

The indicators defined in Table 1 give some hints about the shape of the storm: triangular, trapezoidal, semicircular, etc. I_2 provides a global assessment of expected overcome wave for the fraction of site where $X(p)$ exceeds the threshold ζ. Finally, I_4 gives an idea of the mean wave height in the subarea that defines the threshold.

As it can be observed in Fig. 7, the active area of the storm decreases sharply, getting the inflection point at approximate 3 m when the relative area is just the 20% of the storm area. The parameters I_2, I_3, and I_4 begin at value of 2.23 m. I_2 and I_3 decrease to zero while I_4 increases up to the maximum wave height. I_3 depicts

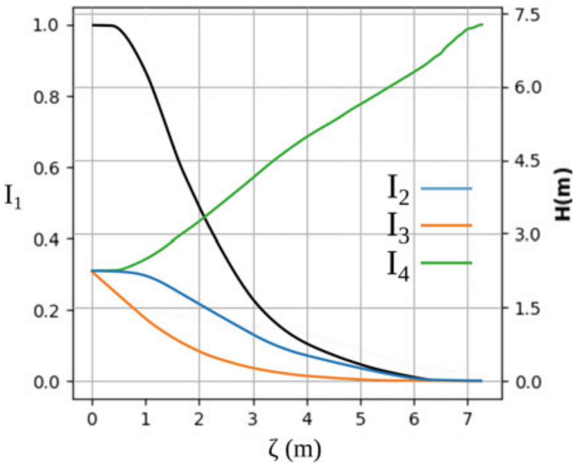


Fig. 7 Results of one-point indicators of seawave heights for the storm. The values of I_1 are given on the left vertical axis. The values of the other indicators are given in the right vertical axis. H stands for wave height

the mean excess over the threshold. As it is observed, the waves that overtake the threshold decrease quickly with increasing threshold. With these information, a piked bell-like storm can be inferred. Uprising threshold planes split the bell in concentric circles whose area reduces exponentially.

Discussion

The methodology proposed to characterize storms in terms of their first two centered moments has shown to be useful in giving clues about the potential damage and harm of storm that approaches to the coast. The prior information is a critical component for inference tasks, unlike the data-driven classical statistics approach that is based purely on sampling information [9]. Hard data were obtained from devices which error is well known and confined. However, the location of the buoys may differ hundreds of meters, and also their positions strongly affect its height due to wave transformation during its propagation over different depths, especially in shallow areas. Devices were located at the continental shelf where water depths overtake 250 m. Under this condition, the seawave height is independent of water depth.

The phenomenon was assumed isotropic and represented by the covariance function (Eq. 6). This hypothesis was founded after the study of the anisotropy. The directional variogram was computed [16]. The spatial and temporal range for the sample correlation was established according to the mean physical characteristics of the storms. So, we set five intervals and the spatial maximum range was set accordingly to the maximum storm shape (∼9°). The time step was set to 6 h. The dissimilitude of directional variogram (results not shown) follows a transition both in time and space that follows an exponential relationship. At short periods, the phenomenon is anisotropic but fastly goes into isotropic. This process can also be seen in the video. The storm propagates in the ES-WN direction. This direction has the maximum spatial gradient and shows the maximum dissimilitude. However, in the normal axis, the minimum dissimilitude is found due to the storm traveling as a wave train.

The sensibility of these results to the chosen time period needs to be further investigated. The threshold value can be subjectively set requiring a balance between sample selection bias and variance. Physically, the independence of consecutive storms should be guaranteed. This problem is usually found when dealing with storm evolution. Furthermore, other multi-point indicators can be employed and give insights of spatial distributions, specifically location or motion of the storm crest or the connectivity, among others. The sensibility of this method to the number of χ_s and χ_h should also be further investigated.

The methodology is feasible to: (i) detect regions where the seawave propagation is more sensible to atmospheric and marine conditions, (ii) better estimate seawave at any given point, or (iii) detect anomalous functionality of recording devices.

Conclusion

The stochastical descriptors of seawave height (mean and variance) have been computed in the basis of a methodology that uses the spatiotemporal random field theory driven by the Bayesian maximum entropy. A spatiotemporal characterization of seawave height during the storm that occurred between January 20 and 24, 2017 at the Spanish Mediterranean sea is presented. Seawave height data were employed to establish the physical bases and site-specific relations and to apply the Bayesian maximum entropy conditionalization.

The methodology estimates adequately the mean values, depicting fairly well the storm evolution. The accuracy of the methodology is ensured with tenfold cross-validation which means absolute error and root mean square error are lower than 0.1 m. Two regions are identified where variance is large ($\sim 0.25\,\mathrm{m}^2$), the Mazarron bay and the Azahar coast. Furthermore, the shape of the storm is bell-like. The vinculation of indicators results with flooding effects which allows the assessment of the potential risk of flooding in coastal areas.

Acknowledgements This research was partially funded by the Campus de Excelencia Internacional del Mar (Cei-MAR) and the program of Precompetitive Research projects for young researchers of the UGR plan (PPJI_B-06). It was also supported by AQUACLEW. Project AQUACLEW is part of ERA4CS, an ERA-NET initiated by JPI Climate, and funded by FORMAS (SE), DLR (DE), BMWFW (AT), IFD (DK), MINECO (ES), ANR (FR) with co-funding by the European Commission. M.C. acknowledges the mobility support received for the research stay at the San Diego University State and grateful to the Department of Geography (SDSU) for kindly hosting him during the analysis of the research.

References

1. Baquerizo, A., Losada, M.A.: Human interaction with large scale coastal morphological evolution. An assessment of the uncertainty. Coast. Eng. **55**(7), 569–580 (2008)
2. Bogaert, Patrick, D'Or, Dimitri: Estimating soil properties from thematic soil maps. Soil Sci. Soc. Am. J. **66**(5), 1492–1500 (2002)
3. Christakos, G.: Modern Spatiotemporal Geostatistics, vol. 6. Oxford University Press (2000)
4. Christakos, G.: Spatiotemporal Random Fields: Theory and Applications. Elsevier (2017)
5. Christakos, G., Hristopulos, D.T.: Characterization of atmospheric pollution by means of stochastic indicator parameters. Atmos. Environ. **30**(22), 3811–3823 (1996)
6. Christakos, G., Olea, R.A.: Sampling design for spatially distributed hydrogeologic and environmental processes. Adv. Water Resour. **15**(4), 219–237 (1992)
7. Christakos, G., Serre, M.L., Kovitz, J.L..: BME representation of particulate matter distributions in the state of California on the basis of uncertain measurements. J. Geophys. Res. Atmos. **106**(D9), 9717–9731 (2001)
8. Church, J.A., White, N.J., Hunter, J.R.: Sea-level rise at tropical pacific and Indian ocean islands. Global Planet. Change **53**(3), 155–168 (2006)
9. Ellison, A.M.: Bayesian inference in ecology. Ecol. Lett. **7**(6), 509–520 (2004)
10. Giorgi, Filippo, Lionello, Piero: Climate change projections for the Mediterranean region. Global Planet. Change **63**(2–3), 90–104 (2008)
11. He, J., Kolovos, A.: Bayesian maximum entropy approach and its applications: a review. Stoch. Environ. Res. Risk Assess. 1–19 (2017)

12. IPCC Impacts. Adaptation, and vulnerability. part a: global and sectoral aspects. Contribution of working group ii to the fifth assessment report of the intergovernmental panel on climate change. Clim. Change (2014)
13. Lejeusne, C., Chevaldonné, P., Pergent-Martini, C., Boudouresque, C.F., Pérez, T.: Climate change effects on a miniature ocean: the highly diverse, highly impacted Mediterranean sea. Trends Ecol. Evol. **25**(4), 250–260 (2010)
14. Losada, M.A., Baquerizo, A., Ortega-Sánchez, M., Ávila, A.: Coastal evolution, sea level, and assessment of intrinsic uncertainty. J. Coast. Res. 218–228 (2011)
15. Moreno, J.M. de la Rosa, D., Zazo, C.: Evaluación preliminar de los impactos en España por efecto del cambio climático. España. Ministerio de Medio Ambiente (2005)
16. Olea, R.A.: A six-step practical approach to semivariogram modeling. Stoch. Environ. Res. Risk Assess. **20**(5), 307–318 (2006)
17. Serre, M.L., Christakos, G., Li, H., Miller, C.T.: A bme solution of the inverse problem for saturated groundwater flow. Stoch. Environ. Res. Risk Assess. **17**(6), 354–369 (2003)
18. Shannon, C.E.: Coding theorems for a discrete source with a fidelity criterion. IRE Nat. Conv. Rec **4**(142–163), 1 (1959)
19. Sun, X.-L., Wu, Y.-J., Lou, Y.-L., Wang, H.-L., Zhang, C., Zhao, Y.-G., Zhang, G.-L.: Updating digital soil maps with new data: a case study of soil organic matter in Jiangsu, China. Eur. J. Soil Sci. **66**(6), 1012–1022 (2015)
20. Yang, Y., Christakos, G.: Spatiotemporal characterization of ambient $\mathrm{PM}_2.5$ concentrations in Shandong Province (China). Environ. Sci. Technol. **49**(22), 13431–13438 (2015)

Energy Time Series Forecasting

Adaptive Methods for Energy Forecasting of Production and Demand of Solar-Assisted Heating Systems

Viktor Unterberger, Thomas Nigitz, Mauro Luzzu, Daniel Muschick and Markus Gölles

Abstract Solar-assisted heating systems use the energy of the sun to supply consumers with renewable heat and can be found all over the world where heating of buildings is necessary. For these systems, both heat production and heat demand are directly related to the weather conditions. In order to optimally plan production, storage, and consumption, forecasts for both the future heat production of the thermal solar collectors as well as the future heat demand of the connected consumers are essential. For this reason, this contribution presents adaptive forecast methods for the solar heat production and the heat demand of consumers using weather forecasts. The developed methods are easy to implement and therefore practically applicable. The final verification of the methods shows good agreement between the predicted values and measurement data from a representative solar-assisted heating system.

Keywords Energy forecast · Production forecast · Demand forecast · Solar heat production · Heat demand

Introduction

Solar-assisted heating systems use the energy provided by the sun to supply consumers with renewable heat and can be found all over the world where heating of buildings is necessary. Especially large-scale solar-assisted heating systems, which benefit from the effect of scale regarding the cost of solar heat, are an important element of future energy systems [1–4].

These solar-assisted heating systems are typically set up with a solar collector field to generate heat, and a buffer storage to store the heat and decouple the occurrence of heat production from heat consumption, at least to a certain degree. Furthermore, there are typically one or more auxiliary heating systems (e.g., a gas burner) in order

V. Unterberger · T. Nigitz · M. Luzzu · D. Muschick · M. Gölles (✉)
Automation and Control, Bioenergy 2020+ GmbH, Graz, Austria
e-mail: markus.goelles@bioenergy2020.eu

V. Unterberger · T. Nigitz · M. Gölles
Institute for Automation and Control, Graz University of Technology, Graz, Austria

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_20

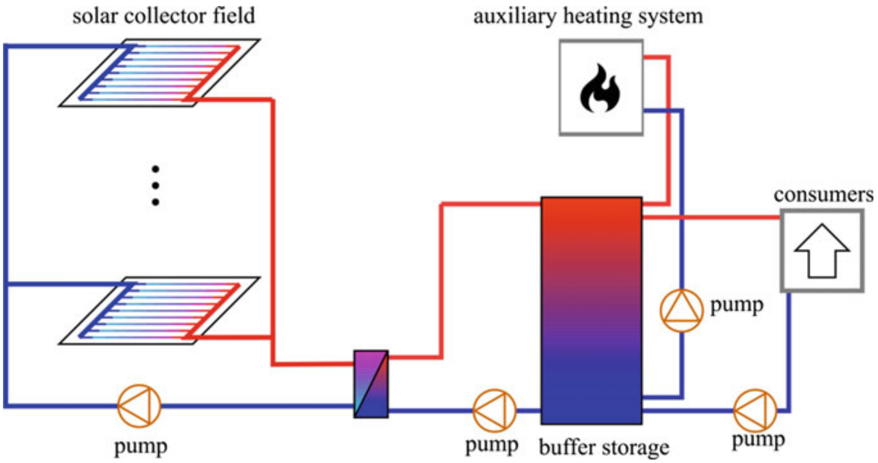


Fig. 1 Schematic representation of a typical solar-assisted heating system

to supply the connected consumers with heat in case there is a lack of solar energy and there is no heat stored in the buffer storage. A schematic representation of such a system is shown in Fig. 1.

However, controlling such a solar-assisted heating system can be particularly challenging because the daily weather fluctuations have significant effects on the system's performance, e.g., [5, 6]. Modern high-level controllers, typically referred to as buffer management or energy management systems, thus rely on forecasts of the future heat production and demand to derive efficient control strategies, e.g., [7, 8]. These strategies can lead to an overall efficiency improvement in the range of 2–5%, e.g., [9–11].

Unfortunately, most of the available forecasting methods for solar heat production as well as for heat demand are tailored for a specific application, not adaptive and often mathematically complicated, e.g., [12–17]. In particular, because of their complex mathematical structure (e.g., artificial neural networks), it is often not possible to easily implement them on controllers typically used in solar-assisted heating systems.

For this reason, this paper presents energy forecasting methods for both the solar heat production and the heat demand of consumers which are adaptive, easy to implement, and thus suitable for the use in controllers of solar-assisted heating systems. Both methods shown in this contribution use weather forecasts from a weather service provider (*meteoblue*, https://www.meteoblue.com) as inputs and their outputs are verified with measurement data from a representative solar-assisted heating system.

In the following, the general principle on which the forecast methods rely on will be described in section "General Principle". Section "Solar Heat Production Forecast" then presents a specific implementation for forecasting the solar heat production, which is suitable for a wide range of solar collectors, and section "Heat Demand Forecast" describes an implementation for forecasting the heat demand of the consumers. Verifications of the presented methods based on comparisons with measurement data from a representative solar-assisted heating system will be presented in both cases. Finally, section "Conclusion" draws conclusions from the presented methods and their practical verification.

General Principle

The forecasting method proposed relies on the natural periodicity of the phenomena underlying both the solar heat production and the heat demand of the consumers. In the case of solar heat production, it is obvious to assume that the sun follows approximately the same path in the sky as the day before and thus generates approximately the same amount of heat at the same time of day as long as the weather does not change significantly. Likewise, the heat demand will only change slightly from one day to the next, assuming similar ambient conditions and user behavior. As a first approximation, it is thus reasonable to predict the solar heat production as well as the heat demand at a particular time of day by using the corresponding values at the same time of the previous day and possibly multiple days in the past. However, looking too far into the past must not be advised since seasonal changes or changes in the users' behavior would not be taken into account sufficiently.

This rough prediction can be enhanced by additionally considering information on external factors influencing the variables to be forecasted, such as variations in the global irradiation due to clouds or variations in the ambient temperature. Forecasts for the global irradiation and the ambient temperature are available from weather service providers, with a typical sampling time of 1 hour. An easy way to express a dependency from external factors is through a multiple linear regression model:

$$\tilde{y}_{h,d} = \beta_{0,h} + \sum_{j=1}^{n} \beta_{j,h} \hat{x}_{j,h,d}. \tag{1}$$

Here, $\tilde{y}_{h,d}$ denotes the prediction of the variable to be forecasted for hour h of day d, $\beta_{0,h}$ is a constant offset, $\hat{x}_{j,h,d}$ is the prediction of the external factor j for hour h of day d, and the model parameters $\beta_{j,h}$ are weighting the influence of the n external factors. The dependency from the external factors was found to be different for the individual hours of the day in some cases; thus, different model parameters are used for each individual hour of the day.

In order to determine the model parameters, the model error $e_{h,d}$ is defined as

$$e_{h,d} := \tilde{y}_{h,d} - y_{h,d}, \tag{2}$$

where $y_{h,d}$ represents the actual value of the variable to be forecasted at hour h of day d. By using historical values for the variable to be forecasted, as well as historical values for the external factors to be considered, the optimal model parameters can be calculated by minimizing a given cost function. A reasonable approach for the cost function is the sum of the squared errors of the model for the corresponding hours of all days in the past which should be used for the model parametrization. It is then possible to determine the optimal model parameters $\beta^*_{j,h}$ by solving an ordinary least squares problem, i.e.,

$$\beta^*_{j,h} = \underset{\beta_{j,h}}{\operatorname{argmin}} \sum_{d=d_0-1}^{d_0-N_d} e^2_{h,d}, \; j = 0, \ldots, n. \tag{3}$$

Here, d_0 denotes the day when the model parameters are determined and N_d denotes the number of previous days to be used for parametrization. As an example, consider the linear regression model (1) with two external factors ($n = 2$) and using 4 training days ($N_d = 4$). For hour $h = 8$ (08:00–09:00 am) of day $d =$ Tuesday, the three model parameters $\beta_{0,8}$, $\beta_{1,8}$, and $\beta_{2,8}$ can be calculated by solving the overdetermined system of equations containing the measured, historical value of the variable to be forecasted $y_{8,\mathrm{Fr-Mo}}$ and the measured historical external factors $x_{1-2,8,\mathrm{Fr-Mo}}$ of the training days. A graphical representation of this example is shown in Fig. 2.

Apart from only considering the time of the day, it could be necessary to also distinguish between different types of days, e.g., working days and holidays. Then, separate model parameters for each time of the day are needed to be identified for each set of days. A further consideration of the day of the year is advised only if the forecasting method should provide forecasts with a horizon of several weeks or months, i.e., long-term forecasts. As the forecasting method proposed should support the control of solar-assisted heating systems, a forecast horizon of several days up to a week is required. For these forecasts, the periodicity in the variable to be forecasted can be deduced more reliably from the previous days than from the previous year. Hence, the day of the year is considered only implicitly, by automatically adapting the model parameters when learning from only a few days in the past.

So far, the forecast for one hour only considers measurement data from the last days, but disregards measurements that are more recent. One hour is incorrectly

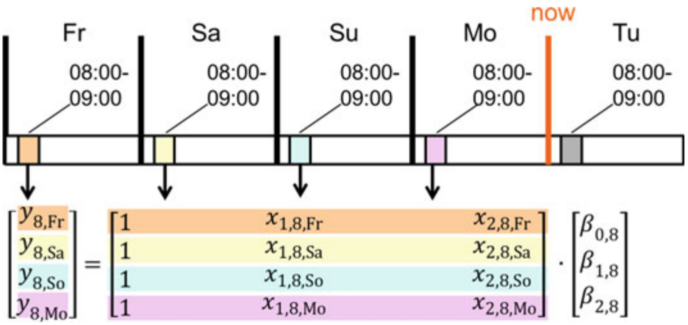


Fig. 2 Graphical representation of the parametrization routine for a simple example

assumed independent from the next. This would mean that, e.g., the prediction for 3 o'clock p.m. of the following day could be calculated at 4 o'clock p.m. of the current day and would not change after that.

In order to reduce this shortcoming, a correction step is introduced that takes the current prediction error into account. This correction step could act in two ways, depending on the variable to be forecasted. If the variable to be forecasted equals the solar heat production, then a prediction error typically occurs from temporary local shading by clouds or buildings close to the solar collectors. Such a prediction error is likely to persist for a certain time, and should be considered by correcting the short-term predictions with the current prediction error. If the variable to be forecasted equals the heat demand, then a prediction error typically occurs from nonperiodic consumer behavior. Changes in timing (e.g., showering earlier than usual) lead to a shift in demand rather than an overall increase or decrease. Then a prediction error does not persist, but rather it is likely to change its sign, which should be considered with the negative current prediction error. However, analysis of measurement data showed that a positive prediction error is more likely to persist for a certain time. Thus, the correction step adds the weighted current prediction error to the following few hours, which can be written as follows:

$$\hat{y}_{h,d} = \tilde{y}_{h,d} + (y_0 - \tilde{y}_0)\Phi(\Delta h), \tag{4}$$

where $\hat{y}_{h,d}$ is the corrected forecast for hour h of day d, y_0 is the most current measurement, $\tilde{y}_0$ is the corresponding forecast, and Φ is a monotonically decreasing function of the time difference Δh between the most current measurement and the hour given by indices h and d.

This simple and general forecasting method needs to consider different external factors and must take into account different periodicity assumptions depending on whether it is used to forecast the solar heat production or the heat demand. The following chapters will go into the detailed implementation for solar heat production forecast (Section "Solar Heat Production Forecast") and heat demand forecast (Section "Heat Demand Forecast").

Solar Heat Production Forecast

For solar heat production, short wave radiation is absorbed by an absorber and the energy is transferred to a heat carrier, typically water mixed with glycol to avoid freezing [18]. The occurring energy flows for a solar collector are the heat input from the sun $\dot{Q}_{\mathrm{in}}$ (where parts of the solar radiation I_{g} are reflected by the transparent cover), the ambient heat losses through heat conduction and convection $\dot{Q}_{\mathrm{l,cc}}$, the heat losses through radiation $\dot{Q}_{\mathrm{l,r}}$, and the heat finally transferred to the heat carrier $\dot{Q}_{\mathrm{out}}$. This leads to the following energy balance equation:

$$\dot{Q}_{\mathrm{out}} = \dot{Q}_{\mathrm{in}} - \dot{Q}_{\mathrm{l,cc}} - \dot{Q}_{\mathrm{l,r}}. \tag{5}$$

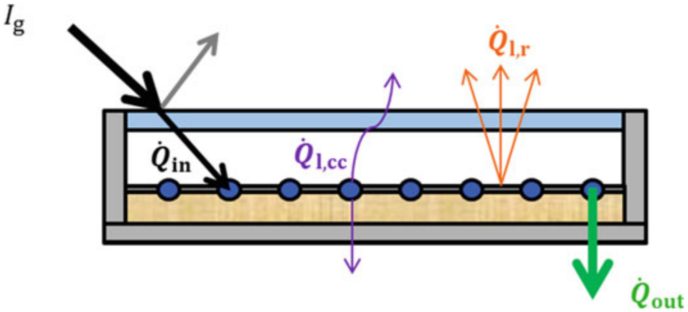


Fig. 3 Schematic structure of a flat plate collector and the occurring energy flows

These different energy flows are shown exemplarily in Fig. 3 for the case of a solar flat plate collector. This type of collector is typically employed in solar-assisted heating systems to efficiently provide hot water below temperatures of 100 °C.

The heat produced by a solar collector $\dot{Q}_{\text{out}}$ during steady-state operating conditions and for near-normal incidence angle of the solar radiation can be approximately expressed by the static energy equation according to the European Standard EN12975:2006:

$$\dot{Q}_{\text{out}} = A_{\text{coll}}\,\eta_0\,I_g - A_{\text{coll}}c_1\left(\overline{T}_{\text{fl}} - T_{\text{amb}}\right) - A_{\text{coll}}c_2\left(\overline{T}_{\text{fl}} - T_{\text{amb}}\right)^2, \tag{6}$$

where A_{coll} represents the net collector area, I_g symbolizes the global solar irradiation hitting the collector surface, $\overline{T}_{\text{fl}}$ is the arithmetic mean fluid temperature between the inlet and the outlet of the collector, and T_{amb} is the ambient temperature. The coefficients represent the optical efficiency η_0, the heat loss coefficients for heat conductance, c_1, and for thermal radiation, c_2. The EN12975:2006 standard covers performance, durability, and reliability testing of almost all solar collector types available on the market [19].

Even if the Standard EN12975:2006 is accepted and widely used, the analysis of measurement data from large-scale solar-assisted heating systems shows that applying this model, with its parameters (A_{coll}, η_0, c_1, and c_2) taken from the datasheet of the collectors, does not directly lead to satisfying results for forecasting the solar heat production, even if the external factors I_g, $\overline{T}_{\text{fl}}$, and T_{amb} are known. This is because the model parameters are only valid for stationary conditions in the laboratory and not for the conditions occurring during daily operation. Furthermore, these model parameters would have to change over time to account for polluted collector surfaces decreasing the optical efficiency (given by η_0), the decay of materials leading to higher heat losses of the collectors (given by c_1 and c_2) and local shading reducing the effective collector area (given by A_{coll}).

These shortcomings can be eliminated when applying the forecast method described in section "General Principle". In order to apply the forecast method, some assumptions must be made with regards to Eq. (6). First, by taking into account that solar collectors are typically operated with a constant desired outlet temperature, which is ensured by a temperature controller, the outlet temperature of the solar collector can be assumed to be constant. Second, because the solar collectors of solar-assisted heating systems are typically connected to the lowest part of a buffer storage (Fig. 1), the inlet temperature can be assumed to be constant as long as the buffer is reasonably sized and not inefficiently operated. Taking these two simplifications into account, a constant inlet and outlet temperature can be assumed, which leads to a constant mean fluid temperature $\overline{T}_{\mathrm{fl}} = \mathrm{const}$. Introducing the variable temperature difference ΔT between the constant mean fluid temperature and the variable ambient temperature, reduces Eq. (6) to

$$\dot{Q}_{\mathrm{out}} = A_{\mathrm{coll}}\ \eta_0\ I_g - A_{\mathrm{coll}} c_1 \Delta T - A_{\mathrm{coll}} c_2 \Delta T^2. \tag{7}$$

Note that this representation of the widely used static collector model is identical in structure to the linear regression model with external factors $\hat{x}_1 = I_{\mathrm{g}}$, $\hat{x}_2 = \Delta T$, and $\hat{x}_3 = \Delta T^2$. The physical parameters correspond to the model parameters, i.e., $\beta_0 = 0$, $\beta_1 = A_{\mathrm{coll}}\eta_0$, $\beta_2 = -A_{\mathrm{coll}}c_1$, and $\beta_3 = -A_{\mathrm{coll}}c_2$.

The forecast model is thus equivalent to the static collector model and is therefore valid for a wide range of collectors. Furthermore, with the forecast method the collector parameters are continuously adapted using measurement data of the produced solar heat, the temperature difference between mean collector temperature and ambient temperature as well as the global solar irradiation. The dependency on the time of day is considered by using different parameter sets for each hour of the day to forecast the solar heat production $\tilde{y}_{h,d}$:

$$\tilde{y}_{h,d} = \beta_{1,h}\hat{x}_{1,h,d} - \beta_{2,h}\hat{x}_{2,h,d} - \beta_{3,h}\hat{x}_{3,h,d}. \tag{8}$$

By this procedure, the influence of pollution of the collector fields and decay of the materials is automatically considered.

What is more, the effects of local shading, which would have to be incorporated into the static collector model by complicated 3D modeling and shadow calculations (see, e.g., [18]), are automatically considered and no manual parameterization is necessary. In addition, later enhancements such as added solar collectors or changing environmental conditions such as additional buildings throwing shadows are automatically incorporated into the forecasting process.

The forecasting method for the solar heat production is evaluated for days of the year where heat production is significant but challenging to predict. Solar heat production during winter is typically very small or negligible, which makes forecasting not very interesting and its benefit small. During the summer months, solar heat production is typically at its highest, but weather conditions are quite steady, which makes forecasts interesting but less challenging. In spring and autumn, solar heat production can be high too while the weather conditions can be very unsteady. This makes forecasts interesting and more challenging.

Therefore, the forecasting method for solar heat production is evaluated using a day in spring, with an additional day in summer for comparison. It is applied to a

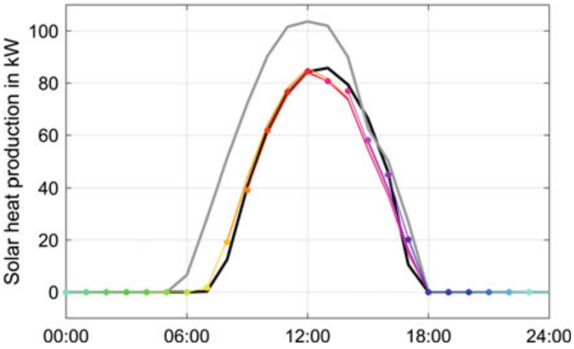


Fig. 4 Forecast of solar heat production during a day in summer

collector field built in 2009 for a solar-assisted heating system in Austria, consisting of flat plate collectors with a net area of 138 m^2.

The evaluation of the reference day in summer is visualized in Fig. 4, showing the measured heat produced by the solar collector (black), the consecutive hourly forecasts (in color, starting with green, going via yellow to red and blue), and the output of the static collector model with the parameters taken from the datasheet (gray).

From 0 o'clock a.m. to 12 o'clock, the forecast of the proposed method is very accurate and predicts the solar heat production very precisely. After that, there is a slight variation which gets corrected over the next hours via the current prediction error, thus slightly improving the next forecast. The energy produced for this day is 563 kWh, and the initial forecast with the method developed in this paper (without any hourly correction applied) is 556 kWh (−1.5%). In comparison, the forecast of the static collector model, with the parameters taken from the datasheet, yields 785 kWh (+40%). Considering that the solar irradiation forecast is almost perfect for this day, this is a large error. This is because using the static collector model will immediately predict heat output as soon as global irradiation levels increase. In reality, however, the system needs time to heat up until the temperature is above the necessary limit and only then will it provide hot water. Furthermore, other effects such as aging of the material or dust on the collector surface can lead to lower yields.

The presented method handles this naturally by using different coefficients for every hour and thus predicting the start and stop times of the solar heat production very accurately, which is an important information for the controller of a solar-assisted heating system.

The evaluation of the day in spring is visualized in Fig. 5. For this spring day, the heat produced is about half that of the summer day. From midnight to 11 o'clock a.m., the forecast is very accurate. At 11 o'clock a.m., the forecast estimates the heat production too high, which is why the next prediction at 12 o'clock, which was already a bit too low, is corrected downward. The opposite is true for 1 o'clock p.m., when actual production drops in an unexpected way, but the forecast is corrected upward because of the underestimation at 12 o'clock. Days where the solar heat

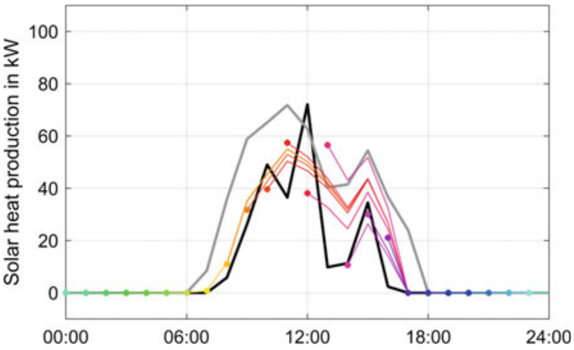


Fig. 5 Forecast of solar heat production during a day in spring

production drops and rises repeatedly, which can happen through rapid cloud movements, are therefore very challenging for the forecast, especially if these changes are not apparent from the global irradiation forecasts. However, after 2 o'clock p.m., the forecast is sufficiently accurate again and the start and stop times of the solar heat production are predicted sufficiently well too. In contrast, the static collector model (gray) predicts the heat output as soon as global irradiation is available, which leads to wrong predictions of the start and stop times of solar heat production. The energy produced this day was 248 kWh, the initial forecast with the model proposed in this paper (without a correction applied) is 343 kWh (+40%). However, the forecast of the static collector model with the parameters taken from the datasheet is worse with 500 kWh (+100%).

Heat Demand Forecast

The heat demand of consumers depends on many external factors such as the users' behavior patterns and weather factors such as ambient temperature, solar irradiation, rain, and wind. However, most of these factors are either hard to predict, follow a periodic pattern themselves or only have a relatively small influence on the heat demand. For example, the wind direction and speed might play a role because of higher ambient losses, but they cannot be predicted sufficiently well for an individual building. Solar irradiation might play a major role especially with modern glass palaces, but its effect is mostly determined by the angle of incidence and thus the time of day. After investigating the influence of the individual factors on multiple test objects (family homes, office buildings, etc.), it was found that considering the ambient temperature T_{amb} is enough for obtaining sufficient heat demand forecasts (see [20]). This reduces the linear regression model (1) to

$$\tilde{y}_{h,d} = \beta_{0,h} + \beta_{1,h}\hat{x}_{1,h,d}, \tag{9}$$

with $\hat{x}_{1,h,d}$ representing the ambient temperature forecast $\widehat{T}_{\mathrm{amb},h,d}$ for hour h of day d. In addition to considering the time of day, a distinction between working days and weekend days improves the forecasting quality significantly. Hence, separate hourly model parameters are considered for working days and weekend days. Seasonal variations and changing users' behavior are considered by continuously updating the model parameters using measured heat demand and ambient temperature of the last 2 weeks.

The evaluation of the forecasting method for heat demand should again be performed using challenging days. The heat demand during winter is mainly dominated by room heating, which typically makes forecasts easy. The heat demand during summer is mainly dominated by hot water usage and thus varying consumer behavior, which typically makes forecasts hard. In spring and autumn, a combination of both is present. Therefore, the forecasting method for heat demand is again evaluated using a day in spring and a day in summer like it was used for the evaluation of the solar heat production forecast in section "Solar Heat Production Forecast". The consumer used for the evaluation is an office building with a connected load of 300 kW supplied by the solar-assisted heating system also used for evaluating the solar heat production forecast in section "Solar Heat Production Forecast". The evaluation of the day in spring is visualized in Fig. 6, showing the measured heat demand of the consumer (in black) and the consecutive hourly forecasts (in color). The reason why the heat demand at 00:00 does not match the one at 24:00 is because the ambient temperature has increased from day to day (like it is very common in spring). This shows the importance of not only relying on periodicity alone but also taking external factors like ambient temperature into account.

From midnight to 6 o'clock a.m., the forecast underestimates the demand by a certain amount, but for the rest of the day the forecast is quite accurate. The consecutive forecasts are corrected via the current prediction error, which improves the next forecast.

The evaluation of the day in summer is visualized in Fig. 7.

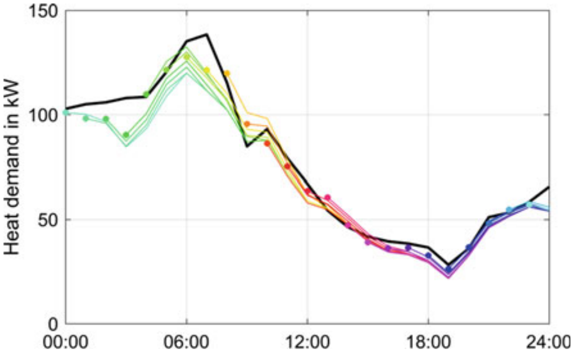


Fig. 6 Forecast of heat demand during a day in spring

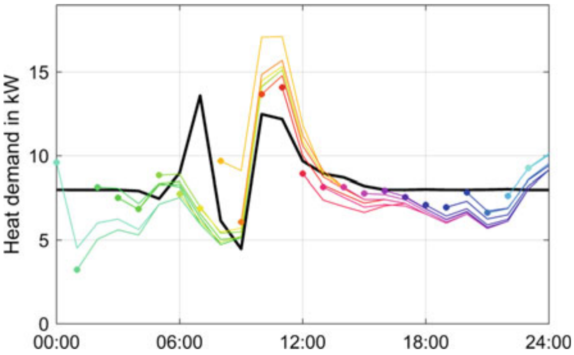


Fig. 7 Forecast of heat demand of a day in summer

Here the demand is around a tenth of the demand in spring. The forecast differs stronger from the measurements, as in summer individual actions by the users dominate the heat demand and neither periodicity nor the predicted external factors play a major role. However, while the relative prediction errors are quite large, the absolute prediction errors are quite small.

Conclusion

The forecasting method proposed in this article is sufficiently simple to be implemented even on very simple computers (programmable logic controllers, PLCs) but still automatically learns from past data and thus contains no parameters that need to be tuned manually. It is based on a linear regression model using forecasts of external factors that accounts for changes relative to a simple periodic signal extrapolation. A simple correction step takes into account recent prediction errors to further improve the forecasts for the next few hours. The forecasting method was shown to work reasonably well for forecasting solar heat production and heat demand. With few modifications, it can also be used to forecast other quantities such as electrical power demand and photovoltaic electrical energy production.

The only requirement of the forecasting method is the availability of measurement data for both the quantity being forecasted and the external factors influencing that quantity, as well as forecasts of the external factors themselves. These forecasts can often be obtained from weather service providers, e.g., using web interfaces. It is important to note that the forecast quality of both production and demand naturally strongly depends on the forecast quality of the external factors such as ambient temperature and global irradiation. Therefore, it is worth considering commercial weather service providers, which typically offer better forecast quality.

By continuously updating the model parameters using past data, the method automatically adapts to changing behavior patterns or environmental conditions and needs no kind of manual parametrization. It is, therefore, a plug-and-play solution that requires no experts for implementation and parametrization. It can be used in software ranging from simple buffer management controllers to complicated, optimization-based energy management systems. At present, the method is already successfully used in two demonstration plants to predictively control a large-scale solar-assisted heating system as well as a district heating system.

References

1. Fisch, M.N., Guigas, M., Dalenbäck, J.O.: A review of large-scale solar heating systems in Europe. Sol. Energy **63**(6), 355–366 (1998)
2. Website. https://www.energy-innovation-austria.at/article/big-solar-2/?lang=en. Accessed 3rd Dec 2018
3. Bauer, D., Marx, R., Nußbicker-Lux, J., Ochs, F., Heidemann, W., Müller-Steinhagen, H.: German central solar heating plants with seasonal heat storage. Sol. Energy **84**(4), 612–623 (2010). https://doi.org/10.1016/j.solener.2009.05.013
4. Lund, H., Werner, S., Wiltshire, R., Svendsen, S., Thorsen, J.E., Hvelplund, F., Vad Mathiesen, B.: 4th Generation District Heating (4GDH): integrating smart thermal grids into future sustainable energy systems. Energy **68**, 1–11 (2014). https://doi.org/10.1016/j.energy.2014.02.089
5. Camacho, E.F., Berenguel, M.: Control of solar energy systems. IFAC Proc. Vol. **45**(15), 848–855 (2012). https://doi.org/10.3182/20120710-4-SG-2026.00181
6. Lemos, J.M., Neves-Silva, R., Igreja, J.M.: Adaptive control of solar energy collector system. In: Advances in Industrial Control. Springer International Publishing (2014)
7. Vesterlund, M., Toffolo, A., Dahl, J.: Optimization of multi-source complex district heating network, a case study. Energy **126**, 53–63 (2017). https://doi.org/10.1016/j.energy.2017.03.018
8. Lichtenegger, K., Wöss, D., Halmdienst, C., Höftberger, E., Schmidl, C., Pröll, T.: Intelligent heat networks: first results of an energy information-cost-model. Sustain. Energy, Grids Netw. **11**(Suppl C), 1–12 (2017). https://doi.org/10.1016/j.segan.2017.05.001
9. Quintana, H., Kummert, M.: Potential of model predictive control (MPC) strategies for the operation of solar communities. In: Proceedings of BS 2013: 13th Conference of the International Building Performance Simulation Association, pp. 2481–2488 (2013)
10. Quintana, H., Kummert, M.: Optimized control strategies for solar district heating systems. J. Build. Perform. Simul. **8**, 1–18 (2014)
11. Unterberger, V., Lichtenegger, K., Innerhofer, P., Gerardts, B., Gölles, M.: Evaluation of the potential for efficiency increase by the application of model-based control strategies in large-scale solar thermal plants. Int. J. Contemp. Energy **4**(1), 549–559 (2018)
12. Fang, T., Lahdelma, R.: Evaluation of a multiple linear regression model and Sarima model in forecasting heat demand for district heating system. Appl. Energy **179**, 544–552 (2016). https://doi.org/10.1016/j.apenergy.2016.06.133
13. Kramer, W., Bitterling, M.: Artificial Neural Networks (ANN) for the prediction of local outside temperatures and solar yields. In: Proceedings SWC 2017 (2017). https://doi.org/10.18086/swc.2017.22.0
14. Mestekemper, T., Kauermann, G., Smith, M.S.: A comparison of periodic autoregressive and dynamic factor models in intraday energy demand forecasting. Int. J. Forecast. **29**(1), 1–12 (2013)
15. Neto, A.H., Fiorelli, F.A.S.: Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. Energy Build. **40**(12), 2169–2176 (2008)
16. Nielsen, H.A., Madsen, H.: Modelling the heat consumption in district heating systems using a grey-box approach. Energy Build. **38**(1), 63–71 (2006). https://doi.org/10.1016/j.enbuild.2005.05.002

17. Zhao, H., Magoulès, F.: A review on the prediction of building energy consumption. Renew. Sustain. Energy Rev. **16**(6), 3586–3592 (2012). https://doi.org/10.1016/j.rser.2012.02.049
18. Duffie, J.A., Beckman, W.A.: Solar Engineering of Thermal Processes. Wiley (2013). Print-ISBN: 9780470873663, Online-ISBN: 9781118671603, https://doi.org/10.1002/9781118671603
19. Kovacs, P.: A guide to the standard EN 12975, SP—Technical Research Institute of Sweden, QAiST—IEE/08/593/SI2.529236, Deliverable D2.3 (2012)
20. Nigitz, T., Gölles, M.: A generally applicable, simple and adaptive forecasting method for the short-term heat load of consumers. Appl. Energy. (submitted Dec 2018)

Short-Term Forecast of Wind Turbine Production with Machine Learning Methods: Direct and Indirect Approach

Mamadou Dione and Eric Matzner-Løber

Abstract The Energy Transition Act defined by the French State has precise implications on Renewable Energies, in particular on its remuneration mechanism. Until then, a purchase obligation contract permitted the sale of wind-generated electricity at a fixed rate. From now, it will be necessary to sell this electricity on the Market (at variable rates) before obtaining additional compensation intended to reduce the risk. This sale on the market requires to announce in advance (about 48 h) the production that will be delivered on the market, thus it is very important to predict this production. The objective of the project is to provide, every day, short-term forecasts (48 h horizon) of wind power production. We use two approaches: a direct one that predicts wind generation directly from weather data, and an indirect one that predicts wind from weather data and converts it into production. In order to forecast the production we use different machine learning algorithms and we propose features engineering to improve the forecasts. Our results are very conclusive compared to those in literature.

Keywords Short-term forecasting · Machine learning · Spatiotemporal dynamics modeling · Wind power prediction

Introduction

At the end of 2015, the estimated share of renewable energies in the world's electricity generation was 24.5% compared to 75.5% of fossil fuels and nuclear energy [1]. Wind power occupied 4% of the world's electricity production. In France, renewable electricity covers 18.8% of electricity consumption in a slippery year with a share of 4.1% for the wind turbine [2]. With the objectives of the United Nations Confer-

M. Dione (✉)
ENGIE Green France, Montpellier, France
e-mail: mamadou.dione@engie.com

E. Matzner-Løber (✉)
CREST-ENSAE ParisTech, Paris Saclay, France
e-mail: Eric.Matzner-Lober@ensae.fr

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*, Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_21

ence on Climate Change (COP21), renewable energies will reach 32% in France's electricity supply by 2030 [3]. In other words, wind power is expected to double by 2030. This shows that the renewable energy sector is booming and particularly wind energy and need a new management to support the energy transition.

In France, this energy transition is starting to have implications in the electricity market. Until then, a purchase obligation contract made possible to sell wind-generated electricity at a fixed rate. From now on, with State-defined Energy Transition Law, the fixed rate ended and the electricity should be sell on the market (before obtaining additional compensation). Producers should announce the quantity of electricity which will be delivered on the market and under or over estimation could exposed them to penalties. Thus there is a huge need for precise forecasts. Wind, which is the fundamental resource for wind farm production is highly variable, and producers need to develop accurate forecasting methods providing reliable forecasts (with as little variance as possible from the actual production that will be delivered). Forecasting wind farm production is of course not new [4–8] and it has been done for decades using different models and techniques. Depending on whether the forecast horizon is a few minutes, hours, or days, some suggest real-time production-based models, satellite and ground-based models, or weather-based models [9].

We will focus on machine learning algorithms to forecast wind farm production using weather data. In the short term scale (from 1 h to 48 h), two different approaches exist in the literature: the direct one which forecast the production directly from the inputs (mostly meteorological variables) or the indirect one which from the inputs forecast the wind and from the predicted wind forecast the production. It is a one step method against a two steps one which can be summarize in Figs. 1 and 2.

In section "State of the Art on Machine Learning Wind Power Forecasting", state of the art on the short-term forecast in the field of wind energy is done. Section "The Data" gives the details of the data and section "Algorithms of Machine Learning"

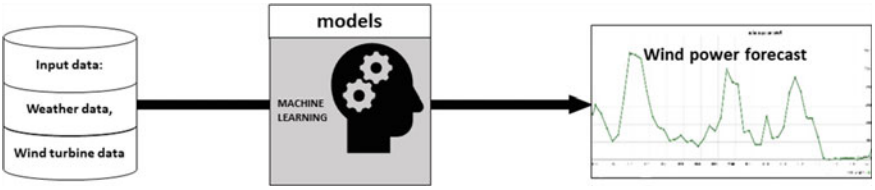


Fig. 1 Direct forecast of wind production

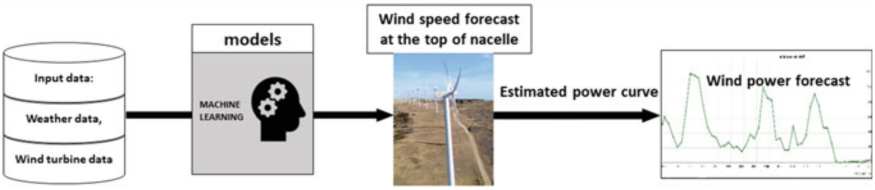


Fig. 2 Indirect forecast of wind power

summarizes the principles of the models applied in our study. The results are discussed in section "Results and Discussions" and finally a general conclusion is reached in section "Conclusion".

State of the Art on Machine Learning Wind Power Forecasting

We find in the literature two types of modeling for the forecast of wind energy: deterministic modeling and probabilistic modeling [10, 11]. Probabilistic models predict directly at each time step of the prediction horizon the probability density. Deterministic models provide a unique value at each time step of the forecast horizon. In the deterministic modeling, we distinguish the physical models and the statistical models. The physical models rely on the modeling of each wind turbine based on the power curve equations [8]:

$$P_w = 0.5 \rho v^3, \tag{1}$$

where P_w the power density, ρ the air density and v the horizontal component of the wind speed. Other statistical techniques rely on direct learning from data. It includes statistical models and machine learning methods used in short-term forecasting. We focus on machine learning methods. The variable to predict is the production of a wind farm P and the statistical criterion used most often to evaluate the forecast errors is the mean absolute error, $MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{P}_i - P_i|$. It is often expressed in terms of percentage of installed power (IP) and becomes $NMAE = 100 * MAE/IP$. The Root Mean Squared Error ($RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\hat{P}_i - P_i\right)^2}$) is often used too. Article [6] discusses the short-term forecast of wind power using a data mining approach by considering data from the Rapid Update Cycle (RUC) model and the North American Mesoscale (NAM) model on sixteen grid points closer to a wind farm. First, the authors select the number of grid points to be considered by the boosting tree algorithm method, then they apply an PCA to reduce the dimension of the variables and finally they apply models of SVMreg, multilayer perceptron network (MLP), radial basis function (RBF), regression trees and random forests. The MLP model gave the best results with an *NMAE* per horizon between 8.41 and 11.49% using as input the RUC data (horizon 12 h) and an *NMAE* per horizon between 5.93 and 13.82% for the NAM input data (horizon 84 h). A predictive study of the wind production of three wind farms with different complexity in France, using a linear model, persistence (reference model) and nonlinear models (neural networks, random forests, and SVM) is made in [7]. The authors used data from the ARPEGE weather forecast model for France over a period of 18 months with a 60 h forecast horizon. Random forests yielded the best results with an *NRMSE* per horizon between 9 and around 15% for forecast horizons ranging from 3 h to 60 h. A study of different probabilistic approaches is made in [4], followed by an evaluation of the deterministic performances (mean of the distributions) with *NMAE* as the evaluation criterion. Their results show an *NMAE* per horizon between 9 and 15% for a 60 h forecast horizon with an overall increase based on forecast horizons. Predicting the density of wind generation using the ECMWF ensemble forecasts is discussed in [12]. In their paper, the authors estimate the mean and the variance of the square of the speed by models AR-GARCH or ARFI-GARCH, then after simulation of 10,000 wind speeds squared according to a Gaussian law, they apply a theoretical power curve to get the production by taking the average of the conversion of the 10,000 speeds. The *MAE* and the maximum log likelihood are used to estimate production forecasts and probability distribution forecasts, respectively. They get on five wind farms (50.25 MW), an average *MAE* between 50 and 150 kW from the first to the tenth day (time step 1 day). It is important to emphasize that in their study, the spatial aspect was not taken into account. The article [5] focuses on the analysis and spatiotemporal modeling of short-term forecast errors in wind generation. With forecasts of the production of 22 wind farms located in Denmark, the authors demonstrated a spatiotemporal structure of production forecast errors, shown the impact of speed and direction on the nature and shape of the structure and proposed a model to capture this structure. After clustering the 22 wind farms, the ACFs and CCFs made it possible to study intra-group and inter-group dependence respectively to demonstrate temporal and spatial correlation. The effects of wind speed and direction were examined by performing an analysis by sector (four sectors) and by velocity interval (interval [0–25] divided into five intervals). They concluded that spatiotemporal information decreased their *NRMSE*.

Among others, parametric regression models, Support Vector Machine (SVM) for regression, classification and regression tree (CART), and random forests are often used. The description of the data is made in the following section.

The Data

The data used in this study come from the European ECMWF model and the wind turbine Supervisory Control and Data Acquisition (scada).

- **ECMWF data**: The European Center for Medium-Range Weather Forecasts is an independent intergovernmental organization funded by 34 states. ECMWF is both a research institute and an operational service, producing and disseminating numerical weather forecasts to its member states. These data are fully available to the national meteorological services of the Member States. The ECMWF data used are hourly forecasts from the 0 h run with a 48 h forecast horizon and a resolution of 0.125° (approximately 13 km in latitude and longitude). We will use the ECMWF model data as input source. With UGRD and VGRD the wind speed W, which is the norm of (UGRD, VGRD) and the direction of the wind (Dir) obtained by tangent arc of (UGRD, VGRD) have been calculated and added in the meteorological variables (Table 1).

Table 1 ECMWF data

Variables	Descriptions	Units
UGRD_10 m	East–West wind component at 10 m above ground	m/s
VGRD_10 m	North–South wind component at 10 m above the ground	m/s
UGRD_100 m	East–West wind component at 100 m above ground	m/s
VGRD_100 m	North–South wind component at 100 m above the ground	m/s
TMP_2	Temperature at 2 m above the ground	°C
SP	Surface pressure	Pa
TP	Total precipitation at the surface	m

Table 2 Scada data

Variables	Descriptions	Units
Power	Power of each wind turbine	Kw
wind speed	Real wind speed on the site	m/s
Operation	Operating index of each wind turbine	

- **Scada data**: the data history of ENGIE Green wind farms is used in this study covering the period from January 2015 to December 2017. This is the production of each wind turbine, its operating condition (operating index) and the wind speed measured by an anemometer on its nacelle every minute. The operating index (equal 0 or 1) depends on the state of the wind turbine at every moment (run, emergency or stop for example), the fact that the wind turbine is able to produce and the fact that the measurement production is not fixed (the scada does not repeat the same measure by malfunction). This index is defined in a way specific to Engie Green and it allows to know at every moment if the value of the production provided by the scada is correct or not. These data are dated in local time. In parallel, wind measurements are also available from a measuring pylon located on each wind farm. But there is more uncertainty about these measures of the pylon (Fig. 4 and Table 2).

Production data and real wind speed are converted to GMT and hourly averages. At each moment t, the production P of the wind farm is equal to the sum of the production of the wind turbines and the real wind speed $W_{nacelle}$ is equal to the average wind speeds measured by the anemometers on the nacelles of the wind turbines. The data are divided into a 2-year learning sample (2015 and 2016) to account for seasonal effects and a one-year test sample (2017).

Let $X^{(i)} = (U_i, V_i, W_i, Dir_i, Temp_i, SP_i, TP_i)$ denote the weather data at a grid point i. Figure 3 shows the sixteen grid points of the ECMWF mesh closest to the wind farm.

For each approach, three situations where tested to study the impact of the spatial aspect: (i) using as input variables X, the weather data at the closest

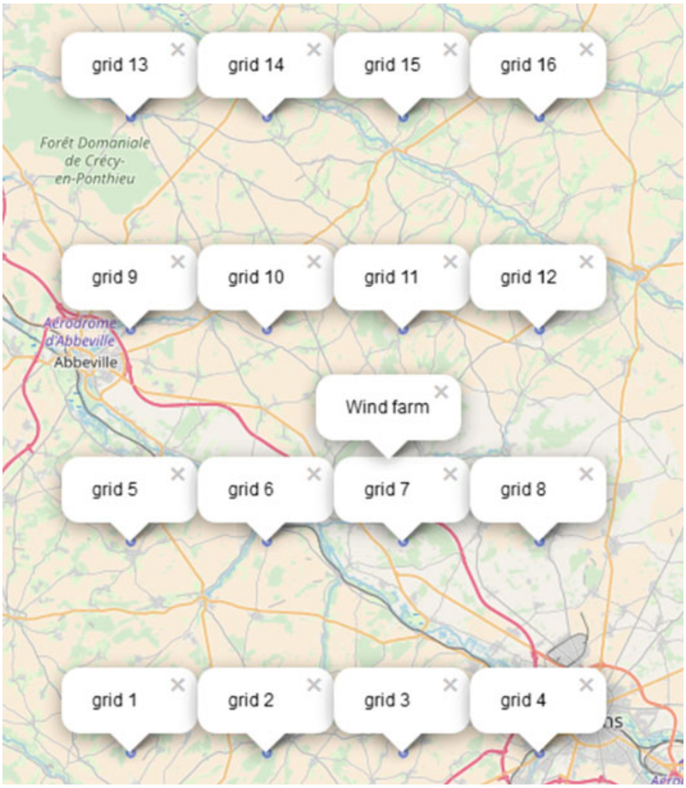


Fig. 3 Sixteen grid points of the ECMWF grid closest to the wind farm (map done with leaflet package from Rstudio)

grid point of the wind farm ($X = X^{(7)}$), (ii) to the four closest grid points ($X = (X^{(6)}, X^{(7)}, X^{(10)}, X^{(11)})$) and (iii) to the nearest sixteen grid points ($X = (X^{(1)}, \cdots, X^{(16)})$). Finally, we will explain the spatiotemporal analysis (Section "Feature Engineering: Spatiotemporal Dynamics of Wind") to model wind dynamics and add it to the input data. For each approach there are two steps in the modeling: the training step and the test step. For example, the indirect approach is as follows:

- Training step:
  - Estimate the real power curve: $P = g(W_{nacelle}) + \eta$ by a smoothing spline to get $\hat{g}$,
  - Wind speed: $W_{nacelle} = h(X) + \zeta$ to learn the wind speed forecast model on the wind farm and obtain the estimator (or the forecasting model) $\hat{h}$ of the unknown function h (a machine learning algorithm, see section "Algorithms of Machine Learning"),

- Test step: use the day head weather data X, to forecast the wind speed on the wind farm $\hat{W}_{nacelle} = \hat{h}(X)$ and then transform the forecasted wind speeds into wind power forecast $\hat{P} = \hat{g}(\hat{W}_{nacelle})$. $W_{nacelle}$ is the real wind speed, P the production of the wind farm, η and ζ are random errors.

In the following section we present the principles of machine learning methods used.

Algorithms of Machine Learning

From the 1980s, the development of computer science facilitated the implementation of nonlinear methods. Classification and regression trees will be introduced in [13]. Breiman [14] then introduced random forests in 2001. Usually the statistical framework is as follows: we want to predict a variable Y from p explanatory variables $X = (X_1, X_2, \ldots, X_p)$ which are vectors. In a general way the statistical model is written: $Y = f(X) + \varepsilon$, where f is a function (Model) estimated minimizing the errors ε. In our case Y will be either the wind farm production P or the wind speed at the top of nacelle $W_{nacelle}$. We apply some models commonly used in the literature.

Persistence

The persistence model is a widely used reference model for wind energy forecasting. It simply consists of using the last observation as a forecast for all horizons [7]. Specifically, $\hat{P}_{t_0+h} = P_{t_0}$ with $\hat{P}$ representing the forecasts, P the measures, t_0 the initial forecast time and h the forecast horizon.

Bagging

Decision trees (see [13] for details) suffer from a large variance. That is, if the training data are randomly separated into two parts, and a decision tree is adjusted on each set, the results obtained could be very different. A natural way to reduce variance is to have multiple samples of the population, build a predictive model on each sample separately, and average predictions. In other words, we calculate $\hat{f}^1(x),\ \hat{f}^2(x),\ \ldots,\ \hat{f}^B(x)$ using B learning sets and we average,

$$\hat{f}_{avg}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(x), \tag{2}$$

to obtain a lower variance. In practice, no more samples are available. However, we can obtain several B samples per bootstrap in order to calculate $\hat{f}^{*b}(x)$ and to average the predictions to obtain,

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x) \tag{3}$$

this is called bagging. The principle remains the same in the context of a regression. In place of decision trees, regression trees will be constructed as part of the use of bagging for regression. The idea of Bagging is that by applying the basic rule on different bootstrap samples, the predictions are modified, and thus a collection of various predictors is constructed. The aggregation step then makes it possible to obtain a powerful predictor.

Boosting

Boosting was introduced by [15] in 1996. This is a general approach that can be applied to several statistical learning methods for regression or classification. Boosting is similar to bagging, except that trees are built sequentially: each tree is built using the information from the previous tree. The principle of boosting is to draw a first bootstrap sample $B_n^{\Theta 1}$, where each observation has a $1/n$ probability of being drawn, and then apply a basic rule (regression tree in our case) to get a first predictor $\hat{f}(., B_n^{\Theta 1})$. Then, the error of $\hat{f}(., B_n^{\Theta 1})$ on the training sample is computed. We draw a second bootstrap sample whose law of drawing of the observations is no longer uniform. The probability for an observation to be drawn depends on the $\hat{f}(., B_n^{\Theta 1})$ prediction on this case. The principle is to increase the probability of drawing an incorrectly predicted observation, and to diminish that of drawing a well-predicted observation. The basic rule is applied to this new sample and then the process is repeated. The collection of predictors obtained is then aggregated using a weighted average.

Random Forests (RF)

Since its introduction by Breiman [14] in 2001, many publications have addressed the theory of random forests with applications in several areas. The principle of random forests is first of all to generate several bootstrap samples $\mathcal{B}_n^{\Theta_1}, \ldots, \mathcal{B}_n^{\Theta_q}$. Then, on each sample $\mathcal{B}_n^{\Theta_l}$, a variant of CART (Classification And Regression Tree) is explained. In other words, a tree is constructed in the following way. The division of a node is done by a random draw of m variables and the search for the best break according to the selected m variables. In addition, the constructed tree is fully developed (maximum tree) and not pruned. Pruning consists in looking for the best sub-tree pruned from the maximal tree (better in the sense of the generalization error). The collection of trees obtained is finally aggregated (average in regression, majority vote in classification) to give the predictor of random forests. The draw, at each node, of the variable m is done, without discount, and uniformly among all variables (each variable has a $1/p$ probability of being chosen). The number m ($m \leq p$) is set at the beginning of the forest's construction and is, therefore, identical for all trees. This is a very important parameter of the method.

For Random Forests, there are, therefore, two sources of randomness to generate the collection of individual predictors: the randomness due to the bootstrap and the random choice of variables to cut each node of a tree. Thus, one disturbs both the sample on which one launches the basic rule, and the heart of the construction of the rule of base. This random draw of variables to cut a node had already been used by Amit and Geman [16] in image recognition problems. Their method greatly influenced Breiman in his development of Random Forests.

Support Vector Machine Regression

Support Vector Machines are a class of learning algorithms initially defined for discrimination, that is, predicting a binary qualitative variable [17]. They were then generalized to the forecast of a quantitative variable. We first consider the case of linear SVM Regression. Let a training data (Y, X) where X is a multivariate set of N observations with observed response values Y. The goal is to find a function $f(X) = X^{'}\beta + b$ that deviates from Y by a value no greater than ε for each training point X, and at the same time is as flat as possible. The formulation in convex optimization problem is to minimize:

$$J(\beta) = \frac{1}{2}\beta^{'}\beta \qquad \text{subject to} \qquad \forall i \; : \; |Y_i - f(X_i))| \leq \varepsilon. \tag{4}$$

If there is no such function $f(.)$ that satisfies these constraints for all the points, slack variables ξ_i and ξ_i^* are introduced for each point. Including slack variables leads to the objective function:

$$J(\beta) = \frac{1}{2}\beta^{'}\beta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*), \tag{5}$$

subject to $\forall i \; : \; Y_i - f(X_i) \leq \varepsilon + \xi_i, \; f(X_i) - Y_i \leq \varepsilon + \xi_i^*, \; \xi_i \, , \xi_i^* \geq 0.$

C is a positive constant that controls the penalty imposed on observations that lie outside the epsilon margin (ε) and helps to prevent over fitting (regularization). The optimization problem previously described is computationally simpler to solve in its Lagrange dual formulation by introducing non negative multipliers α_i and α_i^* for each observation X_i. The function used to predict new values depends only on the support vectors:

$$f(X_j) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)(X_i^{'}X_j) + b. \tag{6}$$

To obtain nonlinear SVM regression one can replace the product $X_i^{'}X_j$ with a non linear kernel function $G(X_i^{'}, X_j)$ (for example Gaussian kernel $G(X_i, X_j) = \exp\left(- \parallel X_i - X_j \parallel^2\right)$, or polynomial kernel $G(X_i, X_j) = (1 - X_i^{'}X_j)^q$, where $q \in \{2, 3, ...\}$).

Generalized Additive Models

Generalized additive models were introduced by Hastie and Tibshirani in [18]. They constitute a generalization of multiple regression. In linear regression, we calculate a linear least squares adjustment for a set of variables $X = (X_1, \cdots, X_p)$ to predict a variable Y. The equation of linear regression can be formulated as follows:

$$Y = b_0 + b_1 X_1 + \cdots + b_p X_p + \varepsilon. \tag{7}$$

Generalized additive models replace the linear form $(\sum \beta_j X_j)$ with a sum of smooth functions $\sum s_j(X_j)$. The $s_j(.)$ are unknown functions that can be estimated by any scatter plot smoother (least squares, splines,...) with a “local scoring” procedure (see [18] for more details). The GAM package is used for the GAM model fit in this study.

Results and Discussions

In this study the wind power forecast of ENGIE Green wind farms is presented. We used the weather forecast at the nearest grid point, at the closest four grid points and then at the 16 grid points closest to the wind farm (Fig. 3). The forecast horizon is from 24 to 48 h because we are interested in the forecasts of wind production on day D + 1, knowing that we are at day D.

Results of Machine Learning Algorithms

We present in this section the results of the different machine learning algorithms (Section “Algorithms of Machine Learning”). *NMAE* is used as a statistical indicator. Taking the direct approach and the indirect approach as well as the modeling at one grid point, four grid points and sixteen grid points, we compare the results of the different methods.

Table 3 shows that, apart from the GAM with the weather forecast at the nearest grid point as input sources, random forests provide the weakest *NMAE* compared to other machine learning methods when the conditions of application are the same. For random forests, bagging and boosting, taking into account weather data at four and then sixteen grid points reduces the *NMAE* compared to models integrating only the weather data at the nearest grid point. However, for SVMs, taking into account more than one grid point increases production forecast errors. Taking several grid points does not systematically reduce forecast errors; this may depend on the machine learning algorithm used. The machine learning algorithms have comparable results with a slight advantage for GAMS and Random forest for this farm.

Table 3 *NMAE* forecast models

Methods	Approach	1 grid point	4 grid points	16 grid points
RF	Direct	8.56	8.65	8.38
	Indirect	**7.70**	**7.76**	**7.62**
Bagging	Direct	8.32	8.24	8.15
	Indirect	7.92	7.77	**7.70**
Boosting	Direct	8.34	8.35	8.35
	Indirect	8.07	8.10	8.23
SVM	Direct	8.03	8.40	8.64
	Indirect	7.68	7.75	7.76
GAM	Direct	7.98	8.74	9.75
	Indirect	**7.65**	8.09	8.85
Persistence	20.40			

The same conclusions were observed by generalizing the study on two other wind farms of 13.8 MW (wind farm 2) and 12.3 MW (wind farm 3) located at different places and with different topography neighborhood. The results are not presented but the order of magnitude of the errors are the same.

In the following section we focus on the comparison between the direct and the indirect approach.

Direct Approach/Indirect Approach

The indirect approach reduces the *NMAE* of wind power forecasts when using a grid point, four or sixteen grid points on the wind farm 1 and 2 and less on the wind farm 3. Let's analyze these results more precisely. We have the ECMWF wind at 100 m at the nearest grid point denoted W and the wind speed at the top of the turbine denoted $W_{nacelle}$. In the case of the indirect approach, the wind is predicted at the head of the wind turbine by a random forest using different ECMWF data denoted $\hat{W}_{nacelle}$. We compare in the following table the *MAE* between the W and $W_{nacelle}$ and the *MAE* between $\hat{W}_{nacelle}$ and $W_{nacelle}$. If we look at Table 4, we find that the predicted wind reduces the *MAE* wind forecast errors on the three wind farms. So the intermediate step of wind forecasting at the wind turbine reduces the *MAE*.

What is important to note is that even if the meteorological wind forecast errors are reduced, if the transfer curve used in the indirect approach is imprecise, the indirect approach may be less advantageous. In fact, the Fig. 4 on the right is only an example of the transfer curve in the direct case with only one input variable (the wind speed) to illustrate the difference between the transfer curves in the direct and indirect approach (transfer curve at left). For example, if we use the power curve

Table 4 *MAE* with wind ECMWF compared to *MAE* with wind forecasted at the top of wind turbines

Wind speed	MAE (m/s) at the nearest grid
Wind farm 1	
ECMWF wind speed at 100 m	1.34
Wind forecast by RF	0.95
Wind farm 2	
ECMWF wind speed at 100 m	1.66
Wind forecast by RF	1.22
Wind farm 3	
ECMWF wind speed at 100 m	1.06
Wind forecast by RF	0.78

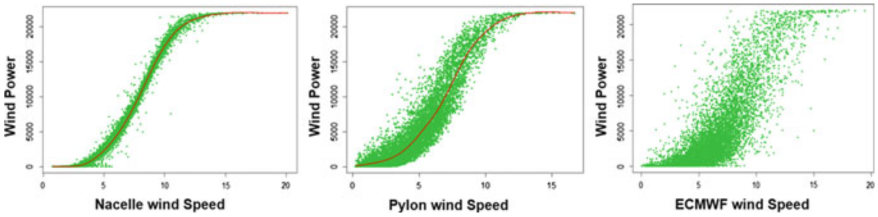


Fig. 4 Transfer curve with anemometer wind on the wind turbines (left), with wind pylon (in the middle) and with wind ECMWF (right)

with wind measurements with a lot of uncertainties (middle), the indirect approach becomes less advantageous.

For the farm 3, the real transfer curve of the wind farm is much more imprecise. There were wind measurement deficiencies of anemometers on two wind turbines among the six that makes up the wind farm 3.

Feature Engineering: Spatiotemporal Dynamics of Wind

In this part, a feature engineering of the meteorological data is done to model the spatiotemporal dynamics of the wind. For each input variable we approached the temporal variation and integrated it into model input. For example, for the wind component *UGRD*, at a grid point i at an instant t, the temporal variation will be $UGRD_{(i,t)} - UGRD_{(i,t-1)}$.

Table 5 *NMAE* with integration of spatiotemporal variations

Models	1 grid point	4 grid points	16 grid points
RF direct	8.56	8.65	8.38
RF indirect	7.70	7.76	7.62
Integration of spatiotemporal variations			
RF direct	8.21	8.28	8.21
RF indirect	7.45	7.57	7.44

To model the spatial variations of the wind at a grid point in an instant t, we calculate for the wind components (UGRD, VGRD, W and Dir) the vertical and horizontal variations between this grid point and the other two grid points located at the top on the vertical axis and left on the horizontal axis at the same time t. For example for UGRD at grid point 6 (Fig. 3) in one instant t, we get $(UGRD_{(6,t)} - UGRD_{(7,t)},\ UGRD_{(6,t)} - UGRD_{(10,t)})$ as spatial variations.

By adding these two derivatives as input variables, we model the spatiotemporal dynamics of the wind (wind gradient).

The number of input variables will increase from 11 to 27 (in the case of a grid point), from 44 to 108 (in the case of 4 grid points) and from 176 to 432 (in the case of 16 grid points). We present in the Table 5, the integration of spatiotemporal variations. In the three modeling cases (one grid point, four grid points and sixteen grid points) there is a reduction in forecast errors compared to the first case where the derivatives are not used. This improvement in forecasts come from the fact that, in addition to the one-time meteorological data t, we take into account the spatiotemporal variations of the wind at this moment in order to predict the production. In other words, at each instant t, the spatiotemporal variations in addition to the instantaneous meteorological data better explain the wind production at this moment. Integrating the spatiotemporal variations increases the results by 3% and the same results were observed for the two other wind farms.

Conclusion

In this paper, different statistical models applied to the short-term forecast of wind energy production are studied. The ECMWF weather model forecasts are used as input data. GAMs and Random Forests had a slight advantage over the short-term forecast (horizon 24–48 h) of wind generation that the other machine learning algorithms tested. Our results showed a reduction in forecast errors with the indirect approach due to improved wind forecasts and a better transfer curve in the indirect case. It is important to take into account a good real transfer curve in the indirect approach. If the local wind reference are rather imprecise, the resulting power curve of these measurements and the production of the wind turbines is not as good and the use of the indirect approach may not be more advantageous than the direct approach. We have also found that the use of the wind gradient provides an additional setback and thus reduces the *NMAE* of wind production forecasts. The use of several grid points does not systematically reduce the forecast errors (this depends on the machine learning algorithm used). In perspective we will refocus on how to take into account wind regime change and also will use deep learning for forecasting wind power. Selecting the grid point according to the importance (precision) of the meteorological variables on each grid point could also improve the production forecasts and especially reduce the dimension of input data.

Acknowledgements This work was done in the framework of the project Predict'EOL which is a PhD project between Engie Green France and CREST-ENSAE ParisTech. The authors would like to thank ENGIE Green and in particular P. Alexandre leading the Studies and Innovation team, C. Penin and N. Girard for discussions which improve the quality of the work. We also would like to thank Meteo France for providing the weather data for this study.

References

1. Renewable 2017 global status report. Technical report. REN21 Secretariat, Paris (2017)
2. Panorama de l'électricité renouvelable au 31 mars 2017. Technical report, Syndicat des Énergies Renouvelables (2017)
3. Panorama de l'électricité renouvelable en 2015. Technical report, Syndicat des Énergies Renouvelables (2015)
4. Fugon, L., Juban, J., Kariniotakis, G.: Uncertainty estimation of wind power forecasts: comparison of probabilistic modeling approaches. In: European Wind Energy Conference Exhibition EWEC 2008, March 2008, Brussels, Belgium, 10 p. EWEC (2008). http://www.ewec2008proceedings.info/. <hal-00506297>
5. Kotwa, E.K., Madsen, H., Tastu, J., Pinson, P., Nielsen, H.A.: Spatiotemporal analysis and modeling of short term wind power forecast errors. Wind Energy **14**(1), 43–60 (2011)
6. Zheng, H., Kusiak, A., Song, Z.: Wind farm power prediction: a data-mining approach. Wind Energy **12**(3), 275–293 (2008)
7. Juban, J., Fugon, L., Kariniotakis, G.: Data mining for wind power forecasting. In: European Wind Energy Conference, p. 6. EWEC, Brussels, Belgium (2008). <hal-00506101>
8. Madsen, H., Costa, A., Crespo, A., Navarron, J., Lizcano, G., Feitosa, E.: A review on the young history of the wind power short-term prediction. Renew. Sustain. Energy Rev. **12**(6), 1725–1744 (2008)
9. Najac, J.: La prévision de production éolienne et photovoltaïque à edf. Séminaire In'Tech, INRIA (2012)
10. Kariniotakis, G., Giebel, G.: Wind Power Forecasting—A Review of the State of the Art, pp. 59–109. Woodhead Publishing (2017)
11. Giebel, G., Kariniotakis, G.: Best practice in short-term forecasting. A users guide. Technical report (2009)
12. McSharry, P.E., Taylor, J.W., Buiza, R.: Wind power density forecasting using ensemble predictions and time series models. IEEE Trans. Energy Convers. **24**, 775–782 (2009)
13. Friedman, J., Stone, C.J., Breiman, L., Olshen, R.A.: Classification and Regression Trees, pp. 5–32. Chapman & Hall/CRC (1984)
14. Breiman, L.: Random forest. In: Machine Learning (2001)
15. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the 13th International Conference on Machine Learning, pp. 148–156, San Francisco (1996)

16. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Comput. **9**(7), 1545–1588 (1997)
17. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
18. Hasti, T., Tibshirani, R.: Generalized additive models. Stat. Sci. **1**(3), 297–318 (1986)

Time Series Analysis and Prediction in Other Real Problems

A Simulation of a Custom Inspection in the Airport

Kalle Saastamoinen, Petteri Mattila and Antti Rissanen

Abstract Time is an essential part when traveling since it is a very time-critical process. One main issue when going abroad is a border crossing. Helsinki–Vantaa Airport (HEL) has extensive flight connections and the shortest routes between Europe and Asia, making it a major hub for Northern Europe. The number of external border traffic at Helsinki–Vantaa Airport is increasing steadily. In this study, we will study through simulation if Helsinki–Vantaa Airport is able to answer future challenge with estimated amount of passenger to increase 74% by 2022 by using cooperation-based data processing in entry and exit checks in the border inspection. Main result of simulation is that by exchanging traveler's information automatically between different authorities, this future increased amount of traffic can be taken care of without increase of airport capacity, so that airport traffic still stays fluent. This simulation model can be used if we want to measure impacts of renovations and increased traveler's flows to the changes of border inspection time.

Keywords Airport · Passenger information · Border inspection · Data · Simulation

Introduction

Here simulation is used to show the real effects of alternative conditions and courses of action. In order to establish our simulation model, we used queueing theory that is the mathematical study of waiting lines or queues [1]. This study can be considered to be part of operations research since the simulation results are used to make decisions about the resources needed to provide a service.

K. Saastamoinen (✉) · P. Mattila · A. Rissanen
Department of Military Technology, National Defence University, P.O. Box 7,
FI-00861 Helsinki, Finland
e-mail: kalle.saastamoinen@mil.fi
URL: http://www.puolustusvoimat.fi/en/

P. Mattila
e-mail: petteri.mattila@mil.fi

A. Rissanen
e-mail: antti.rissanen@mil.fi

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_22

In modern traveling, airport operations and the implementation of border checks are very time-critical. Short connecting flight schedules at Helsinki–Vantaa Airport, coupled with high level of the passengers passing through the airport to the next flight, require smoothly running border control process. This demand challenges the border inspection capability to be flexible and able to fill the level required for a fluent border traffic. Passengers' delays from connecting flights can be a significant operational and financial risk for airlines and ultimately passengers themselves. The border inspection authority has no legal right to waive border controls or to change the content of inspections to ensure the smooth running of traffic. The flow of traffic cannot get priority over the border control protocols that are in line with the regulations. However, it is very important that the border inspection process is operated without significant queuing times [2].

Main research question is to simulate how well can Helsinki–Vantaa Airport answer future challenge with increasing number of passengers and can new demands be handled using cooperation-based data processing with different participating operators? Where main benefits are avoidance of overlapping information and use of automated border checks in entry and exit.

In this article, first chapter presents the problem in hand at Helsinki–Vantaa Airport, second chapter goes through possibilities of collaboration-based data handling, third chapter presents the simulation model and results, and fourth chapter gives conclusions and discusses future directions of this research.

Problem Description

The development of Helsinki–Vantaa Airport emphasizes the growth of passenger traffic across the Schengen border, which has a direct impact on the Border Guard's operations as well. Estimated growth forecast for traffic will take into account Finavia's (Formerly the Finnish Civil Aviation Administration that is responsible for maintaining and developing its 21 airports and Finland's air navigation system) and Finnair plans and projections. The largest operator in the HEL airport is Finnair, which invests in traffic between Europe and Asia. Currently, Asian traffic accounts for a little over 50% of Finnair's traffic [3]. Finavia's future growth strategy underlines the strong growth of transit travelers. In Helsinki–Vantaa, this means that Asian passenger traffic will increase, which significantly adds the number of passengers with a visa requirement [4].

Helsinki–Vantaa Airport has extensive flight connections and the shortest routes between Europe and Asia, making it a major hub for Northern Europe. When planning for a smooth running of the border control process, it is good to know that many of the passengers pass through the airport only to switch the flight. The number of external border traffic at Helsinki–Vantaa Airport is increasing steadily. Table 1 shows estimation for number of passengers up to the year 2035. Estimation forecasts an average growth rate of 4.7% per annum. Traffic growth is expected to be particularly high on third-country passengers.

Table 1 Estimated passenger growth between 2013 and 2035 [5]. Published with permission of Finavia

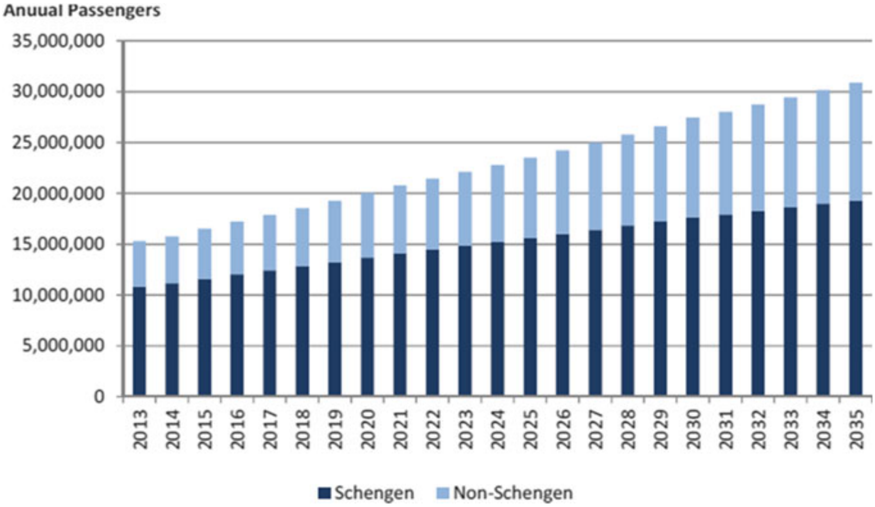
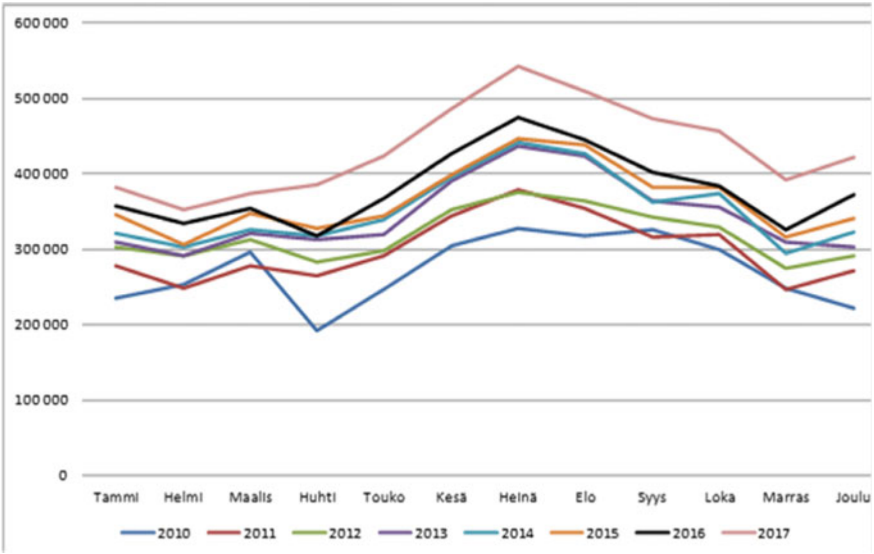


Table 2 shows the realization and development of cross-border traffic in the Helsinki–Vantaa external border traffic 2010–2017. In 2016, Helsinki–Vantaa's total passenger traffic was 17.2 million passengers (Table 2). During the year 2017, the growth was 800,000 passengers. By the year 2022, the number of passengers on

Table 2 Development of traffic in 2010–2017 [7]. Published with permission of Finavia



external border traffic is estimated to increase approximately by 74%, from 4.6 million (2016) to 8 million passengers. In 2017, more than 5.2 million border checks were carried out [6].

The Head of the Coast Guard of the Gulf of Finland has pointed out that Helsinki—Vantaa Airport's volume and profile of the passenger traffic, especially during the afternoon hours, pose a challenge for the operators, airline carriers, and authorities [8]. To respond to the challenge, the Border Guard's operational processes are continuously evaluated and developed. Efforts to streamline and manage passenger traffic have been attempted to meet the needs of growing passenger flows. These actions include the efficient use of personnel and passenger guidance as well as other development measures for the border inspection process. The passenger guidance system has been developed as a cooperation between the airport authority and the Border Guard [9].

In the actual border checks, processes have been continuously enhanced. As an example, automatic self-service border control desks in the Helsinki–Vantaa Airport have helped to speed up the increase in passenger flows. The major factor in the automated border control utilization is the increase in the proportion of biometric passports. Safety regulation and features for biometrics passports and travel documents are provided at EU level [10]. As such, they made an important step toward the development of technical components at border control in Europe.

The share of EU/EEA/CH nationals in the Helsinki–Vantaa external border traffic has been steadily about 70% of the total number of passengers traveling each year. In 2017, EU passengers accounted for around 66% of Helsinki–Vantaa's external border traffic. The number of automatic border checks on total traffic has still been below 40% [7]. However, the predicted growth of passenger traffic in the group of third-country nationals means also growth in the group with limited access to the automated system, which demands biometric passport [7, 11]. This necessitates the need for the development of audit processes. As an operating environment, the airport requires that the border authority must take into account, in its own activities, the time- and activity-related factors determined by the carrier and the airport [8]. These are important for the airport competitiveness. Effective processes are important success factors. Strengths of Helsinki–Vantaa Airport are short distances and at best plane can be changed in 35 min. Quick access to connecting flights is a competitive factor and as Asia's transit traffic increases, the smoothness of processes is an essential part to the success of the airport [12].

The challenge for border checks at Helsinki–Vantaa Airport is the fluctuations in the time distribution of border-crossing air traffic. This means number of passengers' to change heavily according to the time. Peaks are related to the structure of Finnair flights. In the afternoon, the congestion of border checks for incoming and outgoing flights is mainly due to the so-called "the Asian wave". At that time, Finnair's traffic generates incoming traffic at border checks of up to 3600 passengers per hour. As an example, if Finnair's long-distance traffic would add two daily flights that would mean that the number of border-crossing individuals would increase between 400 and 600 border-crossing individuals per day. On an annual level, increase would be up to 200,000 border checks more.

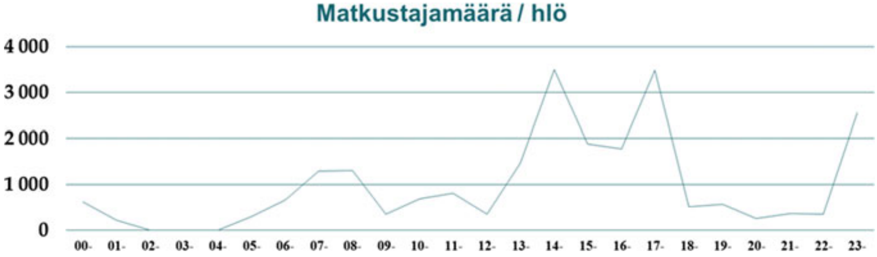


Fig. 1 Time fluctuation in passenger volume for 24 h at Helsinki–Vantaa [13]. Published with permission of Finavia

According to Finavia's data, in summer, especially in May, the number of passengers is growing strongly [13]. Figure 1 illustrates the hourly variation for air travelers on May 31.

When assessing the number of passengers, Finnair's own growth forecasts should not be considered as the only source of information. With regard to Helsinki–Vantaa, it would also be worth noting the growing interest of all airline carriers of transit traffic in Helsinki–Vantaa Airport. The admission of new companies to airlines operating via Helsinki–Vantaa may have a major impact on passenger transport growth. However, making forecasts is challenging because it is hard to anticipate passenger flows. Mobility is affected by various phenomena and decisions made outside Finland's borders. Surprising changes in people's behavior, mobility, and travel can be caused by international security situation, immigration, or climate change [14].

Automation can significantly assist in verifying a person. The whole set of personal and travel information that has been identified at various stages of the border check process has become complex. Technical progress has made it possible to carry out checks of documents and persons in parallel. Future factors will be the automation of the audit process and the wider utilization of biometric identifiers. The border inspection process is then advancing toward solutions in which at the border-crossing point the passengers themselves sovereignly master the procedures while the authorities are there mainly to supervise and help. Technical solutions can ensure the reliability of inspection activities and reduce queues at border-crossing points. The implementation of technical services to facilitate the flow of cross-border traffic requires the pooling of data for different authors [15].

Possibilities of Collaboration-Based Data Handling

In order to identify the information needed for border control, a border inspection application is utilized. The necessary data in the application is combined for sources to allow the user to view, process, and compare different data. The Border Inspection Application is part of the Border Guard's Operational Information System (RVT) where all data entered at the border check is stored [11].

Air carriers provide passenger information to the border inspection authority system Ratas using a secure connection using a standardized message format. At reception of passenger information, an aero-based MQ-based network is used. The communication follows the standards defined by the International Civil Aviation Organization (ICAO), the World Customs Organization (WCO), and the International Air Transport Association (IATA). The message format is the PAXLST (passenger list) message standard that passes within the Type B frame. The Border Guard has its own address on the network, where the sent messages are sent to the Ratas system of the Frontier Guard [2].

Passenger information and regulation on border checks revealed the convergence of entries in border checks in relation to carriers' data in air traffic. It is clear that the current passenger data collected by the air carrier contains data which will also be surveyed and recorded in the entry and exit checks. Passenger information can be used to find out the necessary information without requiring the verification of documentary evidence presented in support of this information.

Unlike Advance Passenger Information (API) data, Passenger Name Record (PNR) data cannot be used to support border checks in accordance with the directives [16]. The data contained in the PNR data set is related to the information needed to determine the cross-border motive in order to establish the purpose and conditions of the intended stay. The duration of the planned stay PNR data supports API data in order for officer to determine the real destination.

As a whole, these "pre-requisite border controls" can be used, for example, in entry checks. An air carrier collects the following information that is consistent with the information available at the border inspection:

1. Country of Departure (API),
2. Transport, ID (API),
3. Destination (API), and
4. Planned (Scheduled) Travel Date (s) (PNR).

To make better use of passenger data, automation is needed to process the information for acting border inspector. API data is designed for originally arriving passengers, so utilization for boarding is a challenge. Wider use of this data does not seem possible because of the delivery time specified in the law. This change would require both a review of the legal basis and new technical arrangements [2].

Uniform operating models for collecting data may also contribute to collaborative information processing. Cooperation-based data processing is not only related to the use of passenger data. Automation is not just a check-in automation. It is expected that technological solutions will not only be limited to checking-in or making reservations but also the services will be further extended in the future [2]. At the same time, it is necessary to ponder the question whether or not independent controls are needed for border controls or would they be more efficient when the activities are linked together into the entire travel process? From the point of view of border checks, identification data from travel and personal documents can be obtained from the check-in desk. Primarily, the use of check-in machines would be airport specific and related to the start-up inspection. The use of check-in machines for entry checks is technically complicated both in collecting and delivery of data [2].

Modeling and Results

Border inspection process can be described by the following flowchart (Fig. 2):

In order to define correct parameters for the simulation model, we did empirical testing about inspection times at Helsinki–Vantaa Airport. Test was carried out on February, 19–21, 2018 and concentrated to the customers coming outside Europe. Sample size was 50 and samples were selected randomly.

In our simulation we were only interested how long time it takes to go through border inspection. Simulation model helped to find critical points from the process and gave valuable insight how time-critical border checkup time is with respect to resources available. Simulation here models multiphase process, from where one can see how passengers' incoming and exiting to the process affect the queue length. Using this simulator, we tested how much sharing information between officials and use of automated border checkup could boost up passenger's flowrate.

Border checkup is queue process. Phases are arrival, waiting, service, and exit. Bottleneck in practice in this process is waiting time in queue. In HEL, border checkup is organized as follows (Fig. 3):

Here we have one line with many servers. In order to make border check as effective as possible, we can affect mainly to the effectiveness of servers. Service

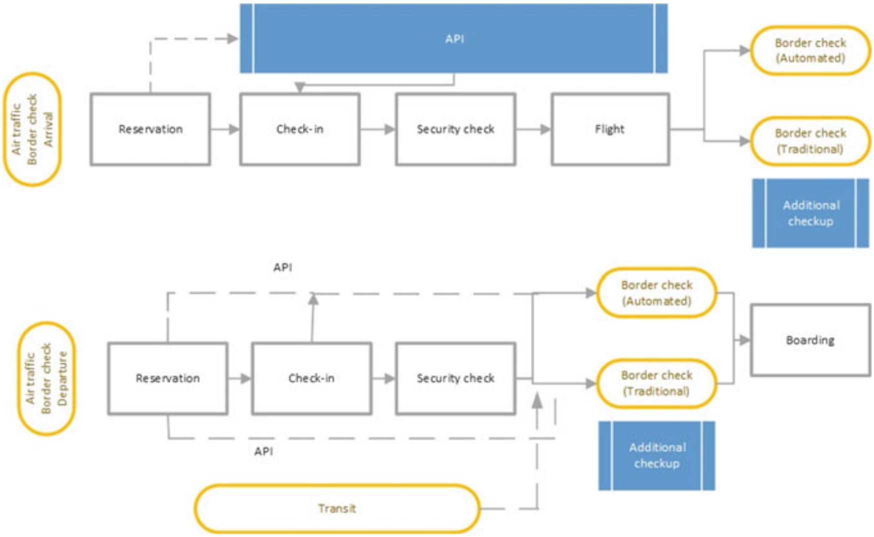


Fig. 2 Process flowchart of the passengers' border crossing

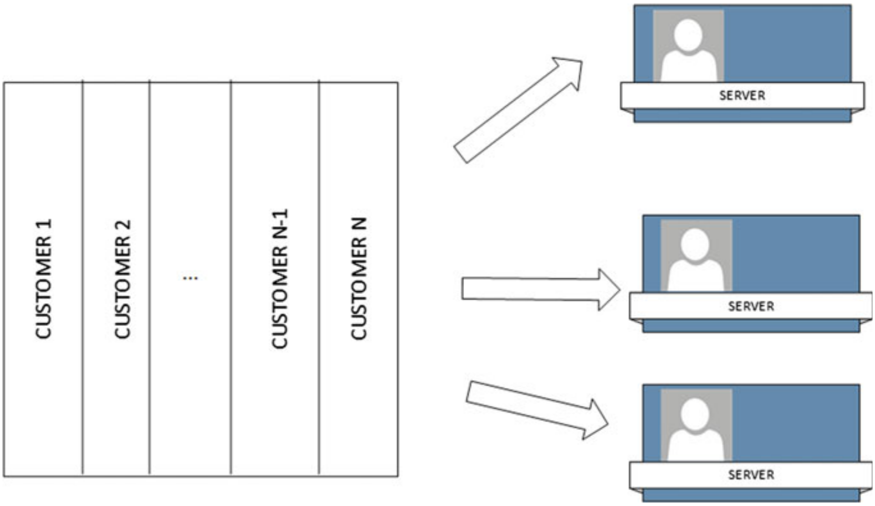


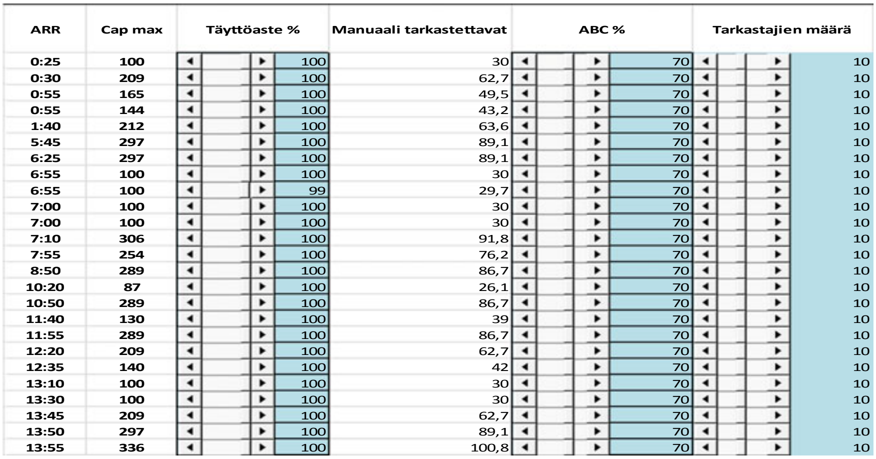
Fig. 3 One line many servers

time distribution is a key factor in the fluent working of the queue. In the simulation model, the following principles form the working model: (1) Number of customers in the queue. (2) Number of customers in the whole system. (3) Average time customers wait in the queue. (4) Average time of the border inspection. (5) Average time in the system.

The model simulated the timing of the queue length by calculating passenger arrivals and exits at border inspection. Flight Passenger Parameters could be determined per flight. The user interface included setting the flight fill rate. The average waiting time for the first incoming flight queue was assumed to be empty at first. The simulation was continued until the last flight had arrived and the passengers had passed the inspection. In this case, the observation from a single simulation time was the average of all wait and wait times for all the passengers in that simulation run (Fig. 4).

A total of 22 manual and 20 automated inspection lines are available for border checks of incoming passengers. Manual lines 8 are located in the first floor and the rest 14 in the second-floor spaces. Border checks on outbound passenger traffic are carried out on one floor, where 16 manual and 15 automated control lines are available [6]. In order to shorten the waiting times, there are separate routes for passengers belonging under the European Union's free movement right [17]. As a result of these arrangements, a total of 20 inspection points are available for third-country nationals and 15 manual checkpoints on outbound checks. With the terminal extension, the capacity of external border controls will also increase in the future. Exact line numbers have not yet been established [6], so existing border control capacity was used in simulation and analysis of results.

The number of flights varies daily. Through the simulation it was essential to illustrate the duration of the checks in relation to the resources used. To illustrate the



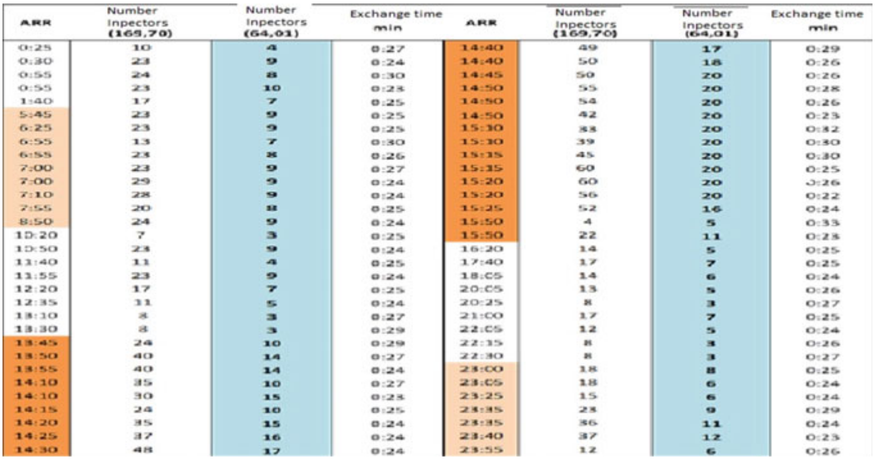
ARR	Cap max	Täyttöaste %	Manuaali tarkastettavat	ABC %	Tarkastajien määrä
0:25	100	◂ ▸ 100	30	◂ ▸ 70	◂ ▸ 10
0:30	209	◂ ▸ 100	62,7	◂ ▸ 70	◂ ▸ 10
0:55	165	◂ ▸ 100	49,5	◂ ▸ 70	◂ ▸ 10
0:55	144	◂ ▸ 100	43,2	◂ ▸ 70	◂ ▸ 10
1:40	212	◂ ▸ 100	63,6	◂ ▸ 70	◂ ▸ 10
5:45	297	◂ ▸ 100	89,1	◂ ▸ 70	◂ ▸ 10
6:25	297	◂ ▸ 100	89,1	◂ ▸ 70	◂ ▸ 10
6:55	100	◂ ▸ 100	30	◂ ▸ 70	◂ ▸ 10
6:55	100	◂ ▸ 99	29,7	◂ ▸ 70	◂ ▸ 10
7:00	100	◂ ▸ 100	30	◂ ▸ 70	◂ ▸ 10
7:00	100	◂ ▸ 100	30	◂ ▸ 70	◂ ▸ 10
7:10	306	◂ ▸ 100	91,8	◂ ▸ 70	◂ ▸ 10
7:55	254	◂ ▸ 100	76,2	◂ ▸ 70	◂ ▸ 10
8:50	289	◂ ▸ 100	86,7	◂ ▸ 70	◂ ▸ 10
10:20	87	◂ ▸ 100	26,1	◂ ▸ 70	◂ ▸ 10
10:50	289	◂ ▸ 100	86,7	◂ ▸ 70	◂ ▸ 10
11:40	130	◂ ▸ 100	39	◂ ▸ 70	◂ ▸ 10
11:55	289	◂ ▸ 100	86,7	◂ ▸ 70	◂ ▸ 10
12:20	209	◂ ▸ 100	62,7	◂ ▸ 70	◂ ▸ 10
12:35	140	◂ ▸ 100	42	◂ ▸ 70	◂ ▸ 10
13:10	100	◂ ▸ 100	30	◂ ▸ 70	◂ ▸ 10
13:30	100	◂ ▸ 100	30	◂ ▸ 70	◂ ▸ 10
13:45	209	◂ ▸ 100	62,7	◂ ▸ 70	◂ ▸ 10
13:50	297	◂ ▸ 100	89,1	◂ ▸ 70	◂ ▸ 10
13:55	336	◂ ▸ 100	100,8	◂ ▸ 70	◂ ▸ 10

Fig. 4 The user interface occupancy of the flights and to determine the auditees and auditors

situation as close as possible to reality, it was important to make use of reality-based timetables, which saw flight times and flight-specific maximum passenger capacity. According to the statistics from Finavia [13], passenger numbers will grow strongly, especially in May. In the simulation model, we wanted to note the effects of the increase in the number of passengers in the summer timetable, so the preliminary data on the busiest day of May that was May 30, 2018, see Fig. 1 was used as basic of simulation model.

Table 3 illustrates the results of the simulation with the current total time control (169.70 s) versus in case of automatically changing the passenger data (64.01 s). The simulation showed that with the selected parameters, the number of inspectors

Table 3 Simulation result of automatic transmission of data on the flow rates of passengers



ARR	Number Inpectors (169,70)	Number Inpectors (64,01)	Exchange time min	ARR	Number Inpectors (169,70)	Number Inpectors (64,01)	Exchange time min
0:25	10	4	0:27	14:40	49	17	0:29
0:30	23	9	0:24	14:40	50	18	0:26
0:55	24	8	0:30	14:45	50	20	0:26
0:55	23	10	0:28	14:50	55	20	0:28
1:40	17	7	0:25	14:50	54	20	0:26
5:45	23	9	0:25	14:50	42	20	0:23
6:25	23	9	0:25	15:30	33	20	0:32
6:55	13	7	0:30	15:30	39	20	0:30
6:55	23	8	0:26	15:35	45	20	0:30
7:00	23	9	0:27	15:35	60	20	0:25
7:00	29	9	0:24	15:20	60	20	0:26
7:10	28	9	0:24	15:20	56	20	0:22
7:55	20	8	0:25	15:25	52	16	0:24
8:50	24	9	0:24	15:50	4	5	0:33
10:20	7	3	0:25	15:50	22	11	0:23
10:50	23	9	0:24	16:20	14	5	0:25
11:40	11	4	0:25	17:40	17	7	0:25
11:55	23	9	0:24	18:05	14	6	0:24
12:20	17	7	0:25	20:05	13	5	0:26
12:35	11	5	0:24	20:25	8	3	0:27
13:10	8	3	0:27	21:00	17	7	0:25
13:30	8	3	0:29	22:05	12	5	0:24
13:45	24	10	0:29	22:15	8	3	0:26
13:50	40	14	0:27	22:30	8	3	0:27
13:55	40	14	0:24	23:00	18	8	0:25
14:10	35	10	0:27	23:05	18	6	0:24
14:10	30	15	0:23	23:25	15	6	0:24
14:15	24	10	0:25	23:35	23	9	0:29
14:20	35	15	0:24	23:35	36	11	0:24
14:25	37	16	0:24	23:40	37	12	0:23
14:30	48	17	0:24	23:55	12	6	0:26

Table 4 Effects of the use of passenger information to the inspection time

	Present (s)	Automated (s) (change %)
In-depth arrival inspection – Phase 4—Departure (API), conveyance, identifier (API) – Phase 6—Destination (API), travel plan (PNR)	169.70	64.01 (−62.28)
In-depth boarding inspection – Phase 5—Destination (API), conveyance, identifier (API)	43.16	41.05 (−4.89)

decreased considerably if the data already collected by the air carrier was automatically transferred to the simulation between the information systems. The time spent on border checks may also be increased within the limits permitted by the airport's border inspection capability, and then the number of border inspectors may be a variable factor.

Table 4 shows the effects of the usage of advanced passengers information (API) to the inspection time. We see that result is significant.

Conclusions and Future

The simulation showed that the current border control total time (169.70 s) cannot achieve the current maximum time set for inspection, i.e., the waiting time for a border check can be 10 min (Table 3). The issue was solved in this simulation model so that the sum of the waiting time and the inspection time was a total of 10 min. It depicted the time taken to inspect one machine for passengers so that even the last one in the queue does not have to wait longer than the target time.

The key result of the simulation was that automatic exchange of passenger data can be used to carry out the necessary flight checks with the current border inspection capacity so that the effectiveness objective is met. In exit checks, benefit was significant. The effects of the utilization of passenger data on the inspection times are presented in Table 4.

The result of the study was that automatic procedures for handling current passenger data should be developed in the future so that border inspectors will be able to access the necessary information electronically. The question is mainly about the new technical implementation and the new sharing of information among the different authors. Passenger data sharing speeds up the required checks to a significant extent in connection with the border-crossing process. The impact was particularly evident in a thorough entry check, which only covered one-third of the current time.

The simulation model built during the research gave grounds for evaluating performance, explaining the flow efficiency of the process, and its impact from the point of view of border authorities: how to allocate resources and how to use them to achieve the overall quality of services sought. Performance was measured in order to see what kind of results was achieved with current activity. The most important elements of the simulation were the waiting time and the time of inspection. It was critical to keep the times within the allowed limits. When attention was paid to the most important variables in the course of the process, it was possible to identify the factors in which the process can be enhanced. The simulation model used was particularly suitable for modeling incoming air traffic. In the simulation model, the percentage of EU/EEA/CH nationals on the external borders was based on the calculated averages of total number of passengers. With a view to monitoring the impacts of a credible growth, the relationship between third-country nationals of each flight should be monitored separately. This is also possible if we want to extend the simulation. This study was based on a simplified model that provides sufficient initial data for the examination of the border inspection process for third-country nationals with selected variables.

References

1. Sundarapandian V.: Probability, Statistics and Queuing Theory. PHI Learning Pvt. Ltd. (2009)
2. Mattila P., Saastamoinen K., Rissanen, A.: Description of a Modern Digitalized and Co-operated Border Control Environment at Airport, Sent ISMS Conference (2018)
3. Finnair Strategy. https://investors.finnair.com/en/finnair-as-an-investment/strategy. Accessed 27 Jun 2018
4. Finavia Annual Report (2017). https://www.finavia.fi/en/newsroom/2018/five-reasons-behind-record-breaking-passenger-volumes-finnish-airports. Accessed 27 Jun 2018
5. Haapasaari, V.: Helsinki–Vantaan Kehitysohjelman Eteneminen Kohti Entistä Parempaa Lentoasemaa, Presentation. http://docplayer.fi/816453-Helsinki-vantaan-kehitysohjelman-eteneminen-kohti-entista-parempaa-lentoasemaa.html. Accessed 27 Jun 2018
6. Internet article, Lentoposti. http://www.lentoposti.fi/uutiset/helsinki_vantaan_lentoaseman_rajatarkastuslaajennus_k_ynnistyy_rajavartiolaitos_saa_lis_tiloja. Accessed 27 Jun 2018
7. RATTI 2018 Boarder Guard internal information system, non-public report
8. Border Guard Resource Plan 2018 and Action Plan and Economic Plan 2019–2022. Accessed 18 Jan 2017
9. Boarder Guard (Finland) Internal filesystem, Document: Rajatarkastushenkilöstötarpeet osana Helsinki–Vantaan lentoaseman rajanylityspaikan kehittämistä, RVLDno/2016/1262 (2016)
10. Council Regulation (EC) No. 2252/2004. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32004R2252. Accessed 27 Jun 2018
11. Finnish Law 579/2005 Act on the Processing of Personal Data by the Border Guard. https://www.finlex.fi/en/laki/kaannokset/2005/en20050579. Accessed 27 Jun 2018
12. Finavia. Finavia invests in developing Helsinki Airport (2017). https://www.finavia.fi/en/about-finavia/development-airports/invests-in-helsinki. Accessed 27 Jun 2018
13. Finavia. Statistics (2017). https://www.finavia.fi/en/about-finavia/about-air-traffic/traffic-statistics/traffic-statistics-year. Accessed 27 Jun 2018
14. Ministry of Internal Affairs (Finland) Publication 15/2017. http://julkaisut.valtioneuvosto.fi/handle/10024/80782. Accessed 27 Jun 2018
15. European Commission, On establishing a framework for interoperability between EU information systems. http://ec.europa.eu/transparency/regdoc/rep/1/2017/EN/COM-2017-794-F1-EN-MAIN-PART-1.PDF. Accessed 27 Jun 2018

16. Directive (EU) 2016/681 of the European Parliament and of the Council of 27 April 2016 on the use of passenger name record (PNR) data for the prevention, detection, investigation and prosecution of terrorist offences and serious crime. https://publications.europa.eu/en/publication-detail/-/publication/2ba036c2-11bd-11e6-ba9a-01aa75ed71a1/language-en. Accessed 27 Jun 2018
17. Regulation (EU) 2016/399 of the European Parliament and of the Council of 9 March 2016 on a Union Code on the rules governing the movement of persons across borders (Schengen Borders Code). https://publications.europa.eu/en/publication-detail/-/publication/42fba6c3-f0c5-11e5-8529-01aa75ed71a1/language-en. Accessed 29 Jun 2018

Comparing Time Series Prediction Approaches for Telecom Analysis

André Pinho, Rogério Costa, Helena Silva and Pedro Furtado

Abstract Prediction of consumption has several applications in industry, including to support strategic decisions, market offering, and value proposition. In telecommunications industry, it can also be used in network resources management and in guaranteeing quality of service to users. But in order to make good predictions, one should choose the algorithm that is best fitted to the considered time series and also configures the parameters correctly. In this chapter, we discuss the use of time series forecasting algorithms over telecommunications data. We evaluate the use of Auto-Regressive Integrated Moving Average (ARIMA), Prophet (launched by Facebook in 2017), and two neural network algorithms: Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM). We ran those algorithms over real data about Internet data consumption and mobile phone cards recharges, in order to forecast time periods of distinct sizes. Forecasted values were qualified in terms of Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Obtained results show that ARIMA is the algorithm that is best suited to most cases.

Keywords ARIMA · Forecasting · Neural networks · Prophet

A. Pinho · R. Costa · P. Furtado (✉)
University of Coimbra, Coimbra, Portugal
e-mail: pnf@dei.uc.pt

A. Pinho
e-mail: apinho@student.dei.uc.pt

R. Costa
e-mail: rogcosta@dei.uc.pt

H. Silva
Altice Labs, Aveiro, Portugal
e-mail: lena@alticelabs.com

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_23

Introduction

The emergence of new services made available through the Internet has further increased competitiveness in the telecommunications industry in the last decade. Companies had to adapt themselves to new challenges and to innovate. Ensuring good quality of service at competitive costs has become essential. On the other hand, an increasing amount of data about customers and services' usage became available and can be used in decision support tools. Also, data science techniques can be used in telecom to forecast consumption and tendencies, having great potential to support business and market decisions, and also to enable efficient management of network resources in order to guarantee high levels of quality of service. Seasonality, trends, and the variations of the number of clients are just some of the factors that influence consumption of telecom services, posting additional challenges to the consumption forecasting.

In this work, we study the use of time series forecasting methods over telecom data. We evaluate the use of Auto-Regressive Integrated Moving Average (ARIMA), Prophet (launched by Facebook in 2017), and two neural network algorithms: Multilayer Perceptron (MLP) and Long Short-Term Memory (LSTM). Although such methods may be implemented using existing time series libraries (e.g., like statsmodels [1], fbprophet [2], scikit-learn [3], and keras [4], for python implementations), they are subject to some parametrization that may greatly influence prediction quality. We ran each of those methods over two real telecom datasets (of mobile phone cards recharges and of Internet data consumption) using several configurations in order to choose the one that leads to the best prediction. We use the Root Mean Squared Error (RMSE) and the Mean Absolute Percentage Error (MAPE) to measure forecasting quality.

In the next section, we present the state of the art. Then, in section "Application of Time Series Forecasting Algorithms", we discuss the application of time series forecasting in telecommunications. In section "Experimental Analysis", we present and analyze experimental results. Section "Conclusion and Future Work" concludes and presents future work.

Related Work

Time series forecasting has been studied for several years, but there is no single algorithm that provides accurate predictions for any data series. In fact, several factors like seasonality, trends, and the amount of historical data available to train a model can affect prediction accuracy and should be considered when selecting algorithm to be used. In this section, we review some work on time series forecasting, mainly in the context of telecommunications.

Branco and Sampaio [5] studied short-term prediction of the consumption of a telecommunications service. In that work, authors evaluated the use of two neural network algorithms, Multilayer Perceptron (MLP) and the Radial Basis Function network (RBF). They argue that the use of neural networks provides good results when data has a nonlinear behavior and that such algorithms would adapt well to telecom demand behavior. In the study, per-month datasets of 3, 4, and 6 months were used to forecast 1 month of demand (*one-step-ahead* prediction). Prediction accuracy was evaluated using Mean Squared Error (MSE) and MAPE. The MLP algorithm obtained its most accurate prediction (MSE of $3.91e + 5$) when using the largest history set (containing 6-month historical data). Authors conclude that the MLP algorithm provided more accurate predictions than RBF, although having worse computational performance. We selected MLP as one of the models to test as part of this work.

Other types of neural networks, like recurrent neural networks (RNN), have also been used in forecasting. For instance, Long Short-Term Memory (LSTM) networks were used to predict rain precipitation [6] and to detect anomalies in time series [7]. LSTM can learn long-term correlation in sequences and eliminates the requirement of prespecified time windows [7].

The Auto-Regressive Integrated Moving Average (ARIMA) was used by Wang et al. [8] to predict income forecasting in telecommunications industry. Authors argue that income prediction is affected by a variety of economic and social rules which are difficult to get and that ARIMA can make good predictions from historical data about incomes even without explicitly considering those rules. Two years of monthly aggregated data was used to fit the model, while 6 months of data was used for model validation. A double differential transformation was applied to the time series in order to make it stationary and enable the use of ARIMA. Authors used the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) to identify the possible values of parameters p and q of ARIMA. Then, a set of models was evaluated by varying ARIMA parameterization. Authors conclude that ARIMA can lead to reasonable predictions in terms of telecom income.

In a recent work, Hayashi [9] compared the prediction accuracies of ARIMA and Prophet [2]. Used dataset consists of 3 years of data about US airline flights aggregated in months. The data about the first 2 years was used to train the model while the data about the third was used to test the model. Initially, it seems that Prophet would provide better results than ARIMA. But after a fine-tuning of ARIMA parameters, such method provided the most accurate results. Thus, the author concluded that ARIMA needs manual parameter tuning in order to provide good predictions. Hence, [8, 9] do show that ARIMA can provide accurate time series forecasting in telecom and other contexts, but has the big drawback that it requires fine-tuning with manual parameter configuration.

Trying to overcome such drawback, we presented and evaluated in [10] a strategy to automate the configuration selection of ARIMA. In such work, the proposed model does an exhaustive search over parameter configuration in a reasonable time, enabling its use by decision support tools.

Furthermore, in our work, it was important to evaluate the alternatives in the context of real telecom consumption data, and then we selected to evaluate ARIMA, Prophet, MLP, and LSTM accuracy in such context.

Application of Time Series Forecasting Algorithms

Time series forecasting methods allow managers to predict the future behavior of key business data from historical data series. These methods are capable of dealing with the existence of tendency, of seasonality (variations occurring in certain periods), and are time dependent.

In the following, we present Auto-Regressive Integrated Moving Average (ARIMA) [11, 12] method. Then, we detail Prophet [2], which was launched by Facebook in 2017 to allow its use by people with less knowledge in the field. The final section briefly describes the use of LSTM and MLP in time series forecasting.

Stationarity and Time Series Forecasting in ARIMA

An important concept in the application of the ARIMA time series method is stationarity, since the model can only be constructed with stationary time series. A series is stationary if its statistical properties remain constant over time. The existence of trend and seasonality are two of the reasons that lead the series to be nonstationary [8, 12].

There are two methods that allow you to check whether a series is stationary or not. The first method consists of the graphical visualization of the variation of the statistical properties of the series, such as the moving average (calculation at each instant of the average of the values corresponding to the last seasonal period, typically of 12 consecutive months) and the moving standard deviation over time. If the properties of the series do not change over time, then the series is stationary. The second method, the Dickey–Fuller test, assumes that the null hypothesis is that the series is nonstationary. This test calculates the value of the statistical test and some critical values for different levels of confidence. If the value of the statistical test is less than the critical value, then the series is stationary [11].

Differentiation [11, 12] is one of the existing techniques that allows us to deal with seasonality and trend of the time series, bringing it closer to stationarity in time. At each instant in the series, differentiation subtracts the original observation, $\mathrm{Y_t}$, from that of the previous instant, $\mathrm{Y_{t-1}}$, using the following formula:

$$\mathrm{Y'_t} = \mathrm{Y_t} - \mathrm{Y_{t-1}} \tag{1}$$

Let us now describe the Auto-Regressive (AR) and Moving Average (MA) models, before describing the Auto-Regressive Integrated Moving Average (ARIMA) method.

1. The AR model [12] extracts the influence of the values of the previous periods from those of the current period. This model is developed using the following linear equation:

$$Y_t = \mathrm{c} + \varphi_1 \cdot Y_{t-1} + \ldots + \varphi_p \cdot Y_{t-p} + e_t$$

The parameter *p* indicates the order of AR in the model and represents the delayed time period of the dependent variable. The remaining parameters of the equation include φ which represents the AR coefficient, *y*, which is the observed value, *e* the deviation of the series at the current instant, and *c*, which is a constant [12].

2. The MA model [12] extracts the influence of the error terms from the previous period in the current period. This model is developed by the following linear equation:

$$Y_t = \mathrm{c} + e_t + \theta_1 \cdot e_{t-1} - \ldots - \theta_q \cdot e_{t-q}$$

The parameter *q* indicates the MA order in the model and represents the delayed forecast errors. The remaining parameters of the equation, θ represents the MA coefficient, *y* represents the observed value, *e* represents the deviation of the series at the current instant, and *c* is a constant [12].

3. The nonseasonal ARIMA model [11, 12] consists of three components, AR, Integrated (I), and MA, each component represented by a positive integer parameter, *p*, *d,* and *q,* respectively. These three components are combined in the following linear equation:

$$Y_t = c + \varphi_1 \cdot Y_{d\,t-1} + \ldots + \varphi_p \cdot Y_{d\,t-p} + e_t + \theta_1 \cdot e_{t-1} - \ldots - \theta_q \cdot e_{t-q}$$

The three components have the following characteristics:

- Component I [11]: This component is represented by parameter d of the model. It indicates the number of times the series has been differentiated to approximate stationary in time. Typically, one order for a nonstationary series and zero order for a series stationary in time. This component is developed by the differentiation formula presented before.
- AR component [6, 11]: represented by the p parameter of the model. This parameter indicates the order of the AR component and represents the delayed time period. Its value is estimated by the Autocorrelation Function (ACF), a function that measures the correlation between the original series and its delayed version of *p* time periods.
- MA component [6, 11]: represented by the q parameter of the model. This parameter indicates the order of the MA component and represents the delayed forecast errors. Its value is estimated by the Partial Autocorrelation Function (PACF), a function that measures the partial correlation between the original series and its backward version.

It should be noted that parameters *p* and *q* are determined when the respective functions, ACF and PACF, cross the upper confidence interval for the first time. The confidence interval of the two functions is calculated as $\pm 1.96/\sqrt{n}$, where the variable *n* corresponds to the size of the historical data [11, 12].

4. The seasonal ARIMA model [12] extends the previous model, combining its components along with the seasonal model. The seasonality component is also represented by a parameter and indicates the period of seasonality of the series.

Time Series Forecasting with Prophet

Time series forecasting with Prophet is more automated due to its ability to find automatically inflection points in the data originated by changes in trend. A novelty of this method in relation to the previous one is the possibility of accommodating the existence of seasonal festive periods. The method combines three components, the trend, the seasonality, and the festive periods, each modeled by some function [13]:

$$\mathrm{y(t)} = \mathrm{g(t)} + \mathrm{s(t)} + \mathrm{h(t)} + \epsilon \mathrm{t}$$

The trend component, g(t), is modeled by a logistic function. The seasonality component, s(t), by a Fourier series. The festive periods, h(t), are adjusted by parameterization in the model. Finally, the error term, ϵt, represents the changes originated by circumstances that are not accommodated by the model [10]. Further information on the formulation details of each of these components can be found in [10].

Time Series Forecasting with Neural Networks

The neural networks are constituted by a set of units connected to each other.

Multilayer Perceptron (MLP) [14] is a feedforward artificial neural network consisting of an input layer, one or more hidden layers, and an output layer. Each layer consists of units called neurons. The inputs of the network correspond to the records of the training data. These pass through the input layer to compute a weighted sum, with adjustable weights per input, to be inserted into the neurons that make up the second layer, called the hidden layer. The outputs of the neurons from the hidden layer can be inserted into another hidden layer based on a successive weighted sum and so on. The weighted outputs of the last hidden layer are inserted into the neurons that make up the output layer capable of predicting the desired quantity for a given month. Each neuron of the hidden and output layers receives as input the weighted sum of the outputs of the neurons from the previous layer. Then, it applies an activation function to the weighted input to calculate its output.

Long Short-Term Memory (LSTM) [15] is a type of Recurrent Neural Network (RNN) capable of learning long-term dependencies. The architecture of the network consists of a set of units called blocks, containing memory cells connected to each other, and gates. The memory cells are responsible for storing the temporal state of the network. Gates are responsible for controlling the flow of information, with each block containing an input, output, and forget gate. The input gate controls the flow of input activations in the memory cell. The output gate controls the output flow of the memory cell activations to the rest of the network. The forget gate decides the internal state of the cell, resetting or forgetting the memory of the cells.

Experimental Analysis

We made several experiments using real-world data in order to compare the accuracy of ARIMA, Prophet, MLP, and LSTM in time series forecasting over telecom data.

Used datasets were provided by a telecom operator and consist on real data about mobile phone card recharges (recharge) and consumption of Internet data (consumption) that was aggregated by month for a time period of 4 years (2014–2017), as it is detailed latter in this section.

For each dataset and selected algorithm, we used data about 2014–2016 to fit the model and made several experiments testing distinct configurations in order to forecast the values for the full year of 2017. We measured MAPE and RMSE in order to select the most accurate configuration for each algorithm.

Then, we used the configuration that leads to the best result of each method and dataset to forecast values for each of 2017s quarters. In these tests, we used a period of 36 months to fit the considered model and used the trained model in order to forecast values for the following 3 months.

Testbed Datasets

The data used in our experiments comes from a medium-sized telecommunication operator. We use data about mobile phone card recharges (i.e., loading the phone account with a certain balance to use in calls and data) and about the volume of Internet data consumed over 4 years (aggregated by month). Figure 1 graphically shows the normalized consumption data variation over the period from 2014 to 2017.

In Fig. 1, we can see that there is a tendency of consumption decreasing in terms of recharging and increasing in terms of data consumption. Seasonality can be seen as the temporarily accentuated increase and decrease of consumption in certain months of each year. In the series of recharges, there is a lower consumption in the months of February, October, and November, and a higher consumption in the months of January, August, and December. Data from the first 3 years was used for analysis of the time series and training of the model, the fourth year for testing the model. The data series in general presents a more accentuated growth in the last months of each year.

Figure 2 shows the stationarity test of the time series of recharges with two methods. The figure above shows the graphical visualization of the statistical properties of the series over time, moving average and moving standard deviation. Both were

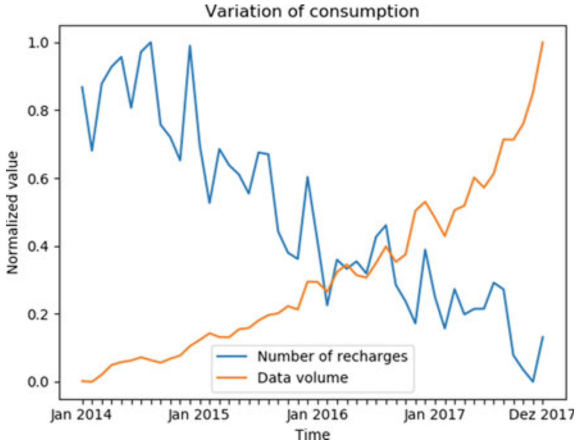
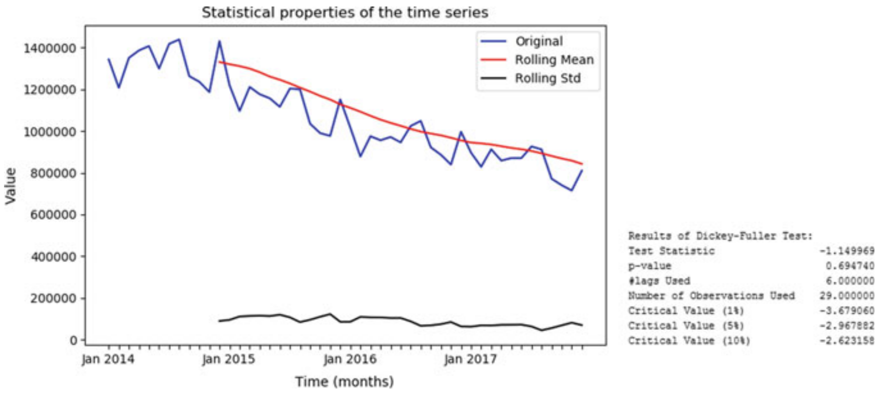


Fig. 1 Mobile phone card recharges and Internet data consumption variation



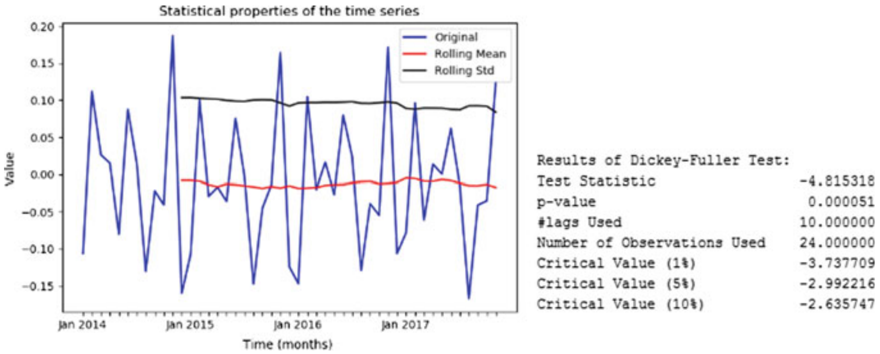
```
Results of Dickey-Fuller Test:
Test Statistic                    -1.149969
p-value                            0.694740
#lags Used                         6.000000
Number of Observations Used       29.000000
Critical Value (1%)               -3.679060
Critical Value (5%)               -2.967882
Critical Value (10%)              -2.623158
```

Fig. 2 Stationarity test of the original time series

configured with the value of 12, corresponding to the seasonal period. The second method is based on the Dickey–Fuller statistical test, with results presented next to the figure.

From the analysis of the previous graph, we can see a decrease of the average over time. This variation over time indicates that the series is nonstationary. The result of the statistical test has a value greater than the critical value with a confidence level of 95% ($-1.149969 > -2.967882$), also indicating that the series is nonstationary in time. Then a first-order differentiation transformation was applied to make the time series stationary in time. Figure 3 presents the stationarity test of the series after this transformation.

From the analysis of the previous graph, it is verified that the statistical properties of the time series have become approximately constant over time. The



```
Results of Dickey-Fuller Test:
Test Statistic                 -4.815318
p-value                         0.000051
#lags Used                     10.000000
Number of Observations Used    24.000000
Critical Value (1%)            -3.737709
Critical Value (5%)            -2.992216
Critical Value (10%)           -2.635747
```

Fig. 3 Stationarity test of the transformed time series

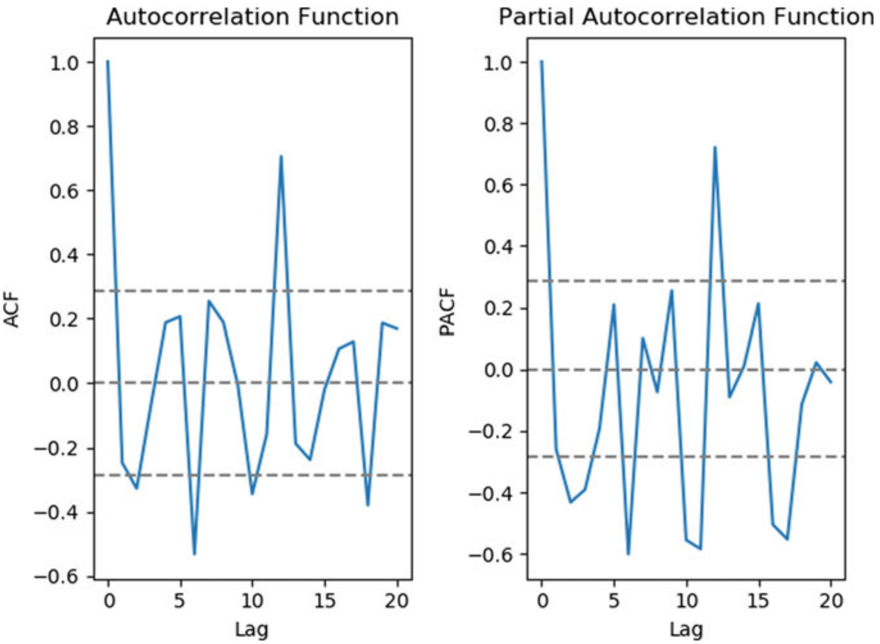


Fig. 4 Graphs of functions ACF and PACF of the temporal series

result of the statistical test is also less than the critical value for a 95% confidence level ($-4.815318 < -2.992216$), also indicating that the time series approached stationarity.

The number of differentiations that allowed the series to become approximately stationary indicates the value of the parameter d of ARIMA. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) can be used to determine the values of ARIMA' parameters p (order of the AR component of the model)

Table 1 ARIMA 1-step versus ARIMA 12-step on recharges—1 year

Method	RMSE	MAPE
ARIMA 1-step	2.69e + 4	2.6
ARIMA 12-step	3.99e + 4	4.2

and q (order of the MA component of the model). Figure 4 shows the graphs of these two functions.

The graph on the left side, representing the ACF function, allows estimation of the value of parameter p, by looking at the position where the function crosses the confidence interval for the first time. In the graph, this happens between $p = 0$ and $p = 1$. The other graph on the right side is the PACF function, which allows to obtain the value of parameter q, also by crossing the function with the confidence interval. In the graph, it happens between $q = 0$ and $q = 1$.

Dynamic Forecasting in ARIMA

The Python library used to run ARIMA supports the use of both *one-step-ahead* forecasting and of dynamic forecasting. In dynamic forecasting, real values are considered up to a certain point and then several values may be forecasted. In *one-step-ahead* forecasting, real values up to a certain point are used to forecast the next value in the time series. Then, the real value for the recently forecasted point is used to forecast the next point, and so on.

One-step-ahead prediction reduces forecasting errors, as shown in Table 1, where we present RSME and MAPE values when using *one-step-ahead* configuration (ARIMA 1-step) or dynamic configuration (ARIMA 12-step) to forecast recharges. In both situations, we used (0, 1, 1) as the values for the parameters (p, d, q) of ARIMA, and 12 as the number of steps for a single seasonal period, as such values were the ones that gave the best results.

The following graphs in Fig. 5 present the values obtained when using both configurations and confirm that using one-step-ahead would lead to better results than dynamic forecasting.

Although the one-step-ahead configuration leads to better results than the dynamic one, one-step-ahead prediction is not suitable when planning resource allocation for several months. Hence, in the following sections, we compare the use of Prophet, MLP, and LSTM with the dynamic prediction configuration of ARIMA.

Forecasting Mobile Phone Card Recharges

Table 2 shows the RMSE and MAPE obtained using ARIMA, Prophet, MLP, and LSTM to forecast a full year of recharges. Those results were obtained using the

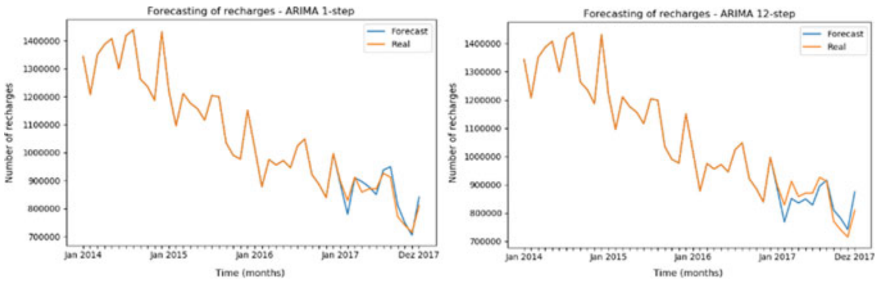


Fig. 5 Recharges forecasting with ARIMA

Table 2 Forecasting recharges—1 year

Method	RMSE	MAPE
ARIMA 12-step	3.99e + 4	4.2
Prophet	4.26e + 4	3.7
MLP	4.20e + 4	4.4
LSTM	6.25e + 4	5.5

dynamic configuration of ARIMA described in the last section. In ARIMA and Prophet, we used a seasonality period of 12 months and in MLP and LSTM we used a validation set size of 33% of the training data.

In the above table, we can see that ARIMA had the lowest value of RMSE while the lowest MAPE is provided by Prophet. The worst results were obtained when using LSTM.

The graphs in Fig. 6 present the real and forecasted values for each considered method. By a visual inspection of such graphs, it is possible to identify that Prophet made a considerably bad estimation of the last value of the time series. Such error has impacted the obtained RMSE value.

We also made several experiments on estimating recharges for periods of 3 months. In each experiment, we used a dataset of 36 months of real values to fit the model and estimated the following 3 months in the time series, each estimated period corresponding to one of 2017s quarters.

In Table 3, we present average and maximum values for RMSE and MAPE for each forecasting method. Again, ARIMA and Prophet generated better results than the other methods.

By analyzing obtained predictions, we verify that Prophet made another bad estimation for the last month of the year, while ARIMA went worse in the second quarter. Obtained results for such quarters (Q2 and Q4) are represented in Fig. 7.

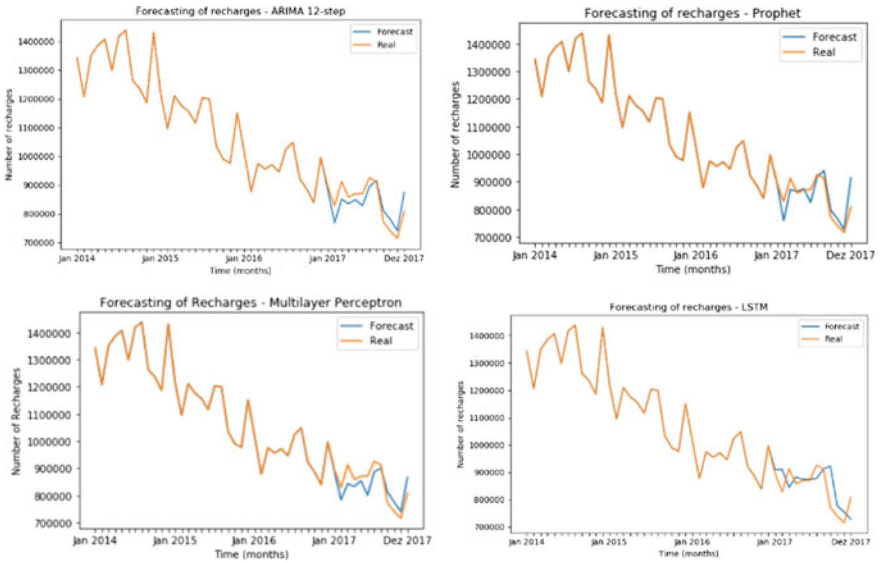


Fig. 6 Recharges forecasting for a period of 12 months

Table 3 RMSE and MAPE for recharge prediction—quarters

Method	RMSE		MAPE	
	Mean	Maximum	Mean	Maximum
ARIMA 3-step	2.96e + 4	4.97e + 4	3.7	5.0
Prophet	4.21e + 4	1.27e + 4	4.1	5.2
MLP	5.88e + 4	1.04e + 5	6.6	13.1
LSTM	5.51e + 4	9.34e + 4	5.4	7.8

Forecasting Internet Data Consumption

We also evaluated the use of ARIMA, Prophet, MLP, and LSTM to forecast data consumption over 2017. First, we made several experiments forecasting 12 months of consumption and looking for the configuration parameters that gave the best result in each method. In ARIMA, the best results were obtained while using (0, 0, 0) for (p, d, q) and used (0, 1, 0) for (P, D, Q) of the order of the seasonal component. All the other configurations remain the same.

Figure 8 shows real and forecasted data for each algorithm. ARIMA led to the best results among all tested algorithms, but in this data series LSTM produced better estimations than MLP and Prophet. That is also presented in Table 4 that contains calculated RMSE and MAPE for each algorithm.

We also evaluated forecasting data consumption in periods of 3 months. Table 5 shows the mean and maximum values of RMSE and MAPE in this configuration.

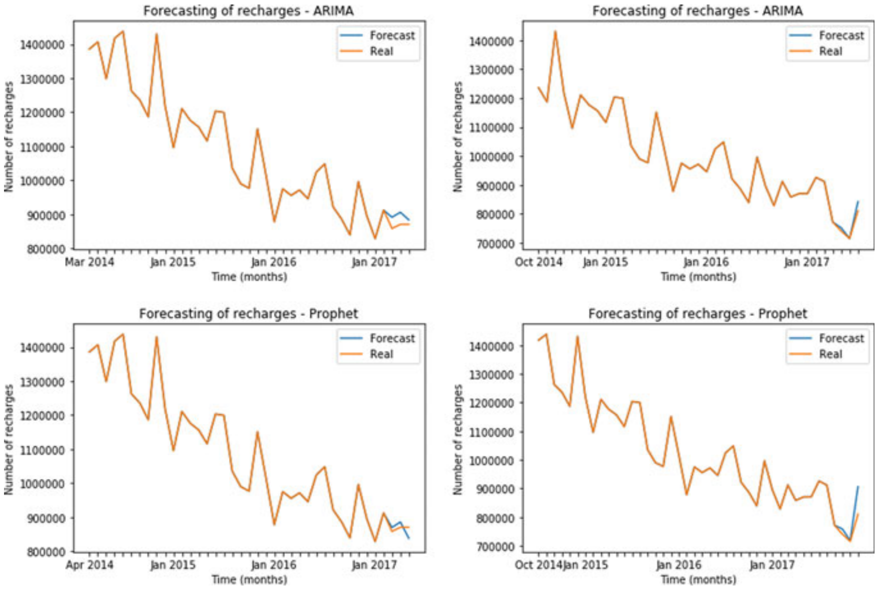


Fig. 7 Forecasting recharges in quarters—Q2 and Q4

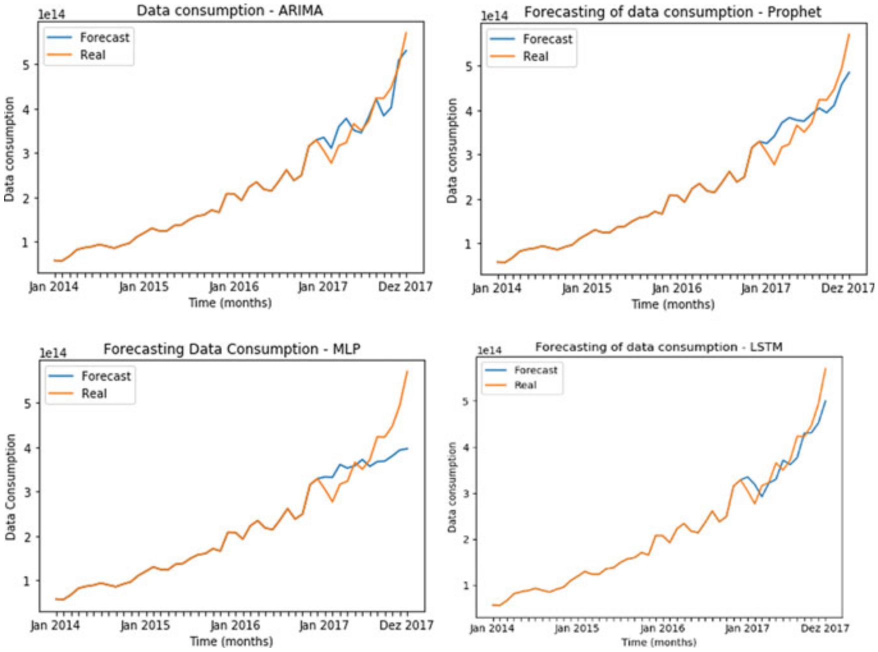


Fig. 8 Data consumption forecasting for a period of 12 months

Table 4 Forecasting data consumption—1 year

Method	RMSE	MAPE
ARIMA 12-step	3.23e + 13	7.1
Prophet	4.41e + 13	9.9
MLP	6.98e + 13	14.0
LSTM	3.44e + 13	7.4

Table 5 Forecasting data consumption—quarters

Method	RMSE		MAPE	
	Mean	Maximum	Mean	Maximum
ARIMA 3-step	3.06e + 13	6.38e + 13	9.96	11.87
Prophet	3.59e + 13	6.21e + 13	7.84	11.12
MLP	3.61e + 13	4.58e + 13	8.22	14.28
LSTM	3.33e + 13	4.81e + 13	7.42	10.72

Forecasting data consumption was more challenging than forecasting recharges, as it can be proved by comparing the higher error values in Tables 3 and 5. ARIMA produced the lowest mean value for RMSE while LSTM gave the lowest mean average value of MAPE.

In all considered situations, the dynamic configuration of ARIMA provided the best results for RMSE. ARIMA also produced the best results in terms of MAPE for the quarters estimation of the recharge series and for the full year estimation of the consumption data series.

Conclusion and Future Work

In this work, we studied the application of four algorithms in forecasting telecom data. We discussed parameterization issues and experimentally tested the use of several parameters to obtain the best result of each method. The quality of obtained predictions was validated in terms of RMSE and MAPE, and could also be checked by visual inspection.

Although Prophet and LSTM provided some good results in some specific cases, the ARIMA model consistently provided good predictions for either mobile phone card recharges or Internet data consumption, even when tested for periods of distinct sizes. Despite the great variation of phone recharges and data consumption over the years (and over a single year, due to the existence of seasonality), we could obtain some reasonably good results, with MAPE values of about 4%. Although ARIMA requires a great deal of knowledge in its parameterization, an automatic parameterization procedure can be used, which enables the model to be integrated into forecasting tools used by managers.

For future work, we intend to test obtained models over other types of telecom data, including SMS and phone calls, evaluate the effects of seasonality over such datasets, and compare the results with the results obtained in this work. Furthermore, we intend to identify alternatives and configurations to be integrated into decision support tools used in telecom industry.

References

1. StatsModels Statistics in Python. http://www.statsmodels.org/devel/generated/statsmodels.tsa.arima_model.ARIMA.html. Accessed 20 Mar 2018
2. Prophet Forecasting at scale. https://facebook.github.io/prophet/. Accessed 16 Apr 2018
3. Neural network models (supervised). https://scikit-learn.org/stable/modules/neural_networks_supervised.html. Accessed 23 Nov 2018
4. Keras: The Python Deep Learning library. https://keras.io/layers/recurrent/. Accessed 24 Nov 2018
5. Branco, S.T., De Sampaio, R.J.B.: A new artificial neural networks forecast model in telecommunications. In: Computing and Computational Techniques in Sciences, pp. 70–77. WSEAS Press (2008)
6. Xingjian, S., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, pp. 802–810 (2015)
7. Malhotra, P., Vig, L., Shroff, G., Agarwal, P.: Long short term memory networks for anomaly detection in time series. In: 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2015, pp. 89–94 (2015)
8. Wang, M., Wang, Y., Wang, X., Wei, Z.: Forecast and analyze the telecom income based on ARIMA model. Open Cybern. Syst. J. **9**(1), 2559–2564 (2015). https://doi.org/10.2174/1874110X01509012559
9. Is prophet really better than ARIMA for forecasting time series data. https://blog.exploratory.io/is-prophet-better-than-arima-for-forecasting-time-series-fa9ae08a5851. Accessed 16 May 2018
10. Pinho, A., Silva, H., Furtado, P.: Experimental comparison and tuning of time series prediction for telecom analysis. In: Proceedings of International Conference on Time Series and Forecasting, ITISE, vol. 2, pp. 586–597 (2018)
11. A comprehensive beginner's guide to create a Time Series Forecast. https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/. Accessed 15 Mar 2018
12. MATH6011: Forecasting. https://www.southampton.ac.uk/~abz1e14/papers/Forecasting.pdf. Accessed 22 Mar 2018
13. Taylor, S.J., Letham, B.: Forecasting at scale. The American Statistician (just-accepted) (2017). Accessed 16 Apr 2018
14. Han, J., Pei, J., Kamber, M.: Data Mining: Concepts and Techniques, 3rd edn. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2011)
15. Azzouni, A., Pujolle, G.: A long short-term memory recurrent neural network framework for network traffic matrix prediction (2017). arXiv preprint arXiv:1705.05690. Accessed 1 Dec 2018

Application of Load Forecasting in Thermal Unit Commitment Problems: A Pattern Similarity Approach

Guilherme Costa Silva, Adriano C. Lisboa, Douglas A. G. Vieira and Rodney R. Saldanha

Abstract This study investigates the application of short-term load forecasting (STLF), which consists of estimating a future demand within a period of time up to one week, to thermal unit commitment problems (TUCP), providing schedule for power plant operations. Both problems have fundamental importance for power system operations and good results on STLF may also influence TUCP performance. The pattern similarity approach is chosen for STLF, which allows the use of regression algorithms based on machine learning applied to time series analysis and forecasting results are used as information for generators scheduling. This study proposes a framework containing these tasks with a deep review of them and provides some statistical information regarding the performance and validation of the framework.

Keywords Pattern similarity · Short term load forecasting · Thermal unit commitment problem · Self-organizing maps · Power systems

G. C. Silva (✉) · A. C. Lisboa · R. R. Saldanha
Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais,
Av. Antônio Carlos, 6627, Belo Horizonte, MG 31270-901, Brazil
e-mail: guicosta@ufmg.br

R. R. Saldanha
e-mail: rodney@cpdee.ufmg.br

A. C. Lisboa
Gaia Solutions on Demand, Rua Professor José Vieira Mendonça, 770,
Belo Horizonte, MG 31270-901, Brazil
e-mail: adriano.lisboa@gaiasd.com

A. C. Lisboa · D. A. G. Vieira
Graduate Program in Mathematical and Computational Modeling, Centro Federal de Educação
Tecnológica de Minas Gerais, Av. Amazonas 7675, Belo Horizonte, MG 30510-000, Brazil
e-mail: douglas.vieira@enacom.com.br

D. A. G. Vieira
ENACOM Handcrafted Technologies, Alameda das Latânias, 215,
31270-800 Belo Horizonte, MG, Brazil

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_24

Introduction

The importance of load demand in power systems tasks leads to the development of several load forecasting models, which can be based on regression, statistical analysis, smoothing, or parametric analysis. Some approaches consist of applying models based on integrated auto-regressive and moving averages, namely ARIMA, exponential smoothing models of Holt-Winters [20], or principal component analysis [18]. There are also some computational intelligence models, among them the ones based on artificial neural networks, studied in [4, 14, 19].

The pattern similarity-based approach, presented in [2, 3], consists of analyzing time series historical data by training a machine learning (ML) algorithm in order to build a model and predict data. These models can be described by their simplicity, few parameters used, and can be applied with several ML approaches. In addition, deseasonality or decomposition methods are not mandatory, only the definition of patterns is performed.

This principle can be successfully applied to short-term load forecasting (STLF) models, which are useful in control, scheduling and safety tasks essential in electrical systems. Such as the Unit Commitment problem (UCP) [8], which consists of scheduling generating units status (ON/OFF) in order to supply the hourly load demand with minimum generation cost. A successful load forecasting is critically important to solve UCP, since it determine the operation of generators in order to satisfy a given demand.

This chapter presents the pattern similarity model and how STLF problems are represented in algorithms, with some examples, such as self organizing maps (SOM) neural network algorithms, as in [4] and the application of results in a thermal unit commitment problem (TUCP), which deals with the scheduling of thermal plants in order to handle the load demand.

Pattern Similarity-Based Forecasting Method

The short-term load forecasting problem predicts future demand in a range of a few hours before up to a few weeks ahead. According to [17], it is an important part of energy management system and presents complex nonlinear characteristics when different factors affect load at the same time, such as seasonal, economic and climatic factors, or even some random effects, however, it is possible to predict statistically a distinct pattern of consumption resulting from the totality of individual loads, according to [7]. In [16], three objectives for the problem are listed, such as scheduling generation, providing safer and more reliable operation of the plants, and increasing system reliability.

The pattern similarity-based forecasting method (PSBFM) is the methodology considered here for load forecasting. Most of the methodology, based on [2, 3] is reviewed.

Considering some factors such as seasonality and trends, it is possible to transcribe a simple and univariate problem based on similarity metrics, based on a simple optimization process that involves a generalization of minimal distance methods, considered as a basis of some pattern recognition and machine learning techniques, using analogies between time series fragments with seasonal cycles.

A pattern is a vector with components that are functions of actual time series elements z. The input and output (forecast) patterns are defined respectively as $x = [x_1, x_2, \ldots, x_n]$, namely x-pattern, and $y = [y_1, y_2, \ldots, y_n]$, namely y-pattern, both forming the pair (x_i, y_i), given the forecast horizon τ which is constant, the number of instants n and taking the following assumption.

Assumption. If a pattern x_a which represents a period preceding the forecasted period is similar to another pattern x_b from historical data, then the resulting forecast pattern y_a is similar to the forecast pattern y_b.

This assumption provides a nonparametric regression model that generates a y-pattern, paired with a corresponding x-pattern, resulting from the evaluation of historical data, all of them represent continuous sequences of a given time series that coincide with daily cycles, through a machine learning method.

The prediction model based on pattern similarity can be defined by:

1. mapping the time series (seasonal cycles) in paired patterns **x** and **y**;
2. selecting similar patterns **x** to the query pattern $\mathbf{x}^{\star}$;
3. defining of the pattern **y** that pairs with **x** in order to forecast $\hat{\mathbf{y}}$ related to $\mathbf{x}^{\star}$;
4. decoding $\hat{\mathbf{y}}$ in the seasonal cycle referring to $\hat{\mathbf{z}}$.

This prediction model considers the original time series z to be pre-processed in order to treat specificities in the series (periodicities, trend, variance) or to normalize the data, according to:

$$x_t = f_x(z_{i,t}, \phi_i), \tag{1}$$

$$y_t = f_y(z_{i+\tau,t}, \phi_i), \tag{2}$$

where i refers to the period analyzed before forecast, t is the hour of the day, τ is the period to be forecasted related to i and ϕ_i is the decoding variable, usually the daily average load. The x-pattern size n can be equal to 24, 48 or 96, for hourly, half-hourly or quarter-hourly load time series, respectively.

Usually, pre-processing can be defined by the following functions:

$$f_x(z_{i,t}, \phi_i) = \frac{z_{i,t} - \phi_i}{\sqrt{\sum_{j=1}^{n} (z_{i,j} - \phi_i)^2}} \tag{3}$$

$$f_y(z_{i+\tau,t}, \phi_i) = \frac{z_{i+\tau,t} - \phi_i}{\sqrt{\sum_{j=1}^{n} (z_{i,j} - \phi_i)^2}} \tag{4}$$

Note that in the construction of **y** only prior data is used, since **y** values are not known at the time of the forecast. There are some other normalization functions in [2].

During the test phase, an inverse function is employed to calculate the seasonal cycle $\hat{z}$ based on the forecasted pattern $\hat{y}$ as follows:

$$f_y^{-1}(y_{i,t}, \phi_i) = \hat{y}_{i,t}\sqrt{\sum_{j=1}^{n}(z_{i,j} - \phi_i)^2} + \phi_i \tag{5}$$

The model is evaluated through similarity functions, usually based on linear or non-linear mapping of a given distance metric D, usually the Euclidean Distance, which provides comparisons among **x** patterns. Distance metrics are applied to training phase, in order to analyze **y** patterns and the correlation between measured distances, and to test phase, in order to estimate **y** pattern from observed **x**-patterns.

With these mechanisms, several algorithms can be used in order to map $x \rightarrow y$ and then obtain new **y** patterns from presented **x** patterns. Among some methods, there are three algorithms based on Self-organizing maps (SOM) neural network that are formulated considering pattern similarity principles as follows:

Algorithm 1. Self-organizing map model 1

1. Concatenation of the paired **x** and **y** patterns in the pattern **z**;
2. train SOM network according to patterns **z**;
3. presentation of the query pattern $\mathbf{x}^\star$ and assigning it to the nearest neuron w_j according to $j^\star = \text{argmin}\|x^\star - w_{x,j}\|$, where $w_{x,j}$ is the x-pattern part of the neuron w;
4. reconstruction of the **y** pattern paired with the query pattern based on the **y** part of the neuron $w_{y,j}$.

Algorithm 2. Self-organizing map model 2

1. Independent grouping of patterns **x** and **y**, with two SOMs for each pattern;
2. estimation of the conditional probabilities $P(C_{y,l}|C_{x,k})$, where $C_{x,k}$ and $C_{y,l}$ are the respectives neurons for x-patterns and y-patterns;
3. presentation of the query pattern $\mathbf{x}^\star$ and assigning it to a set of nearest neurons $w_{x,j}$;
4. reconstruction of the **y** pattern paired with the query pattern based on neuron $w_{y,k}$ from SOM of pattern **y** and probabilities $P(C_{y,l}|C_{x,k})$.

Algorithm 3. Self-organizing map model 3

1. Only **y** patterns are developed;
2. neurons w_j are labeled according to the following information: day numbers i and day types δ;
3. calculation of the number of entries e_j which satisfy two conditions:
   (a) query pattern is from the same period of the year as the forecasted **y** pattern;
   (b) day type is the same as for the forecasted **y** pattern;

4. reconstruction of the **y** pattern paired with the query pattern based on neuron w_j and label entry numbers e_j.

Some clustering algorithms, such as k-means, can also be applied to generate these models, as proposed in [4].

These algorithms consist of examples of PSBFMs, which are tested in short-term load forecasting problems. The resulting forecast is applied to the scheduling of thermal generators, provided by TUCP solutions, as further presented.

Thermal Unit Commitment Problem

The Unit Commitment deals with generators scheduling considering a demand to be supplied. In order to achieve it, each generator status (ON or OFF) is set during a given hour. The problem formulation, specific to thermal generators, defines some costs and a set of constraints, being based on formulations defined in [1, 12, 15] and reviewed here. Since most units are based on thermal generators, we refer to this formulation as TUCP. This formulation also includes the Economic Dispatch subproblem.

The complete formulation is defined as follows:

$$\min C = \sum_{t=1}^{T}[PC_t + EC_t + SU_t + SD_t], \tag{6}$$

$$PC_t = \sum_{n=1}^{N_{Gen}} a_n G_{n,t}^2 + b_n G_{n,t} + c_n, \tag{7}$$

$$EC_t = \sum_{n=1}^{N_{Gen}} \alpha_n G_{n,t}^2 + \beta_n G_{n,t} + \gamma_n, \tag{8}$$

$$SU_t = \sum_{n=1}^{N_{Gen}} \min(0, S_{n,t} - S_{n,t-1} + 1) \times CSU_n, \tag{9}$$

$$CSU_n = \begin{cases} CSH_n & \text{if } S_{n,t} \leq CT_n \\ CSC_n & \text{otherwise} \end{cases}, \tag{10}$$

$$SD_t = \sum_{n=1}^{N_{Gen}} \max(0, S_{n,t} - S_{n,t-1} - 1) \times -CSD_n, \tag{11}$$

where:

C Total cost of unit commitment;
PC_t Total production costs at hour t, in $\$/MW$;

EC_t	Total emission costs at hour t, in $\$/ton$;
SU_t	Total unit startup cost at hour t, in \$;
SD_t	Total unit shutdown cost at hour t, in \$;
$G_{n,t}$	Active power of unit n at hour t, (MW);
a_n, b_n, c_n	Production cost coefficients of unit n;
$\alpha_n, \beta_n, \gamma_n$	Emission cost coefficients of unit n;
$S_{n,t}$	State of unit n at hour t, [0, 1];
CSU_n	Startup costs of unit n, \$;
CSH_n	Hot startup costs of unit n, \$;
CSC_n	Cold startup costs of unit n, \$;
CT_n	Cold start-up threshold for generation n, \$;
CSD_n	Shutdown costs of unit n, \$.

Generator operation are also subject to the following equality and inequality constraints:

$$\sum_{n=1}^{N_{Gen}}(G_{n,t} \times S_{n,t}) = \sum_{l=1}^{N_{Load}} P_{l,t}, \forall t, \tag{12}$$

$$G_{min,n} \leq G_{n,t} \leq G_{max,n} \iff S_{n,t}, \forall n, t, \tag{13}$$

$$-RDN_n \leq G_{n,t} - G_{n,t-1} \leq RUP_n \iff S_{n,t} - S_{n,t-1} = 0, \forall n, \tag{14}$$

$$\sum_{n=1}^{N_{Gen}}(G_{n,t} \times S_{n,t}) = \sum_{l=1}^{N_{Load}} P_{l,t} + SR_t, \forall t, \tag{15}$$

$$\sum_{h=t-T_{ON,n}}^{t-1} (S_{n,h} = T_{ON,n}) \iff S_{n,t} - S_{n,t-1} = -1, \forall n, \tag{16}$$

$$\sum_{h=t-T_{OFF,n}}^{t-1} (S_{n,h} = 0) \iff S_{n,t} - S_{n,t-1} = +1, \forall n, \tag{17}$$

where:

$P_{l,t}$	Amount of load demand at t time interval and l load point, in MW;
$G_{min,n}$	Minimum power limits of generator n, in MW;
$G_{max,n}$	Maximum power limits of generator n, in MW;
RUP_n	Ramp rate upper limit of generator n;
RDN_n	Ramp rate lower limit of generator n;
SR_t	Minimum up time of generator n;
$T_{ON,n}$	Minimum down time of generator n;
$T_{OFF,n}$	Minimum down time of generator$n$.

The TUCP can be solved by several binary optimization methods. In this case, dynamic programming methods, as studied in [8, 12, 15], can solve this problem deterministically. Before solving TUCP, load forecasting tasks should be performed in order to obtain future demand of a power system, so that generators operations can be obtained.

This framework was previously studied in [5], by finding hourly and daily forecasting through neural networks and applying the results on unit commitment schedule decisions. The novelty of this study is the introduction and application of the pattern similarity concepts and then applying them at the forecasting and schedule framework, as further presented.

Test Framework

A set of simulations is proposed as a test framework that contain the main tasks of a power system planning. There are five phases, two of them for data analysis and the others related to load forecasting, unit commitment and the overall analyses, as illustrated in Fig. 1 and described throughout this work.

ISO New England Data

Main data from the framework is derived from New England area. ISO New England dataset is available in [10, 13] and contains information regarding the 8 regions within New England area, the demand and market located marginal prices applied in TUCP approaches.

This dataset is used during the STLF phase, in which the pattern similarity-based method is applied to one of 8 zones, The training data consists of load demand

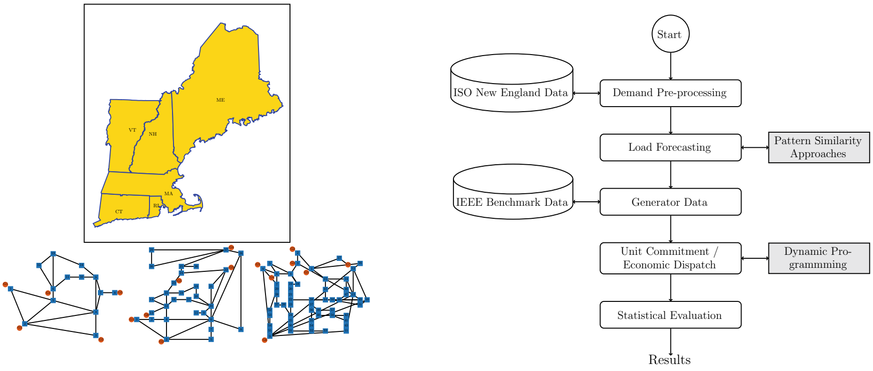


Fig. 1 Proposed framework

between 2007 and 2013, which is processed in order to forecast information regarding 2014 demand.

IEEE Benchmark Systems

Three benchmarks systems from IEEE, namely IEEE 14-bus, 30-bus and 57-bus, are tested during the unit commitment phase with their generators being scheduled according to the demand forecasted during STLF phase.

All database used for experiments are given in [11], except for the demand, which is replaced by one contained in ISO NE database, as previously presented.

Pattern Similarity SOM Network

In order to provide information regarding load demand, the three methods presented in section "Pattern Similarity-Based Forecasting Method" and some statistical analysis is provided regarding error and correlation analysis.

Forecasts quality can be measured by the root mean square error (RMSE), the mean absolute percent error (MAPE), or the mean absolute error (MAE) between the forecasted data F_t and the actual data A_t considering the period within n instants:

$$RMSE = \sqrt{\frac{\sum_{t=0}^{n}(F_t - A_t)^2}{n}} \tag{18}$$

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{|A_t - F_t|}{A_t} \tag{19}$$

$$MAE = \frac{\sum_{t=0}^{n}|F_t - A_t|}{n} \tag{20}$$

While RMSE is scale-dependent, useful when applied to the same set of data and being more sensitive to outliers [9], MAPE is widely used in short-term load forecasting problems [22], defined by sampling energy demand in days, with its hourly information as a dimension of the problem. As a scale-independent metric, can be used across different data sets, however, it may assume large values if $A_t < 1$ or become undefined if $A_t = 0$. However, with the application of inverse function $f_y^{-1}(y_{i,t})$ on forecasted pattern, this problem is minimized. The MAE, studied in [21] along with RMSE, measures the magnitude of mean error in tests. For all measurements, F_t are obtained from inverse function applied to the forecasted patterns, and A_t are obtained from original time series values $z_{i,t}$.

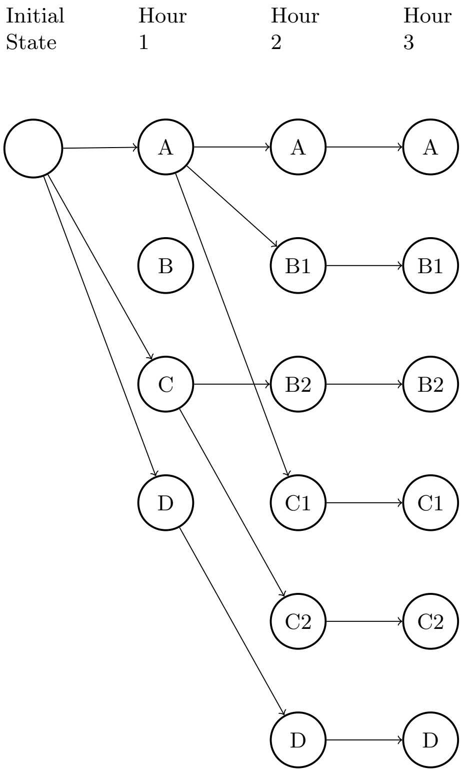


Fig. 2 Description of the dynamic programming approach used in this work, as in [8]

Dynamic Programming

The Dynamic Programming (DP) approach, proposed in [8], is applied to TUCP. This approach features a mechanism named hourly state restriction, which is employed in order to determine which secondary states are feasible, as presented in Fig. 2.

The DP application is performed considering the evaluation of all 24 h of a day, considering working days, Saturday s, Sunday s and holidays. Four scenarios are considered a given season of the year, each of them obtained considering the mean of historical hourly data defined by the described period. In addition, all costs and constraints, including spinning reserve and ramp constraints, are considered in the simulations.

Solutions Evaluation

In the last stage, TUCP solutions are evaluated according to the following hypothesis test Ω_S:

$$\varOmega_S = \mathrm{H}(S_{g,t}, S^\star_{g,t}) \tag{21}$$

i.e., there are significant differences between the obtained scheduling $S_{g,t}$ and an ideal solution $S^\star_{g,t}$, which is obtained considering the actual load demand of test data. For this purpose, the χ^2 test [6] is employed for each case, considering all day types.

Results

Load Forecasting

The STLF experiments were performed through MATLAB R2016b, running under Ubuntu 18.04 under the Intel Core i7 processor. Parameters are defined according Table 1 and some comparisons between the studied approaches, some classical methods (ARIMA and Holt-Winters) and a pattern similarity-based k-means performing forecasting procedures 1 and 2.

Load forecasting error evaluation results are presented in Table 2, according to the control area and in Fig. 3, considering the 8 zones of ISO-NE. Overall, presented

Table 1 Parameters of load forecasting methods

ID	Description	Value	Algorithm
It	Iterations	2000	All PS-based methods
Cx	x-pattern clusters/neurons	50	All PS-based methods (except SOM3)
Cy	y-pattern clusters/neurons	50	SOM2 and KM2
Cy	y-pattern neurons	150	SOM3
p	Autoregressive lag	5	ARIMA
d	Differencing degree	1	ARIMA
q	Moving average order	0	ARIMA

Table 2 Load forecasting results for Control Area zone, with the best solution in bold

Method	RMSE	MAPE	MAE
SOM1	762.56	3.56	628.28
SOM2	**695.61**	**3.47**	**587.25**
SOM3	849.63	4.25	735.27
KM1	766.95	3.70	646.88
KM2	745.94	3.75	632.71
H-W	1921.2	6.09	1650.3
ARIMA	1215.1	5.12	1008.5

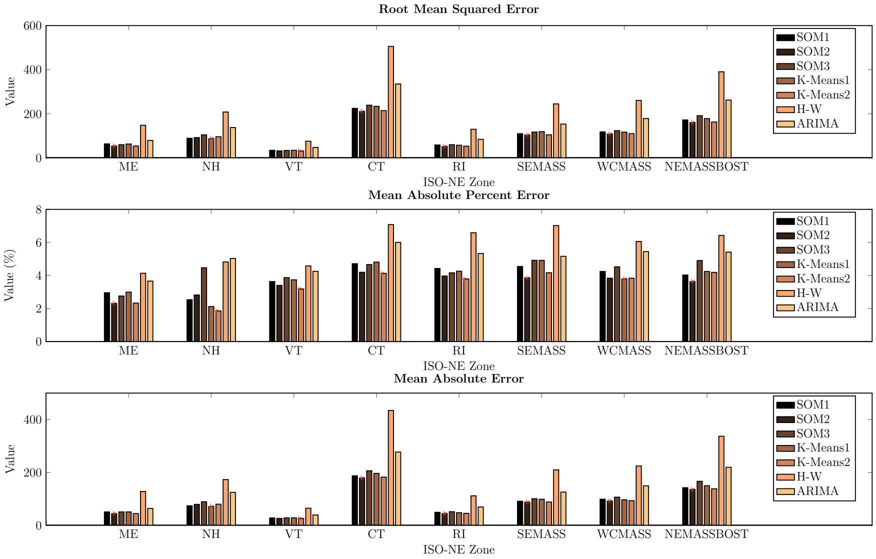


Fig. 3 Load forecasting results per zone. Bars with red stars indicate best performance in tests

results indicated satisfactory performances for most PBSFMs, mainly SOM1, SOM2 and KM1. According to error metrics, SOM model 2 has achieved best results, although MAPE metric has indicated that k-means-based models are closer to the actual data.

Even in some areas, such as the Connecticut zone, which are more difficult to predict, PBSFMs achieved reasonable resultsand may be applied to the TUCP.

Such methods, considering these results, may produce similar performance, as implied by Fig. 4, this may indicate minimal differences in the model considered for TUCP phase.

Thermal Unit Commitment

This phase consists of applying TUCP at the forecasted demand from ISO New England zones, which are partially supplied by IEEE benchmark systems as solved by DP. Some case studies are presented, according to the description of Table 3, the study of such cases aim at obtaining solutions that exploit the proper operation of all units. Results from SOM Model 2 are applied to the scheduling. Spinning reserve is 5% of the supplied demand. Four schedules are provided, according to the type of day considered.

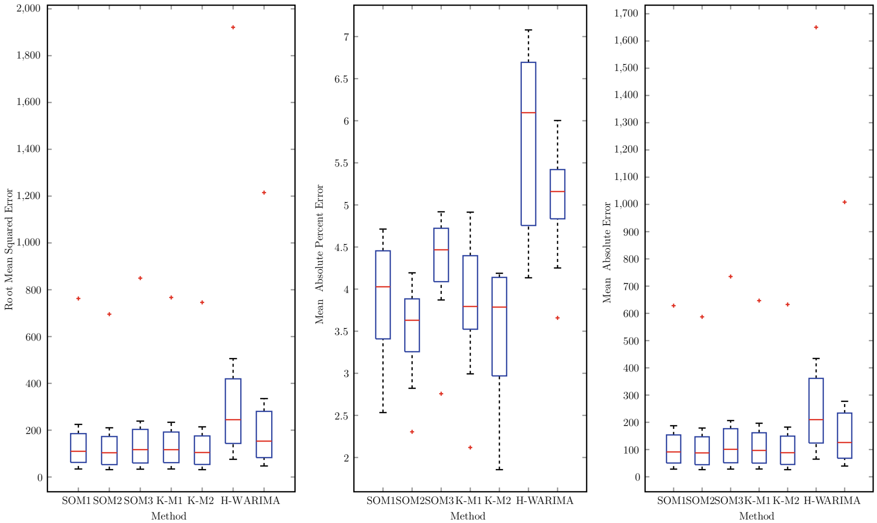


Fig. 4 Boxplot of error metrics of all load forecasting models studied

Table 3 Description of TUCP case studies developed

Case ID	Zone code	Zone ID	Supplied demand (%)	System	Season
1	NEMASSBOST	9	10	IEEE14	Winter
2	SEMASS	7	20	IEEE30	Spring
3	WCMASS	8	65	IEEE57	Summer

Table 4 TUCP simulation results

Case ID	Working day costs	Saturday costs	Sunday costs	Holiday costs	Average time (s)
1	25616	21867	24468	25847	176.42
2	26341	23223	25600	21361	203.03
3	251657	212500	244771	269410	290.05

Results are presented in Table 4 with total planning costs for each period and the average run time of DP algorithm. These results are detailed in Fig. 5 for Case 1, in Fig. 6 for Case 2, and in Fig. 7 for Case 3.

For all cases, the forecasted demand was supplied for the available generation. However, since this schedule is based on information provided by forecasted data, it should be verified according to a validation provided by an ideal plan.

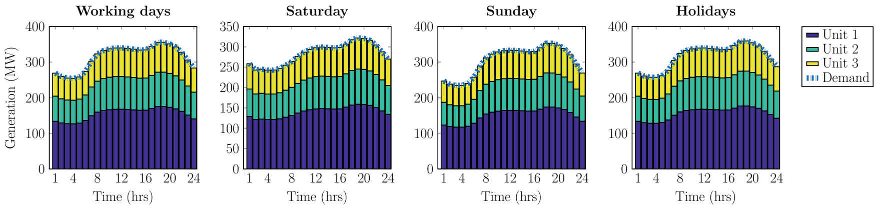


Fig. 5 Detailed results for Case 1

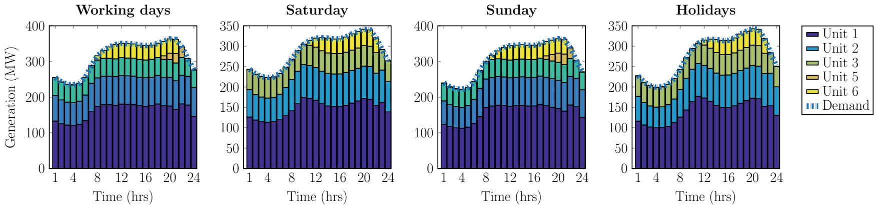


Fig. 6 Detailed results for Case 2

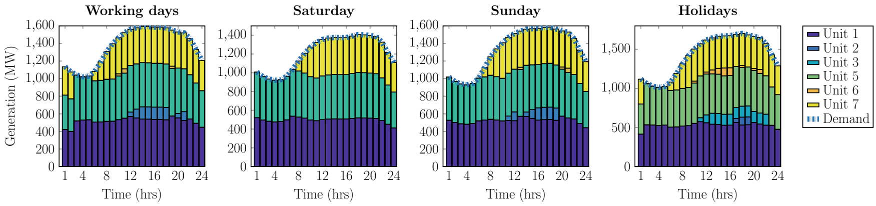


Fig. 7 Detailed results for Case 3

Overall Evaluation

The last step of the proposed framework consists of evaluating the observed planning of all cases according to the expected values, obtained from actual demand as calculated through Ω_S, which verifies the following null hypothesis, whose results are presented in Table 5:

Table 5 Chi-squared hypothesis test results

Case ID	χ^2 value	p-value	Degree of freedom	H_0
1	144	0.4843	144	Reject
2	173.6190	0.4725	173	Reject
3	159.0233	0.5069	160	Reject

H0: There are significant differences between planning obtained through forecasting and an ideal plan.

Overall, the null hypothesis was rejected in all presented tests, which means that unit commitment results may handle the actual demand. Even considering different methods that achieved a similar performance, as evaluated in load forecasting step, the demand may be supplied by available generation in most cases.

Concluding Remarks

The PBSFMs were applied to a framework that solves both STLF and TUCP, with promising results that confirm the importance of forecasting methods in generation schedule planning tasks. The statistical information presented how these steps are crucial for power systems operation.

This study can be reinforced by some other methods applied to unit commitment, as well as some alternate load forecasting methods and other sources of generation as well. The study of several other power system tasks such as power flows, generation, transmission and distribution expansion planning, transient analysis, and their influence or integration possibilities can be considered for a deeper analysis.

Acknowledgements The authors acknowledge the support of CNPq, FAPEMIG and CAPES in this study. Special thanks to Vladimir Stanojevic for his DP implementation in MATLAB and to prof. Frederico Coelho (DELT-UFMG) for explanations regarding some statistical tests.

References

1. Anjos, M.F., Conejo, A.J.: Unit commitment in electric energy systems. Found. Trends Electr. Energy Syst. **1**(4), 220–310 (2017)
2. Dudek, G.: Pattern similarity-based methods for short-term load forecasting—Part 1: principles. Appl. Soft Comput. **37**, 277–287 (2015)
3. Dudek, G.: Pattern similarity-based methods for short-term load forecasting—Part 2: models. Appl. Soft Comput. **36**, 422–441 (2015)
4. Dudek, G.: Neural networks for pattern-based short-term load forecasting: a comparative study. Neurocomputing **205**, 64–74 (2016)
5. Girgis, A.A., Varadan, S.: Unit commitment using load forecasting based on artificial neural networks. Electr. Pow. Syst. Res. **32**(3), 213–217 (1995)
6. Greenwood, P.E., Nikulin, M.S.: A Guide to Chi-squared Testing. Wiley Series in Probability and Statistics. Wiley, New York (1996)
7. Gross, G., Galiana, F.D.: Short-term load forecasting. Proc. IEEE **75**(12), 1558–1573 (1987)
8. Hobbs, W.J., Hermon, G., Warner, S., Shelbe, G.B.: An enhanced dynamic programming approach for unit commitment. IEEE Trans. Power Syst. **3**(3), 1201–1205 (1988)
9. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. Int. J. Forecasting **22**(4), 679–688 (2006)
10. ISO-NE Homepage. http://www.iso-ne.com/

11. Kamboj, V.K., Bath, S.K.: A solution to energy and environmental problems of electric power system using hybrid harmony search-random search optimization algorithm. Cogent Eng. **3**(1), 1–29 (2016)
12. Karthikeyan, S.P., Palanisamy, K., Rani, C., Raglend, I.J., Kothari, D.P.: Security constrained unit commitment problem with operational, power flow and environmental constraints. WSEAS Trans. Power Syst. **4**(4), 53–56 (2009)
13. Krishnamurthy, D., Li, W., Tesfatsion, L.: An 8-Zone test system based on ISO New England data: development and application. IEEE Trans. Power Syst. **31**(1), 234–246 (2016)
14. Lang, K., Zhang, M., Yuan, Y.: Improved neural networks with random weights for short-term load forecasting. PLOS ONE **10**(12), e0143175 (2015)
15. Li, C.-A., Johnson, R.B., Svoboda, A.J.: A new unit commitment method. IEEE Trans. Power Syst. **12**(1), 113–119 (1997)
16. Raza, M.Q., Khosravi, A.: A review on artificial intelligence based load demand forecasting techniques for smart grid and building. Renew. Sustain. Energy Rev. **50**, 1352–1372 (2015)
17. Song, C., Zheng, Y.: The delicate analysis of short-term load forecasting. IOP Conf. Ser. Mater. Sci. Eng. **199**(1), 012097 (2017)
18. Taylor, J.W., McSharry, P.E.: Short-term load forecasting methods: an evaluation based on european data. IEEE Trans. Power Syst. **22**(4), 2213–2219 (2007)
19. Wei, S., Mohan, L.: Application of improved artificial neural networks in short-term power load forecasting J. Renew. Sustain. Energy, **7**(4) (2015)
20. Weron, R.: Modeling and Forecasting Electricity Loads and Prices. Wiley, Chichester (2006)
21. Willmott, C., Matsuura, K.: Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Res. **30**, 79–82 (2005)
22. Zor, K., Timur, O., Çelik, Ö., Yildirim, H.B., Teke, A.: Interpretation of error calculation methods in the context of energy forecasting. In: 12th Conference on Sustainable Development of Energy, Water and Environment Systems, Dubrovnik, Croatia, vol. 722, pp. 1–9 (2017)

ICA-Derived Respiration Using an Adaptive R-Peak Detector

Christina Kozia, Randa Herzallah and David Lowe

Abstract Breathing Rate (BR) plays a key role in health deterioration monitoring. Despite that, it has been neglected due to inadequate nursing skills and insufficient equipment. ECG signal, which is always monitored in a hospital ward, is affected by respiration which makes it a highly appealing way for the BR estimation. In addition, the latter requires accurate R-peak detection, which is a continuing concern because current methods are still inaccurate and miss heart beats. We describe a systematic approach for robust and accurate BR estimation based on the respiration-modulated ECG signal. Increased accuracy is obtained by a time-varying adaptive threshold for QRS complex identification, and discriminating high amplitude Q-peaks as false R-peaks. Improved robustness derives from the use of an Empirical Mode Decomposition (EMD) approach to R-peak detection and an Independent Component Analysis (ICA) used to separate out the respiration signal in the frequency domain as opposed to the more usual time domain approaches. The performance of our system, tested on real data from the Capnobase dataset, returned an average mean absolute error of less than 0.7 breaths per minute compared with up to 15 breaths per minute produced by some of the best time domain analysis approaches. Additionally, the QRS detector component part of our system is competitive with the best current published methods, achieving a detection rate of over 99.80% on real data.

Keywords ICA · Frequency domain analysis · R-peak detection · EMD · Local signal energy

Supported by Isansys Lifecare.

C. Kozia (✉) · R. Herzallah · D. Lowe
Aston University, Birmingham B4 7ET, UK
e-mail: koziac@aston.ac.uk

R. Herzallah
e-mail: r.herzallah@aston.ac.uk

D. Lowe
e-mail: david.lowe@deeplearn.net

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1_25

Introduction

Over the past two decades, there has been a dramatic increase in the extraction of the respiration signal and its derivative, the Breathing Rate (BR), from a single-lead ECG [1]. The BR is a key indicator in health deterioration monitoring and provides valuable information on the physiological status of a person [2]. The first serious discussions and research into extraction of the respiratory signal using the ECG emerged during the 1980s and the use of the latter was suggested by reporting that a normal respiratory cycle affects the heart rate, which causes sinus arrhythmia [1]. An ECG-Derived Respiration (EDR) method was suggested in [3]. EDR uses the variations in the QRS peak amplitudes through time due to the chest movement amplitude displacements, whereas Respiratory Sinus Arrhythmia-Derived Respiration (RSA) uses the Instantaneous Heart Rate (IHR) variability which increases during inspiration and drops during exhalation due to baroreflex sensitivity [4].

More recent attention has focused on the analysis of the beat-to-beat changes in the QRS complex in order to estimate the BR. In [5] a new method for extracting the respiration signal based on the Principal Component Analysis (PCA) was suggested. It has been argued that PCA can describe the variability of the QRS complex due to respiration. Their proposed algorithm has been compared with existing EDR methods and the results were promising. In [6] an EDR method based on Independent Component Analysis (ICA) and the power spectrum of the independent components was proposed. This method is more robust as ICA assumes statistical independence between the components, whereas PCA assumes that the components are mutually uncorrelated.

As was pointed out above, it is clear that the respiration signal extraction requires an accurate QRS detector [1, 3, 4]. The identification of QRS complexes is a major area of interest as it is the most discernible feature of the ECG. However, R-peak detection turns out to be complex due to signal contamination from various types of noise and the morphological variability of the QRS complexes. Several studies have investigated the QRS complex detection by developing a number of methods [7–11]. The signal is pre-processed in order to amplify the QRS complex and remove noise and baseline wander and then a set of thresholds is applied in order to locate the real R-peaks in the signal. In [7, 8] the derivative of the signal was used in order to obtain the slope and width information of the QRS complex. In [11] the Hilbert transform of the derivative of the ECG was suggested in order to locate the R-peaks. More recent studies have investigated the use of Empirical Mode Decomposition (EMD) [12] as an effective pre-processor which amplifies the QRS complex and decomposes the signal into a set of Intrinsic Mode Functions (IMF) [13–15]. A serious weakness of the above methods is that the threshold of R-peak detection is derived from the full length ECG. Generally, difficulties arise when the signal includes very large R-peaks, making the threshold high. This results in the failure to detect lower amplitude R-peaks.

The current study proposes an alternative adaptive QRS detector which overcomes the aforementioned problems in the current state-of-the-art EMD methods by introducing an adaptive threshold which is calculated from the local energy of the reconstructed ECG signal from the EMD. The pre-processing stage contains a band-pass filter which eliminates noise and reduces the number of the IMFs. The reconstruction of the signal using the EMD method facilitates the removal of low-frequency interference and the absolute value of the reconstructed signal amplifies the QRS complexes. Compared with prior work our locally-adaptive approach has three distinct features. Firstly, the proposed detector provides a solution for the detection of small R-peaks by establishing a threshold derived from the mean of the root mean square (rms) over recent data segments. Secondly, the threshold based on the local signal energy of each segment allows tracking of non-stationary noise. Thirdly, it allows the system to distinguish R-peaks from large Q-peaks using the absolute value of the signal and the first derivative of the ECG signal.

The respiratory-induced modulations of the ECG signal are extracted using statistical Independent Component Analysis (ICA) on the matrix of temporally-ordered R-peak synchronised time series segments. The Independent Component (IC) corresponding to respiration is identified based on its frequency spectrum, and this respiratory IC is then filtered in order to extract the ICA respiration signal. An advantage of working with the frequency domain analysis over the time domain is that the time domain is sensitive to double-humped peaks, which makes BR estimation inaccurate in the time domain. Finally, the identification of the respiratory IC in frequency domain gives the respiratory bands, which makes the BR estimation more accurate.

Proposed QRS Detector

The main assumption of our proposed R-peak detector is that the QRS complex can be further amplified by reconstructing the signal from its first three IMFs after applying the EMD algorithm. Our assumption has been verified on all the tested recordings as will be shown in section "QRS Detector Algorithm".

Empirical Mode Decomposition

EMD decomposes the signal, $x(t)$, into a series of narrow-band signals, $c_i(t)$, which are called IMFs, and fulfil special conditions. An oscillatory mode of the signal is an IMF exclusively under the conditions that: first, in the whole dataset, the number of zero-crossings and the number of extrema are either equal or differ at most by one; and second, at any point, the mean value of the maximum and the minimum envelope is zero. The key advantage of EMD is that it is a data-driven analysis method. At each iteration the algorithm needs to decide if the i-th component, $h_i(t)$, extracted from the data, is an IMF based on the conditions mentioned above. In order to achieve this, the EMD method uses a systematic way which is called the *sifting process* and is described as follows: for a given signal $x(t)$, the extrema points are first identified, followed by approximation of the upper, $\hat{r}(t)$, and lower, $\underline{r}(t)$, envelopes of the signal through a cubic spline interpolation. The mean of the envelopes is then obtained, and the ith component, $h_i(t)$, is computed as the difference between the signal and the mean. The sifting process has to be repeated as many times as required to reduce the extracted signal to an IMF. For our implementation, in order to stop the sifting process, the number of zero-crossings and the number of extrema are checked for equality or whether they differ at most by one. If the final residue, $r_N(t)$, is obtained as a monotonic function, the EMD algorithm is terminated, $c_i = h_i$, and the signal, $x(t)$, can be written as follows:

$$x(t) = \sum_{i=1}^{N} c_i(t) + r_N(t), \tag{1}$$

where N is the total number of the extracted IMFs.

QRS Detector Algorithm

To summarise, the proposed QRS detector is as follows.

Pre-processing Stage

1. The raw signal, $x(t)$, is first filtered with a band-pass filter, whose coefficients were designed using a Kaiser-Bessel window [16]. The band-stop frequencies were set at 8 and 20 Hz [17] in order to amplify the QRS complex, eliminate noise and reduce the number of IMFs. The output of the filter is denoted as $x_f(t)$,
2. The EMD algorithm is applied on $x_f(t)$ to extract the IMFs, $c_1(t), \ldots, c_N(t)$, where N is the total number of the extracted IMFs,
3. The signal is reconstructed by adding the first three IMFs,

$$x_r(t) = \sum_{i=1}^{3} c_i(t), \tag{2}$$

   where the number of IMFs is experimentally selected and it will be discussed later,
4. Then, the absolute value of the reconstructed signal is computed, that is $a(t) = |x_r(t)|$. This makes all data points positive and implements a linear amplification of the reconstructed signal emphasising the higher cardiac frequencies, such as the QRS complex.

Decision Stage

5. In order to increase the efficiency of the algorithm, we divide $a(t)$ into k segments of 3 s duration, that is $k = (\text{total number of samples})/(3 * fs)$.

The starting point of the k-th segment should match the last R-peak located on the $k-1$ segment in order to increase the accuracy,

6. Compute the envelope of the maxima, $\hat{a}_k(t)$, of $a_k(t)$ for each segment k through a cubic spline interpolation of the local maxima,
7. Compute the local signal energy for each segment as,

$$RMS_k = \sqrt{\frac{1}{M}\sum_{t=1}^{M}[\hat{a}_k(t)]^2}, \tag{3}$$

where k is the current segment and M is the number of samples in the segment, that is $M = 3 * fs$,

8. The threshold of the k-th segment is set to be the mean of the eight most recent RMS_k values,

$$T_k = \frac{1}{8}\sum_{j=k-7}^{k} RMS_j, \tag{4}$$

9. The peaks, which exceed threshold T_k in the absolute sequence $a_k(t)$, are classified as candidate peaks,
10. In order to segregate large Q-peaks from R-peaks, we compute the first derivative of $x_r(t)$. Peaks with a negative derivative will be investigated further at the refractory period check given next,
11. Apply a refractory period check when the R-R interval of two adjacent peaks is less than 200 ms. Keep the peak with the maximum amplitude.

The main assumption of the proposed R-peak detector is that the first three IMFs can be used in the QRS complex amplification, because their frequency spectra correspond to the QRS frequencies (3–40 Hz [18]). Furthermore, the IMFs which represent slower oscillations, like P- or T-waves (0.7–10 Hz [18]), are omitted in order to enhance the R-peaks. To enhance the QRS complex and reduce computation time and the number of the extracted IMFs, it is recommended that the ECG signal is filtered with a band-pass filter. Figure 1 shows a filtered ECG signal and its first five IMFs, obtained after the EMD algorithm. It is evident from Plots 7 to 11 that as the order of the IMFs increases, the frequency of the mode decreases. It is also shown that the spectra of the first three IMFs correspond to the frequency band of the QRS complex. Figure 2 shows that the filtered signal, $x_f(t)$, can be approximated by the reconstructed signal, $x_r(t)$, because the difference of the two signals (yellow dotted line) is small and the shape of the QRS complex is preserved.

Our study used a convenience sample of 20 children recordings (median age: 9.1, range 1–16 years) and 5 adult recordings (median age: 46.2, range 37–64) from the Capnobase dataset [19]. The latter contains ECG, pulse oximetry and capnography recordings of 8 min duration acquired during elective surgery and routine anaesthesia. The sampling frequency, which was used for the ECG acquisition, was set to be at 300 Hz. Furthermore, the dataset contains fully annotated data, providing the actual R-peaks and reference BR data. The results obtained from our proposed QRS detector

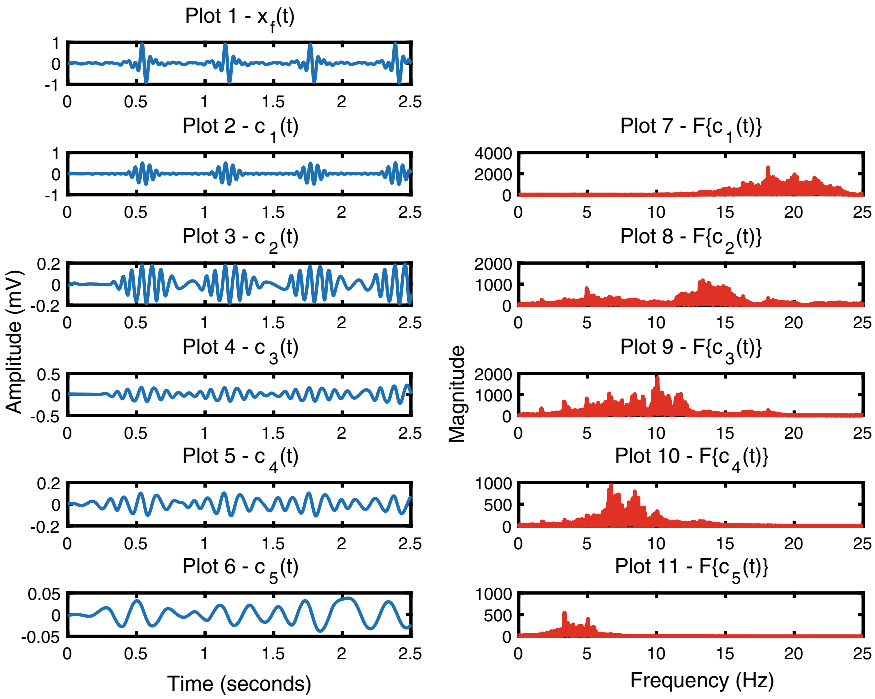


Fig. 1 The result on the EMD and the spectrum of each IMF. Plot 1 corresponds to the filtered ECG, $x_f(t)$. Plots 2–6 correspond to the first five IMFs. Plots 7–11 correspond to the Fourier transform of each IMF

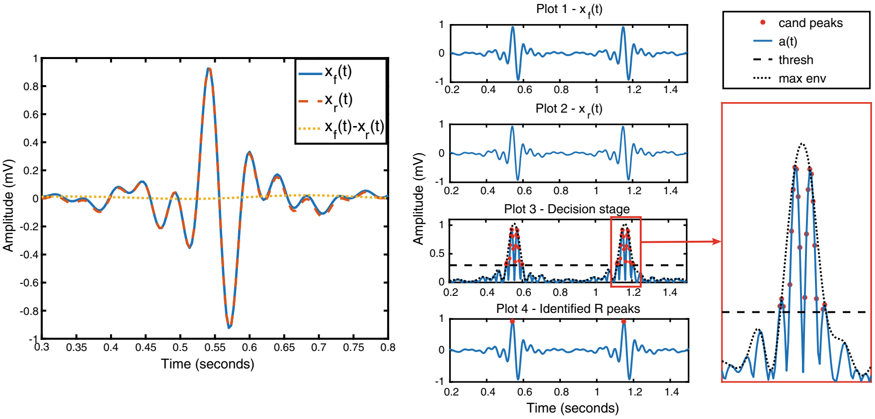


Fig. 2 Left plot—Reconstruction of the filtered signal, $x_f(t)$, by the summation of the first three IMFs, $x_r(t)$, and their difference, $x_f(t) - x_r(t)$. Right plot—Steps of the QRS detector. Plot 1, corresponds to the filtered ECG signal, $x_f(t)$. Plot 2, corresponds to the reconstructed signal, $x_r(t)$. Plot 3, shows the absolute sequence, $a_k(t)$, (blue line) and its maximum envelope, $\hat{a}_k(t)$, (dotted black line) along with the threshold (dashed black horizontal line) and candidate peaks marked with a red asterisk ‘*’. Plot 4, shows the identified R peaks on $x_f(t)$ as red asterisk ‘*’

Table 1 QRS detection performance using the Capnobase dataset

Rec	Annotated peaks	DER (%)	*Se* (%)	+*P* (%)
9	815	0.00	100	100
15	960	0.00	100	100
16	1012	0.00	100	100
18	1131	0.00	100	100
23	818	0.00	100	100
32	685	0.00	100	100
35	900	0.18	100	99.89
38	956	0.00	100	100
103	826	0.00	100	100
104	912	0.00	100	100
105	530	0.37	100	99.62
121	579	0.00	100	100
122	588	0.00	100	100
125	627	0.00	100	100
127	615	0.00	100	100
128	541	0.18	100	99.82
134	578	1.53	100	98.50
142	739	0.00	100	100
147	538	3.58	100	97.52
148	624	0.00	100	100
311	555	0.17	100	99.82
312	432	0.00	100	100
313	588	0.00	100	100
322	589	0.16	100	99.83
325	584	0.00	100	100
Average	**17716**	**0.24**	**100**	**99.80**

Table 2 Comparison of QRS detector performance with other methods for Capnobase dataset

Method	DER (%)	*Se* (%)	+*P* (%)
Derivative based [7]	0.25	100	99.78
Hilbert transform [11]	0.31	100	99.70
Our method	**0.24**	**100**	**99.80**

are shown in Table 1. Table 2 shows a comparison of our method's performance with other existing detectors. Figure 2 shows the sequential steps of the R-peak detection method. The detected R-peaks are marked by an asterisk '*' on the filtered signal, $x_f(t)$ (Plot 4). A false negative (FN) occurs when the algorithm fails to detect an actual R-peak. A false positive (FP) represents a false peak detection. *Sensitivity*

(Se), *Positive Predictivity* (+P) and *Detection Error Rate* (DER) were calculated for each recording using the following formulas, respectively:

$$\mathrm{Se}(\%) = \frac{TP}{TP + FN}\% \ , \tag{5}$$

$$+\,\mathrm{P}(\%) = \frac{TP}{TP + FP}\% \ , \tag{6}$$

$$DER(\%) = \frac{FP + FN}{\text{total number of R peaks}}\%, \tag{7}$$

where TP (true positives) is the total number of R-peaks correctly identified. As can be seen from Tables 1 and 2 our QRS detector shows a better performance for the Capnobase records, achieving a *Se* of 100%, a higher $+P$ of 99.80% compared to 99.78% in [7] and 99.70% in [11], as well as a lower DER of 0.24% compared to 0.25% in [7] and 0.31% in [11].

An important observation is that the absolute amplitude of a Q-peak occurs to be larger than the R-peak in some cases. This was found to identify the Q-peak as a real R-peak, because the threshold is applied to the absolute waveform. Therefore, the first derivative of the ECG signal is computed which facilitates the Q-, R-peak segregation. Peaks with a negative derivative will be investigated further in the decision stage by applying the refractory period check. Secondly, another significant advantage of our proposed method is that it can be implemented in real-time with a detection delay equal to the duration of the segment. Online implementation requires a small alteration of our method. The segmentation can be executed at the very beginning of the pre-processing stage and the sequential steps of our algorithm can be implemented for each segment.

Proposed BR Estimation

The main idea of our proposed BR estimation method is to analyse the beat-to-beat changes in the QRS complex in order to extract the respiration signal using the ICA. In order to derive the respiration signal, the R-peaks are located using our proposed QRS detector (section "Proposed QRS Detector"). The QRS complexes are then stacked into a matrix and then the ICA is applied. Using the frequency domain to identify the IC corresponding to the respiration signal provides a robustness that is not accessible from time domain processing [20]. The main assumption is that the BR provides the most power and hence provides the dominant peak of the respiration frequency spectrum.

Independent Component Analysis

Blind Source Separation (BSS) represents a model where a set of unobserved source signals is been recovered from a set of observed mixtures [21, 22]. In other words, BSS is the separation of a set of source signals, $s(t) = \{s_1(t), s_2(t), \cdots, s_n(t)\}$, from a set of mixed signals, $x(t) = \{x_1(t), x_2(t), \cdots, x_m(t)\}$. In ECG signal processing, we assume that the sources are signals like the electrical heart activity, the respiration and some noise, which are mixed, and give as result the observed ECG (Fig. 3). Since it was reported in 1986, ICA has been attracting a lot of interest in machine learning and signal processing [23, 24]. ICA is a statistical approach whose main assumption is that the observations are actually linear combinations of statistically independent variables [21, 25], as follows:

$$x_i = a_{i,1}s_1 + \cdots + a_{i,k}s_k + \cdots + a_{i,n}s_n, \tag{8}$$

where $\boldsymbol{a}_k$ are the basis vectors which form the columns of the mixing matrix $\boldsymbol{A}$ (Fig. 3) and $i = 1, \cdots, m$.

ICADR Algorithm

To summarise, the proposed ICA-Derived Respiration (ICADR) method is as follows:

1. Extract the R-peaks from a single-lead ECG signal using our proposed QRS detector,
2. Create the data matrix and whiten the data,
3. Apply PCA method for dimensionality reduction and keep the first 10 Principal Components (PC),

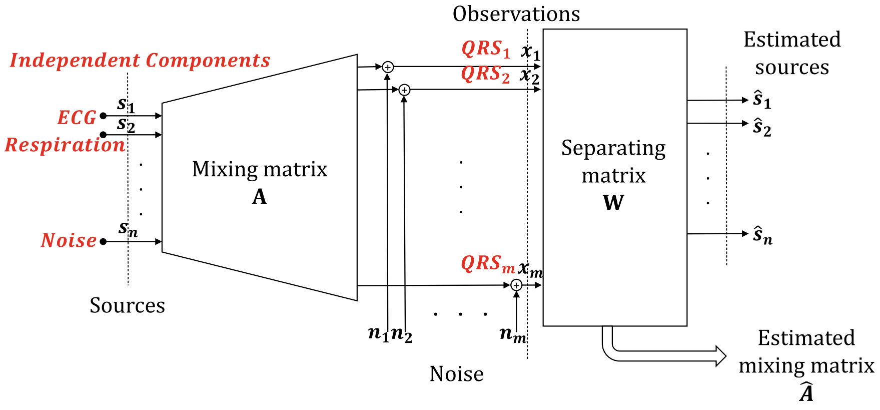


Fig. 3 Schematic representation of the BSS model in ECG signal processing

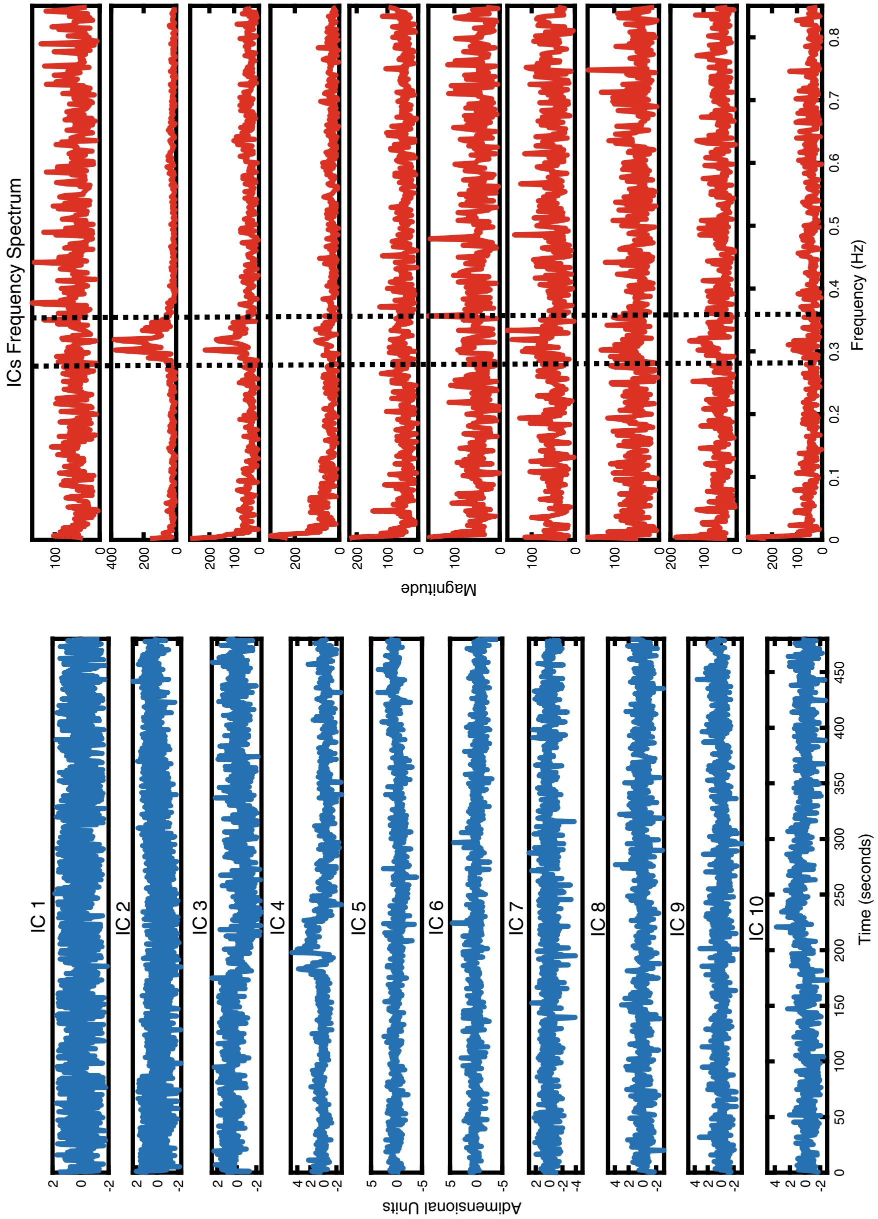
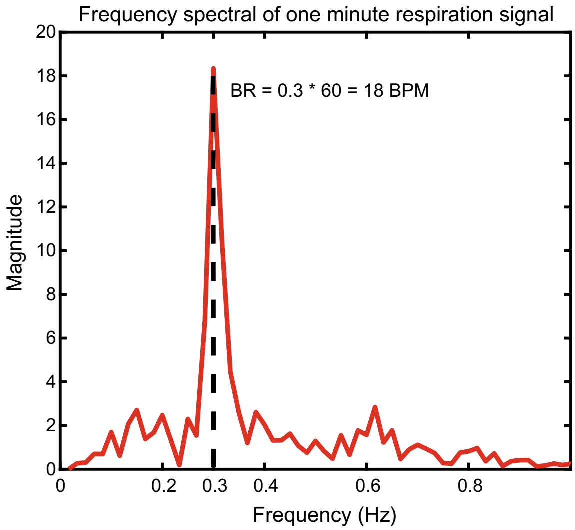


Fig. 4 The plots on the left show the 10 extracted ICs (blue) from an ECG recording and the plots on the right show the corresponding frequency spectra (red). The vertical dotted lines define the frequency range of the IC2, which is chosen to be the ICA respiration signal, because its frequencies lie between reasonable respiration rates (12–24 bpm)

Fig. 5 The frequency spectrum of 1-minute window respiration signal. The dominant peak is located (black dashed line) and it is converted to breaths per minute (bpm)



4. Apply the fastICA algorithm to the reduced matrix in order to get the source signals as independent components (Fig. 4),
5. Compute the Fourier transform of the ICs produced and choose as respiration signal the IC whose frequencies lie between reasonable respiration rates (Fig. 4),
6. Filter the output of step 5 within reasonable respiration frequencies (0.0666–0.5 Hz) to get the ICA respiration signal,
7. Divide the respiration signal into 1-minute windows,
8. For each window, compute the Fourier transform and locate the frequency that corresponds to the most dominant peak in the frequency spectrum (Fig. 5),
9. Convert frequency to bpm to get the BR.

The data matrix construction is based on the fact that we are interested in analysing the inter-beat variations of the ECG features. Using the detected R-peaks (section "QRS Detector Algorithm") the data matrix $\boldsymbol{X}$ is constructed from a single-lead ECG by stacking m segments of length n around the R-peak, that is, $\boldsymbol{X}_{m\times n} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_m]^T$, where $\boldsymbol{x}_i$ is a row vector of length n, that is $\boldsymbol{x}_i = [x_{i,1} \cdots x_{i,j} \cdots x_{i,n}]$. Hence, the first row contains n samples around the first R-peak, the second row contains n samples around the second R-peak etc. For example, in recording 0009 there are 815 detected R-peaks. We are taking 100 points around the R-peak, thus the size of the matrix is 815 × 100. After applying the PCA for dimensionality reduction, the size of the matrix reduces to 10 × 815.

For all the recordings the results obtained from our ICADR method are shown in Table 3. In order to evaluate the performance of our method, we computed the Mean Absolute Error (MAE) in bpm, as follows:

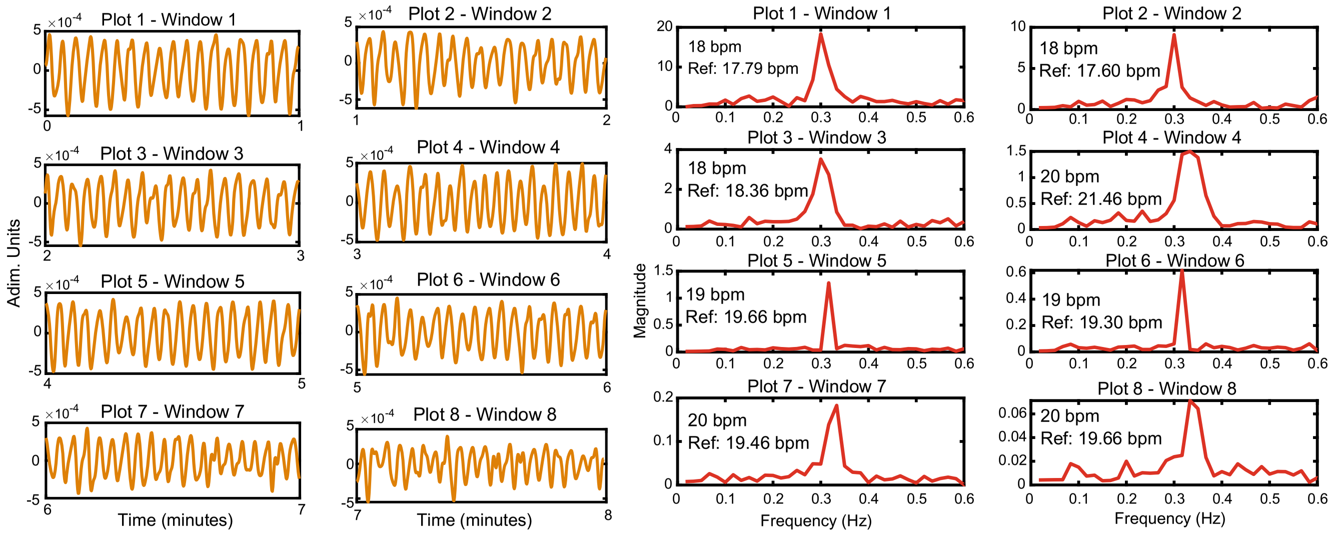


Fig. 6 Left plot—The respiration signal obtained from our ICADR method for recording 0009. Plots 1–8 correspond to the 8 one minute windows obtained from our algorithm. Right plot—Frequency domain analysis of the respiration signal for recording 0009. Plots 1–8 show the frequency spectra of each 1-minute window, along with the reference BR

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |BR_{ref_i} - BR_{est_i}|, \tag{9}$$

where N is the number of the extracted BRs, BR_{ref} and BR_{est} are the reference and estimated BR respectively. The obtained ICA respiration signal was analysed using our proposed frequency domain analysis for the BR estimation. Furthermore, the respiration signal was later upsampled at 8 Hz and then filtered using a band-pass

Table 3 ICA-derived respiration evaluation performance

	Frequency domain		Time domain
	No post-processing	Post-processing	
	MAE (bpm)		MAE (bpm)
Children recordings			
0009	0.6359	0.5109	7.2088
0015	0.2086	0.2086	1.0354
0016	0.0048	0.0048	37.4550
0018	1.2587	1.3837	14.8083
0023	0.4867	0.6117	8.9675
0032	2.3126	1.2261	18.1932
0035	2.1316	1.9887	17.0079
0038	1.6166	1.6166	16.8503
0103	0.0079	0.0079	22.3282
0104	0.0049	0.0049	22.0298
0105	0.0276	0.0276	12.8290
0121	0.0069	0.0069	13.2939
0122	0.0070	0.0070	19.5208
0125	0.3816	0.3623	15.0810
0127	0.4262	0.4262	7.5363
0128	0.3477	0.3477	11.7071
0134	0.3788	0.5038	13.4641
0142	0.1368	0.1368	18.4440
0147	8.8772	5.3786	19.2597
0148	0.0051	0.0051	19.5607
Adult recordings			
0311	1.8822	0.8878	14.4469
0312	0.0096	0.0096	12.3631
0313	0.0110	0.0110	8.8237
0322	0.5638	0.5648	4.2980
0325	1.5150	0.1400	18.4905
Average	**0.9297**	**0.6551**	**15.0000**

filter within a reasonable respiration frequency range. It is evident from our results that the post-processing stage improved the accuracy of the estimation about 0.2747 bpm, achieving an average MAE of 0.6551 bpm. Figure 6 shows the respiration signal obtained from our method for recording 0009, along with the frequency domain analysis, which shows a clear dominant peak in each window. It can be also observed that the error for recording 0147 is high about 8 bpm, and after the post-processing stage it drops to 5 bpm. One possible explanation is that the existence of false positives in the R-peak detection affects the accuracy. For recording 0147, the number of false positives was 13 compared to 0 false positives for recording 0009.

In order to prove that our BR estimation method in the frequency domain outperforms the time domain analysis [20], we compared our approach with the time domain of the respiration signal. The results obtained are shown in Table 3. A basic peak detector is used in order to identify the peaks in the respiration signal, along with their time locations [20]. The evaluation of the Instantaneous Breathing Rate (IBR) from the time intervals of consecutive peaks in the respiration signal is attempted. As soon as the IBRs are computed, they are converted to bpm in order to estimate the BR. It can be observed that the MAE obtained from the time domain analysis is high about 15 bpm. The major drawback of this method is that the existence of double-humped peaks in the respiration signal affects the peak detection.

Conclusion

To conclude, a new ICA-Derived Respiration (ICADR) method was presented based on an accurate and adaptive R-peak detector. Moreover, a frequency domain analysis of the respiration signal was proposed in this study for the BR estimation. The ICADR achieved a low average MAE of 0.6551 bpm for both children and adult recordings. The frequency domain analysis outperforms the BR estimation in the time domain.

The present study also suggested a new R-peak detector based on the EMD of the ECG signal and an adaptive threshold which relies on the local signal energy. Our method facilitates the detection of small R-peaks by deriving a threshold from the mean of the rms over eight segments. The decision stage also contains gradient-based and refractory period checks to differentiate large Q-peaks and omit false R-peaks. For the Capnobase dataset, our R-peak detector is competitive with the best published methods, with a detection accuracy of 99.80%.

References

1. Moody, G.B., Mark, R.G., Zoccola, A., Mantero, S.: Derivation of respiratory signals from multi-lead ECGs. Comput. Cardiol. **12**, 113–116 (1985)
2. Goldhill, D.R., McNarry, A.F., Mandersloot, G., McGinley, A.: A physiologically-based early warning score for ward patients: the association between score and outcome. Anaesthesia **60**(6), 547–553 (2005)

3. Babaeizadeh, S., Zhou, S.H., Pittman, S.D., White, D.P.: Electrocardiogram-derived respiration in screening of sleep-disordered breathing. J. Electrocardiol. **44**(6), 700–706 (2011)
4. Helfenbein, E., Firoozabadi, R., Chien, S., Carlson, E., Babaeizadeh, S.: Development of three methods for extracting respiration from the surface ECG: a review. J. Electrocardiol. **47**(6), 819–825 (2014)
5. Langley, P., Bowers, E.J., Murray, A.: Principal component analysis as a tool for analyzing beat-to-beat changes in ECG features: application to ECG-derived respiration. IEEE Trans. Biomed. Eng. **57**(4), 821–829 (2010)
6. Tiinanen, S., Noponen, K., Tulppo, M., Kiviniemi, A., Seppänen, T.: ECG-derived respiration methods: adapted ICA and PCA. Med. Eng. Phys. **37**(5), 512–517 (2015)
7. Pan, J., Tompkins, W.J.: A real-time QRS detection algorithm. IEEE Trans. Biomed. Eng. BME **32**(3), 230–236 (1985)
8. Hamilton, P.S., Tompkins, W.J.: Quantitative investigation of QRS detection rules using the MIT/BIH arrhythmia database. IEEE Trans. Biomed. Eng. MBE **33**(12), 1157–1165 (1986)
9. Maglaveras, N., Stamkopoulos, T., Diamantaras, K., Pappas, C., Strintzis, M.: ECG pattern recognition and classification using non-linear transformations and neural networks: a review. Int. J. Med. Inform. **52**(1), 191–208 (1998)
10. Kohler, B.-U., Hennig, C., Orglmeister, R.: The principles of software QRS detection. IEEE Eng. Med. Biol. Mag. **21**(1), 42–57 (2002)
11. Benitez, D.S., Gaydecki, P.A., Zaidi, A., Fitzpatrick, A.P.: A new QRS detection algorithm based on the Hilbert transform. In: Computers in Cardiology 2000, pp. 379–382. IEEE (2000)
12. 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.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 454, pp. 903–995. The Royal Society (1998)
13. Yang, X.-L., Tang, J.-T.: Hilbert-Huang transform and Wavelet transform for ECG detection. In: 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2008, pp. 1–4. IEEE (2008)
14. Arafat, M.A., Hasan, M.K.: Automatic detection of ECG wave boundaries using empirical mode decomposition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2009, pp. 461–464. IEEE (2009)
15. Taouli, S.A., Bereksi-Reguig, F.: Detection of QRS complexes in ECG signals based on empirical mode decomposition. Glob. J. Comput. Sci. Technol. **11**(20) (2011)
16. Proakis, J.G., Manolakis, D.G.: Digital Signal Processing: Principles, Algorithms, and Applications, 3rd edn. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (1996)
17. DeBoer, F., Elgendi, M., Jonkman, M.: Frequncy bands effects on QRS detection. In: International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSIGNALS 2010, Valencia, Spain,vol. 5, pp. 428–431 (2005)
18. Malmivuo, J., Plonsey, R.: Bioelectromagnetism: Principles and Applications of Bioelectric and Biomagnetic Fields. Oxford University Press, USA (1995)
19. Karlen, W., Raman, S., Ansermino, J.M., Dumont, G.A.: Multiparameter respiratory rate estimation from the Photoplethysmogram. IEEE Trans. Biomed. Eng. **60**(7), 1946–1953 (2013)
20. Shah, S.A.: Vital sign monitoring and data fusion for paediatric triage. Ph.D. Thesis, University of Oxford (2012)
21. Jutten, C., Herault, J.: Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. Signal Process. **24**(1), 1–10 (1991)
22. Cardose, J.F.: Blind source separation: statistical principles. IEEE Proc. **9035**, 2009–2026 (1998)
23. Herault, J., Jutten, C.: Space or time adaptive signal processing by neural network models. In: AIP Conference Proceedings, vol. 151, pp. 206–211. AIP (1986)
24. Tanskanen, J.M.A., Viik, J.J.: Independent component analysis in ECG signal processing. In: Advances in Electrocardiograms-Methods and Analysis. InTech (2012)
25. Comon, P.: Independent component analysis, a new concept? Signal Process. **36**(3), 287–314 (1994)

Author Index

O. Valenzuela et al. (eds.), *Theory and Applications of Time Series Analysis*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-26036-1