

Wei Lee Woon · Zeyar Aung
Oliver Kramer · Stuart Madnick (Eds.)

Data Analytics for Renewable Energy Integration

Informing the Generation
and Distribution of Renewable Energy

5th ECML PKDD Workshop, DARE 2017
Skopje, Macedonia, September 22, 2017
Revised Selected Papers

LNAI 10691

Lecture Notes in Artificial Intelligence

10691

Subseries of Lecture Notes in Computer Science

LNAI Series Editors

Randy Goebel

University of Alberta, Edmonton, Canada

Yuzuru Tanaka

Hokkaido University, Sapporo, Japan

Wolfgang Wahlster

DFKI and Saarland University, Saarbrücken, Germany

LNAI Founding Series Editor

Joerg Siekmann

DFKI and Saarland University, Saarbrücken, Germany

More information about this series at <http://www.springer.com/series/1244>

Wei Lee Woon · Zeyar Aung
Oliver Kramer · Stuart Madnick (Eds.)

Data Analytics for Renewable Energy Integration

Informing the Generation
and Distribution of Renewable Energy

5th ECML PKDD Workshop, DARE 2017
Skopje, Macedonia, September 22, 2017
Revised Selected Papers

Editors

Wei Lee Woon

Masdar Institute of Science and Technology
Abu Dhabi

United Arab Emirates

Zeyar Aung

Masdar Institute of Science and Technology
Abu Dhabi

United Arab Emirates

Oliver Kramer

Carl von Ossietzky University of Oldenburg
Oldenburg

Germany

Stuart Madnick

MIT Sloan School of Management
Cambridge, MA

USA

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Artificial Intelligence

ISBN 978-3-319-71642-8

ISBN 978-3-319-71643-5 (eBook)

<https://doi.org/10.1007/978-3-319-71643-5>

Library of Congress Control Number: 2017959618

LNCS Sublibrary: SL7 – Artificial Intelligence

© Springer International Publishing AG 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by Springer Nature

The registered company is Springer International Publishing AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume presents a collection of papers focusing on the use of data analytics and machine learning techniques to facilitate the integration of renewable energy resources into existing infrastructure and socioeconomic systems. The papers included were presented at DARE 2017, the 5th International Workshop on Data Analytics for Renewable Energy Integration, which was hosted by ECML PKDD 2017.

In recent times, climate change, energy security, and sustainability have focused a lot of attention on the development of clean and renewable energy sources. However, of equal importance is the issue of integrating these sources into existing infrastructure and socioeconomic systems. While increasing the generating capacities of renewable energy sources is still important, issues such as efficient and cost-effective storage and distribution, demand response, planning, and policy making must be resolved in parallel. These challenges are inherently multidisciplinary and depend heavily on robust and scalable computing techniques and the ability to handle large, complex data sets. The domains of data analytics, pattern recognition, and machine learning are uniquely positioned to offer solutions to many of these challenges. Examples of relevant topics include time series forecasting, the detection of faults, cyber security, smart grid and smart cities, technology integration, demand response, and many others.

This year's event attracted numerous researchers working in the various related domains, both to present and discuss their findings and to share their respective experiences and concerns. We are very grateful to the organizers of ECML PKDD 2017 for hosting DARE 2017, the Program Committee members for their time and assistance, and to the Masdar Institute, MIT, and the University of Oldenburg for their support of this timely and important workshop. Last but not least, we sincerely thank the authors for their valuable contribution to this volume.

October 2017

Wei Lee Woon
Zeyar Aung
Oliver Kramer
Stuart Madnick

Organization

Program Chairs

Wei Lee Woon	Masdar Institute of Science and Technology, UAE
Zeyar Aung	Masdar Institute of Science and Technology, UAE
Oliver Kramer	University of Oldenburg, Germany
Stuart Madnick	Massachusetts Institute of Technology, USA

Program Committee

Fabian Gieseke	University of Copenhagen, Denmark
Paul Kaufmann	Paderborn University, Germany
Anne Kayem	Hasso-Plattner Institute, Germany
Latifur Khan	University of Texas at Dallas, USA
Xiaoli Li	Institute for Infocomm Research, Singapore
Francisco Martínez- Álvarez	Pablo de Olavide University of Seville, Spain
Daisuke Mashima	Advanced Digital Sciences Center, Singapore
Sanaz Mostaghim	Otto von Guericke University Magdeburg, Germany
Taha Ouarda	National Institute of Scientific Research, Canada
Jimmy Peng	National University of Singapore, Singapore
Filipe Saraiva	Federal University of Para, Brazil
Kamran Shafi	University of New South Wales, Australia
A.B.M. Shawkat Ali	University of Fiji, Fiji
Markus Wagner	University of Adelaide, Australia

Contents

Solar Energy Forecasting and Optimization System for Efficient Renewable Energy Integration	1
<i>Diana Manjarres, Ricardo Alonso, Sergio Gil-Lopez, and Itziar Landa-Torres</i>	
Gradient Boosting Models for Photovoltaic Power Estimation Under Partial Shading Conditions	13
<i>Nikolaos Nikolaou, Efstratios Batzelis, and Gavin Brown</i>	
Multi-objective Optimization for Power Load Recommendation Considering User's Comfort	26
<i>Jaroslav Loebl, Helmut Posch, and Viera Rozinajová</i>	
An Approach for Erosion and Power Loss Prediction of Wind Turbines Using Big Data Analytics.	33
<i>Dina Fawzy, Sherin Moussa, and Nagwa Badr</i>	
Minimizing Grid Interaction of Solar Generation and DHW Loads in nZEBs Using Model-Free Reinforcement Learning	47
<i>Adhra Ali and Hussain Kazmi</i>	
Improving Time-Series Rule Matching Performance for Detecting Energy Consumption Patterns	59
<i>Maël Guillemé, Laurence Rozé, Véronique Masson, Cérés Carton, René Quiniou, and Alexandre Termier</i>	
Probabilistic Wind Power Forecasting by Using Quantile Regression Analysis	72
<i>Mehmet Baris Ozkan, Umut Guvengir, Dilek Küçük, Ali Unver Secen, Serkan Buhan, Turan Demirci, Abdullah Bestil, Ceyda Er, and Pinar Karagoz</i>	
Identifying Representative Load Time Series for Load Flow Calculations.	83
<i>Janosch Henze, Tanja Kneiske, Martin Braun, and Bernhard Sick</i>	
Scalable Gaussian Process Models for Solar Power Forecasting	94
<i>Astrid Dahl and Edwin Bonilla</i>	
Wind Speed Forecasting Using Statistical and Machine Learning Methods: A Case Study in the UAE	107
<i>Khawla Al Dhaheri, Wei Lee Woon, and Zeyar Aung</i>	

NWP Ensembles for Wind Energy Uncertainty Estimates.	121
<i>Alejandro Catalina and José R. Dorransoro</i>	
Author Index	133

Solar Energy Forecasting and Optimization System for Efficient Renewable Energy Integration

Diana Manjarres^(✉), Ricardo Alonso, Sergio Gil-Lopez,
and Itziar Landa-Torres

TECNALIA Research and Innovation, 48160 Derio, Spain
{diana.manjarres,ricardo.alonso,sergio.gil,itziar.landa}@tecnalia.com

Abstract. Solar energy forecasting represents a key issue in order to efficiently manage the supply-demand balance and promote an effective renewable energy integration. In this regard, an accurate solar energy forecast is of utmost importance for avoiding large voltage variations into the electricity network and providing the system with mechanisms for managing the produced energy in an optimal way. This paper presents a novel solar energy forecasting and optimization approach called SUNSET which efficiently determines the optimal energy management for the next 24 h in terms of: self-consumption, energy purchase and battery energy storage for later consumption. The proposed SUNSET approach has been tested in a real solar PV system plant installed in Zamudio (Spain) and compared towards a Real-Time (RT) strategy in terms of price and energy savings obtaining attractive results.

Keywords: Solar energy · Renewable energy integration · Optimization · PV energy forecast

1 Introduction

According to the 2020 climate & Energy package, a set of binding legislation is being developed in order to ensure the EU meets its climate and energy targets for the year 2020. Concretely, this package addresses three key points which comprises: (1) a 20% cut in greenhouse gas emissions (from 1990 levels), (2) a 20% of EU energy from renewables and (3) a 20% of improvement in energy efficiency.

Moreover, 2012/27/UE European Directive Article 7 establishes a minimum objective of energy savings in terms of end use of energy for the 2014–2020 period. More specifically, in Spain the cumulated energy savings are expected to reach 15,979 ktep, which represents 659 ktep per year. Furthermore, self-consumption is going to be enhanced and promoted by lawmakers. Regarding the installation of new equipment, European Directive 2009/72/EC establishes that the 80% of end consumers will be equipped with smart meters for 2020 and individual

thermal meters are mandatory from 1 January of 2017. This will facilitate and enable the energy consumption data acquisition and its utilization by the energy forecasting and optimization modules.

Due to the substantial growth of renewable energy systems' installation, an increasing interest is focused on enhancing the modelization and forecasting of solar irradiance and power generation. This information is a key issue for improving the operations control and the optimization of solar energy systems. However, most related studies in the literature ([1–4]) provide a very short-term forecasting of global solar irradiance which is not sufficient for applying an optimal operations control of solar PV systems. Moreover, the large integration of renewable energy sources into the existing or the near-future energy supply structure need for a precise estimation of the electricity production and consumption at every moment. It enables an efficient power grid management and the continuity of the production/consumption relationship. In this context, an accurate forecasting of the solar power generation is required for providing an effective renewable energy integration and avoiding unpredictable large voltage variations into the electricity network.

Regarding Photovoltaic (PV) energy forecasting works in the literature, authors in [5] focus on predicting hourly values of solar power from 21 PV systems located on rooftops in a small village in Denmark for horizons of up to 36 h. A two-stage technique based on statistical normalizations of the solar power and adaptive linear time series models is proposed. Moreover, in [6] a neural network is proposed for 24 h ahead solar power forecasting. The method requires as input past power measurements and meteorological forecasts of solar irradiance, relative humidity and temperature at the site of the PV power system. The key point is that in order to determine the inputs for the Neural Network (NN) and improve the forecast accuracy, a Self-Organized Map (SOM) is trained to classify the local weather type of 24-hour ahead provided by online meteorological services. In the same line of research, authors in [7] proposes a two-dimensional (2-D) representation model of the hourly solar radiation data that includes the between-day correlations along the same hour information. The 2-D forecasting performance is tested through feed-forward NN achieving better forecasting results than linear prediction filters in both 1-D and 2-D. Hourly solar radiation data from the solar observation station in Iki Eylül campus area of Eskisehir region of 1 year is used herein.

In [8] a medium-term solar irradiance forecasting model is proposed by adopting predicted meteorological variables from the US National Weather Service's (NWS) forecasting database as inputs to an Artificial Neural Network (ANN) model. The study also focuses on developing a set of criteria for selecting relevant inputs. The work presented in [9] evaluates and compares several forecasting techniques using no exogenous inputs for predicting the solar power output of a PV power plant operating in Merced, California. The findings show that the ANN-based forecasting models perform better than the other forecasting techniques and also that substantial improvements can be achieved with a GA optimization of the ANN parameters. A diagonal recurrent wavelet neural network

(DRWNN) is newly established in [10] towards presenting a fine forecasting of hourly and daily global solar irradiance that improves the generalization capability of NN in solar forecasting. The proposed method is based on a two phase training algorithm that is validated using both hourly and daily irradiance forecast data sets in Shanghai and Macau. Finally, authors in [11] present a practical method for solar irradiance forecast using a Multilayer Perceptron MLP-model for forecasting the solar irradiance 24 h in advance. The inputs for the NN are values of the mean daily solar irradiance and air temperature in Trieste, Italy. The goodness of the proposed model is validated taking into consideration the forecasted values and the energy produced by the GCPV plant installed on the rooftop of the municipality.

One of the main goals of being capable of inferring the expected PV generation for the next 24 h is to provide the system with mechanisms for managing the produced energy in an optimal way and thus, ensuring an efficient renewable energy integration. In this context, several works in the literature focus on optimally scheduling the energy use in order to enhance self-consumption and minimize the purchase of energy and consequently, the cost of energy. Authors in [12] provide methods to determine the optimal storage size for grid-connected dwellings with PV panels. The work is focused on houses connected to the grid with a small-scale storage to store a part of the solar power for postponed consumption within the day or the next days. Regarding the residential PV self-consumption, the work presented in [13] aims at enhancing PV self-consumption with a battery energy storage system and demand side management (DSM). Similarly, [14] shows the potential to increase PV self-consumption through scheduling and load shifting of programmable appliances in Swedish single-family buildings. With respect to load shifting approaches, authors in [15] and [16] propose different algorithms for optimally scheduling a set of appliances at the end-user premises.

As observed in the literature, most of the works focuses on PV energy forecasting or energy optimization models but are not simultaneously applied to a specific problem. The proposed work goes beyond the state of the art by proposing not only a novel solar energy forecasting model, but also an energy optimization system based on machine learning techniques, coined as SUNSET. The proposed SUNSET approach has been tested in a real PV solar system plant installed in Zamudio (Spain) for which an assessment of the results has been performed. Obtained results shed light on the goodness of the proposed approach which makes use of the PV energy forecast in order to optimally infer the energy use for the following 24 h by minimizing the price of energy and enhancing the PV self-consumption.

The structure of the presented manuscript is the following: Sect. 2 describes the considered solar PV power system for which the proposed approach in Sect. 3 has been applied. Finally, conclusion and future research work is summarized in Sect. 4.

2 Solar PV Power System Description

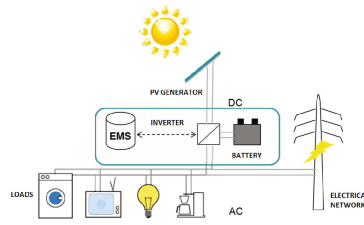
The Solar Thermal Power System proposed in this paper consists of a PV generation system that integrates electricity storage and a smart energy management system.

Figure 1(a) depicts the SUNSET PV solar power system employed in this paper. The PV system is installed in Zamudio (Spain) on a flat roof. It is oriented to the South and mounted on a structure with an inclination of 40° . More concretely, the PV string under study consists of 8 PV panels BP-580F connected in series and to the grid through a PV Inverter SMA Sunny Boy 700.

As already stated in Sect. 1 the main drawback of PV energy is its dependence on variable solar resource. A lot of studies have been carried out to analyze the impact of this unpredictability on grid operation (see Sect. 1): voltage and power flow fluctuation, frequency fluctuation and difficulty of demand-supply management. External costs of solar variability, such as transmission capacity, operating reserve and balancing costs, are believed to be insignificant at low penetration levels, but are expected to rise with higher penetration levels.



(a) SUNSET PV solar power system in Zamudio (Spain).



(b) SUNSET Solar Thermal Power System Diagram.

Fig. 1. Solar thermal power system.

In order to make PV power become predictable and manageable, a potential solution is the utilization of energy storage, as shown in Fig. 1(b). If properly designed, PV self-consumption systems with integrated electrical storage can increase energy savings at the same time they ease the planning and operation of distribution grids. For this purpose, PV generation (and electrical consumption) forecasting tools are required. Economic benefits come from a better utilization of PV excess for peak-shaving and maximizing self-consumption at the most profitable time, taking advantage of known daily evolution of electricity tariffs. As a result, the interface between the prosumers and the rest of the electricity system will be smartened exchanging information about their anticipated electrical performance. Real-time variable electricity pricing can encourage users to adjust usage to take advantage of periods when power is cheaply available and avoid periods when it is scarcer and more expensive, facilitating grid planning and operation.

3 Proposed SUNSET Approach

The proposed SUNSET approach is designed in order to provide a more accurate solar energy forecast and efficiently suggest the optimal energy management proposal for the following hours in terms of self-consumption, energy purchase and battery energy storage for later consumption.

SUNSET provides a compact topology that integrates on the same equipment the energy storage, thus yielding to a more efficient system. Moreover, its flexibility for adapting to the consumer's needs makes SUNSET a cheaper approach, thus optimizing the Capital Expenditures (CapEx) storage. Last but not least, it involves an Energy Management System (EMS) that provides a reliable PV generation and demand forecast, which optimizes the Operational EXpenditures (OpEx) storage and yields to a more cost-effective approach than actual PV systems.

Figure 2 depicts the scheme of the proposed SUNSET approach which comprises the Solar Energy forecasting model and the Energy Optimization model. It is important to note that the PV generation forecast highly depends on the accuracy of the weather forecast services. In this context, the SUNSET approach incorporates an advanced Solar Energy Forecasting Model based on Machine Learning (ML) techniques which is presented in Subsect. 3.1 with the aim at improving the PV energy forecast for a specific localization. As shown in Fig. 2, the estimated PV energy forecast (\hat{E}_{inv}) is the input of the Energy Optimization Model, described in Subsect. 3.2. This model incorporates also as inputs the estimated demand forecast (\hat{E}_{load}), the maximum contracted power (Pot_{max}) with the aim at optimally deciding the energy usage for the following day based on energy cost minimization and an enhanced battery usage.

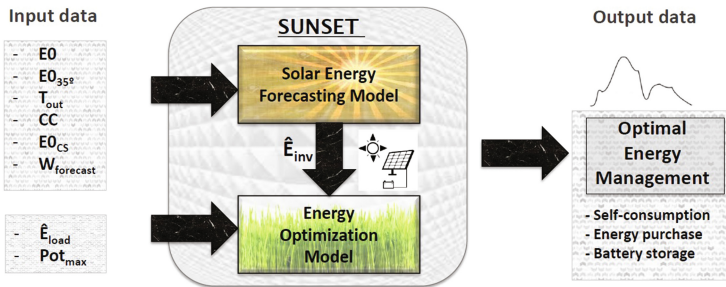


Fig. 2. Scheme of the proposed SUNSET approach.

3.1 SUNSET Solar Energy Forecasting Model

The main core of the proposed SUNSET Solar Energy Forecasting Model is based on an intelligent machine learning method based on Random Forest (RF) [17].

RF technique is a non-linear multiparameter regressor which tries to infer the relationships between the external temperatures (T_{out}), the irradiance on horizontal surface (E_0), the irradiance on 35° (E_{35°), the irradiance on clear sky (E_{CS}), the Cloud Cover (CC) and the weather forecast ($W_{forecast}$) in order to accurately estimate the energy generated by the solar inverter. Therefore, the solar PV system is modelled by means of a black-box model with the aim at inferring the estimated energy generated by the solar system (\hat{E}_{inv}) for the next 24 h.

The proposed SUNSET Solar Energy Forecasting Model has been tested with data coming from a solar PV system of Zamudio (Spain) described in Sect. 2. Current and historical weather variables needed for performing the simulation are obtained in a 3-hour and a 6-hour time horizon by means of the Accuweather platform. In addition, it is evidenced that the accuracy of solar energy forecasting models strongly depends on seasonal characteristics of solar variability. Therefore, input data is filtered based on seasonal characteristics and estimation errors are computed per different types of days, i.e. sunny and cloudy days.

Equations 1 and 2 represent the daily ϵ_d and hourly ϵ_h mean error formulations employed for assessing the proposed SUNSET Solar Energy Forecasting Model, respectively.

$$\epsilon_d = \frac{1}{N_d} \sum_{i=1}^{N_d} \frac{\sum_{j=1}^{N_h} E_{inv|i} - \sum_{j=1}^{N_h} \hat{E}_{inv|i}}{\sum_{j=1}^{N_h} E_{inv|i}} \cdot 100 \quad (1)$$

$$\epsilon_h = \frac{1}{N_d \cdot N_h} \sum_{i=1}^{N_d} \sum_{j=1}^{N_h} \frac{E_{inv|(i,j)} - \hat{E}_{inv|(i,j)}}{E_{inv|(i,j)}} \cdot 100, \quad (2)$$

where N_d refers to the number of simulation days, N_h the number of operating hours (from sunrise to sunset), E_{inv} the actual value of Solar Energy Generation and \hat{E}_{inv} the estimated value.

The following Table 1 presents the error estimation results when forecasting the solar energy generation (E_{inv}) obtained for distinct days' typologies. The proposed model incorporates also ephemerides information to exactly know the time of sunrise and sunset for each day. Therefore, the estimation of PV generation considers the times interval until the sunset time.

The first column presents the hourly and daily error estimation of the Solar Energy Generation (E_{inv}) (%) for sunny days in which the $CC \leq 15\%$. As can be observed the mean hourly error is less than 10%, whereas the mean daily error is around 25%.

By means of the identification and minimization of outlier data, i.e. observations that appear to deviate markedly from other observations in the sample, mean hourly and daily error results significantly decrease (see second column), i.e. both error estimation results are below 10%.

Finally, the third column of Table 1 provides the error results for the total simulation days (224 days), yielding to mean hourly errors near to 15% and mean daily error estimations below 30% with a 6-hour time horizon. Moreover, when referring to Clear Sky (CS) these errors considerably decrease until 10%.

Table 1. Mean error estimation (%) for sunny days, sunny days with outlier minimization and all days without and with respect Clear Sky (CS) irradiance value.

	Sunny days: $CC \leq 15\%$ 6h/3h time horizon	Sunny days: outliers min. 6h time horizon	All days/all days with respect CS 6h time horizon
Hourly error (%)	8.11/7.83	4.70	14.65/10.63
Daily error (%)	29.46/25.67	7.25	27.14/8.91

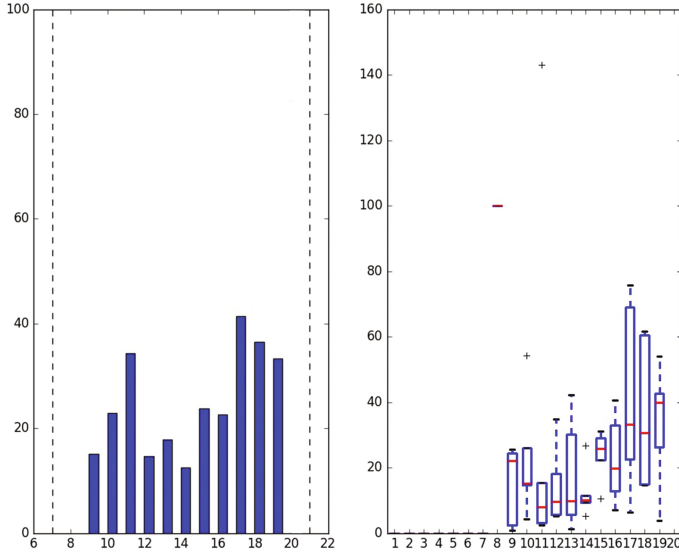
**Fig. 3.** Error estimation for SUNSET Solar Energy Forecasting Model. The histogram of PV generation error estimation (left) and the boxplot representation of % PV generation error estimation per hour (right).

Figure 3 depicts the histogram of PV generation error estimation (left) and the boxplot representation of % PV generation error estimation per hour (right). Note that boxes delimit the lower and upper quartiles, the medians are depicted with a solid line and the outliers are marked with asterisks. It can be observed that median error estimation values are around 20%, growing up at the end of the day.

It is already known that the major number of errors in the PV generation forecast comes from intrinsic errors present in the weather forecast magnitudes. These are in fact the inputs of the proposed SUNSET Solar Energy Forecasting model. In order to check the accuracy of the proposed model without the affection of these intrinsic errors, the model is trained with measured magnitudes. Figure 4 shows the boxplot representation of PV generation error estimation (%) in case the model is trained with measured weather magnitudes. As can be observed,

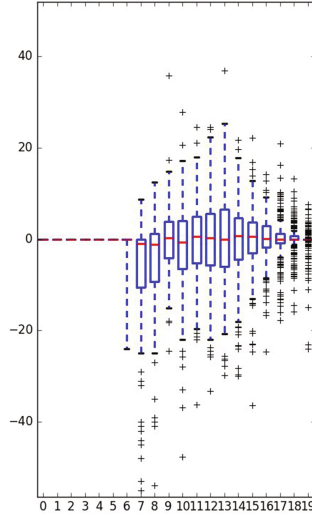


Fig. 4. Boxplot representation of error estimations (%) for SUNSET Solar Energy Forecasting Model by employing measured weather magnitudes as inputs of the model.

error estimation is almost null, i.e. the model is well tuned for forecasting the PV generation.

3.2 SUNSET Energy Optimization Model

This subsection presents the SUNSET Energy Optimization model which steps are depicted in detail in Algorithm 1. The goal of the proposed energy management system consists on optimally deciding the energy usage for the following day based on energy cost minimization and an enhanced battery usage. Therefore, the proposed SUNSET Energy Optimization model prioritizes first self-consumption, then lacks of energy above the maximum contracted power and finally the remaining lacks of energy. In all cases, battery is charged with PV energy from previous excesses or in case there is no PV energy excess, energy is purchased at low price hours controlling the maximum capacity of the battery.

SUNSET Energy Optimization model is compared towards the usually applied Real-Time (RT) control strategy in which PV energy is self-consumed at each time and stored in the battery in case there is excess of PV energy. Nevertheless, RT strategy does not follow any intelligent procedure of purchasing energy, energy is purchased when necessary regardless the energy price or the exceeding of the maximum contracted power.

Figures 5 and 6 depict the energy management performed by SUNSET and the RT approach, respectively. As observed, the SUNSET approach prioritizes energy purchase at low-prices hours (see Table 3) in order to satisfy the total demand and also controls not to exceed the maximum contracted power. Thus, instead of buying energy at the time of the lack of energy, the battery is

Algorithm 1. Proposed SUNSET Energy Optimization Model

```

1 Ask the Accuweather weather forecast service for the estimated  $W_{forecast}$ .
2 Forecast the energy demand of the user  $\hat{E}_{load}$ 
3 Train the SUNSET Solar Energy Forecasting Model with the historical data
  and the weather forecast and infer the estimated PV energy generated by the
  solar system ( $\hat{E}_{inv}$ ) for the next 24h:
4 Prioritize self-consumption:
5 for  $i \leftarrow 00:00h$  to  $23:00h$  do
6   | if  $\hat{E}_{inv}|_i \neq 0$  &  $\hat{E}_{load}|_i \neq 0$  then
7   |   | Self-consume  $\min\{\hat{E}_{inv}|_i, \hat{E}_{load}|_i\}$  at  $i$  hour.
8   end
9 Prioritize lacks above the maximum contracted power ( $Pot_{max}$ ):
  Supply the energy exceeding  $Pot_{max}$ .
10 if  $\hat{E}_{inv}|_{i-1}^0 \neq 0$ : then
11   | Charge the battery with PV from previous excesses  $\hat{E}_{inv}|_{i-1}^0$  in order to
  | supply the energy exceeding  $Pot_{max}$  controlling the maximum capacity of
  | the battery.
12 if  $\hat{E}_{inv}|_{i-1}^0 = 0$ : then
13   | Buy energy and charge the battery evenly at low price hours until the lack
  | controlling its maximum capacity.
14 Remaining lacks of energy: Order the lacks of energy by energy prices at
  the time of the lack.
15 if  $\hat{E}_{inv}|_{i-1}^0 \neq 0$ : then
16   | Charge the battery with PV from previous excesses  $\hat{E}_{inv}|_{i-1}^0$  in order to
  | supply the lack of energy at  $i$  hour.
17 if  $\hat{E}_{inv}|_{i-1}^0 = 0$ : then
18   | Purchase energy at the time of the lack of energy by controlling not to pass
  | the contracted  $Pot_{max}$ .
19

```

intelligently charged in order to enhance the battery lifetime by means of uniform charges and minimize the price of energy. This fact is enlightened when comparing the battery usage for the RT and the SUNSET approach between 10:00 h to 15:00 h. Whereas RT employs a sharp profile, SUNSET responds to a smoother curve which benefits its lifetime.

In order to compare the goodness of the proposed SUNSET Energy Optimization approach, a total of 339 simulation days are considered and energy savings are computed based on the energy price at each hour. A price penalty, i.e. double price, is included for energy purchases above the maximum contracted power ($Pot_{max} = 30$ Kw). Pot_{max} value has been set according to the characteristics of the employed Solar Thermal Power System (Sect. 2). Table 2 presents the overall results achieved by both strategies in terms of self-consumption, energy purchase, days that exceed the maximum contracted power and the total price of energy purchasing. As can be observed, RT and SUNSET provide the same self-consumption profile due to the prioritization of it in both schemes. However,

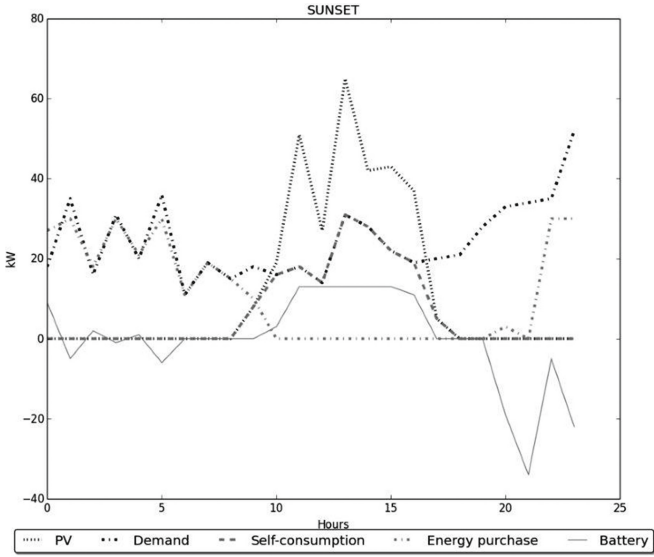


Fig. 5. Example of energy management for the SUNSET approach for a simulation day.

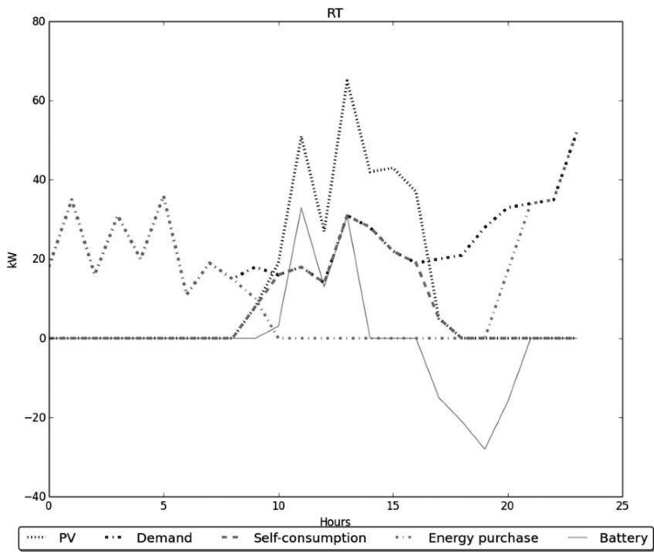


Fig. 6. Example of energy management for the RT approach for a simulation day.

Table 2. Simulation results for the RT strategy and the proposed SUNSET Energy Optimization approach for a total of 339 simulation days.

	Self-cons. (Kw)	Energy purchase (Kw)	Days > Pot_{max}	Price (€)
RT	50102	109155	321	13721
SUNSET	50102	105542	70	12103

the total price of energy that needs to be purchased in order to satisfy the demand is higher for the RT scheme. Similarly, the number of days that exceed Pot_{max} is approximately four times the SUNSET approach. Consequently, this results in a 12% of price saving by employing SUNSET approach instead of the RT strategy considering the energy prices in Table 3.

Table 3. Energy price per hour.

	0:00–7:00	8:00–17:00	18:00–21:00	22:00–23:00
Energy price (€)	0.091417	0.132440	0.167838	0.132440

4 Conclusions and Future Work

This paper presents a novel solar energy forecasting and optimization approach for optimally managing energy based on different strategies: self-consumption, energy purchase and battery energy storage for later consumption. The proposed SUNSET approach is tested in a real solar PV system plant in Zamudio (Spain) yielding to excellent results in terms of price minimization and battery usage enhancement.

Future work will be devoted to generalize the proposed SUNSET approach to other real scenarios and different countries' legislations in terms of energy use.

Acknowledgment. This work has been supported in part by the ELKARTEK program of the Basque Government (BID3ABI project), and EMAITEK funds granted by the same institution.

References

1. Golestaneh, F., Pinson, P., Gooi, H.B.: Very short-term nonparametric probabilistic forecasting of renewable energy generation-with application to solar energy. *IEEE Trans. Power Syst.* **31**(5), 3850–3863 (2016). <https://doi.org/10.1109/TPWRS.2015.2502423>
2. Boland, J., David, M., Lauret, P.: Short term solar radiation forecasting: Island versus continental sites. *Energy* **113**(15), 186–192 (2016)

3. Join, C., Fliess, M., Voyant, C., Chaxel, F.: Solar energy production: short-term forecasting and risk management. *IFAC (International Federation of Automatic Control)* **49**(12), 686–691 (2016)
4. Severiano, C., Guimaraes, F.G., Cohen, M.W.: Very short-term solar forecasting using multi-agent system based on extreme learning machines and data clustering. In: *IEEE Symposium Series on Computational Intelligence (SSCI)*, Athens (2016). <https://doi.org/10.1109/SSCI.2016.7850162>
5. Bacher, P., Madsen, H., Nielsen, H.A.: Online short-term solar power forecasting. *Solar Energy* **83**(10), 1772–1783 (2009)
6. Chen, C., Duan, S., Cai, T., Liu, B.: Online 24-h solar power forecasting based on weather type classification using artificial neural network. *Solar Energy* **85**(11), 2856–2870 (2011)
7. Hocaoglu, F.O., Gerek, Ö.N., Kurban, M.: Hourly solar radiation forecasting using optimal coefficient 2-D linear filters and feed-forward neural networks. *Solar Energy* **82**(8), 714–726 (2008)
8. 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)
9. Pedro, H.T., Coimbra, C.F.: Assessment of forecasting techniques for solar power production with no exogenous inputs. *Solar Energy* **86**(7), 2017–2028 (2012)
10. Cao, J., Lin, X.: Study of hourly and daily solar irradiation forecast using diagonal recurrent wavelet neural networks. *Energy Convers. Manag.* **49**(6), 1396–1406 (2008)
11. Mellit, A., Pavan, A.M.: A 24-h forecast of solar irradiance using artificial neural network: application for performance prediction of a grid-connected PV plant at Trieste, Italy. *Solar Energy* **84**(5), 807–821 (2010)
12. Mulder, G., De Ridder, F., Six, D.: Electricity storage for grid-connected household dwellings with PV panels. *Solar energy* **84**(7), 1284–1293 (2010)
13. Luthander, R., Widén, J., Nilsson, D., Palm, J.: Photovoltaic self-consumption in buildings: a review. *Appl. Energy* **142**, 80–94 (2015)
14. Widén, J.: Improved photovoltaic self-consumption with appliance scheduling in 200 single-family buildings. *Appl. Energy* **126**, 199–212 (2014)
15. Agnetis, A., de Pascale, G., Detti, P., Vicino, A.: Load scheduling for household energy consumption optimization. *IEEE Trans. Smart Grid* **4**(4), 2364–2373 (2013)
16. Chen, X., Wei, T., Hu, S.: Uncertainty-aware household appliance scheduling considering dynamic electricity pricing in smart home. *IEEE Trans. Smart Grid* **4**(2), 932–941 (2013)
17. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>

Gradient Boosting Models for Photovoltaic Power Estimation Under Partial Shading Conditions

Nikolaos Nikolaou¹(✉), Efstratios Batzelis², and Gavin Brown¹

¹ School of Computer Science, University of Manchester, Kilburn Building,
Oxford Road, Manchester M13 9PL, UK

{nikolaos.nikolaou,gavin.brown}@manchester.ac.uk

² Department of Electrical and Electronic Engineering, Imperial College London,
Exhibition Road, London SW7 2AZ, UK
e.batzelis@imperial.ac.uk

Abstract. The energy yield estimation of a photovoltaic (PV) system operating under partially shaded conditions is a challenging task and a very active area of research. In this paper, we attack this problem with the aid of machine learning techniques. Using data simulated by the equivalent circuit of a PV string operating under partial shading, we train and evaluate three different gradient boosted regression tree models to predict the global maximum power point (MPP). Our results show that all three approaches improve upon the state-of-the-art closed-form estimates, in terms of both average and worst-case performance. Moreover, we show that even a small number of training examples is sufficient to achieve improved global MPP estimation. The methods proposed are fast to train and deploy and allow for further improvements in performance should more computational resources be available.

Keywords: Gradient boosting · Solar energy · Photovoltaic (PV) system · Maximum power point (MPP) · Partial shading · Machine learning

1 Introduction

The photovoltaic (PV) penetration has remarkably increased worldwide the last decades, with several applications ranging from rooftop and building-integrated systems to MW-scale power plants. Especially in the former cases installed in urban environments, the operating conditions are often non-ideal with surrounding obstacles casting shadows on the PV system, leading to non-uniform illumination. Under such conditions, the *power-voltage (P-V) characteristic curve* presents several local *maximum-power-points (MPPs)*, a situation that hinders the effective tracking of the *global MPP* which provides the maximum power output. This phenomenon, commonly referred to as *partial shading*, has gathered the interest of researchers lately due to its non-linear nature and strong effect on the energy yield of the PV system.

There are several PV models in the literature, varying in terms of accuracy, complexity and scope of application, classified into two generic categories: the *circuit-based models* and the *heuristic methods*. The former models have strong theoretical foundation and can provide any needed information, but require tedious simulations of complicated circuits. The latter approaches, on the other hand, are simpler and provide directly the global MPP, but generally suffer from lower accuracy [2]. In PV energy yield studies, there is a need for a fast and reliable method to easily calculate the maximum power of the PV system at numerous different scenarios; in these applications, the heuristic alternatives seem more appropriate.

According to [2], these methods can be further classified to: (a) *Empirical efficiency-based models* which are derived from empirical observations and have simple formulation, but exhibit moderate accuracy due to their weak theoretical background [4, 8, 18]; (b) *Explicit mathematical equations* that are based on the equivalent circuit and provide all local MPPs, presenting good average accuracy in principle, yet occasionally high estimation errors [3, 13, 17]; and (c) *Artificial Neural Networks (ANN)* trained on the actual data of the study-case PV system which provide adequately fast execution and probably the best estimations [9, 14].

Even though several applications of machine learning algorithms are reported in the literature for the simple uniform illumination case [10], the relevant research for partial shading conditions is still limited to only the two aforementioned studies [9, 14]. As an alternative, we investigate in this paper *gradient boosting* [11, 12] models trained on data generated from the equivalent circuit. In the following, three gradient boosting models are implemented and evaluated across a wide range of operating conditions, concluding to very interesting and promising results. This is the first paper in the literature to apply this method in PV energy forecasting under partial shading.

2 PV Power Generation Under Partial Shading

2.1 Main Concepts and Examples

The smallest commercially available PV unit is the *module* (or *panel*). Usually several PV modules are connected in series to form a *string* in order to produce appropriate levels of *voltage* and *power* output. A typical PV string composed by 12 modules is depicted in Fig. 1 [TOP], operating under partial shading conditions (common case of two different *irradiance* levels). Based on the notation of [1, 3], 10 out of 12 modules are unshaded and illuminated at full irradiance $G = 800 \text{ W/m}^2$ (or 0.8 *per unit (pu)*), whereas the 2 remaining shaded modules are subject to half the irradiance (*shading ratio* $s = 50\%$); this corresponds to a *shadow extent* of $n_{sh} = 2/12 = 0.17$. Common *temperature* $T = 45^\circ\text{C}$ is assumed across the entire string, as usually the temperature difference between the shaded and unshaded part is small [3]. In the general case, the *operating conditions* of a PV string partially shaded at two different irradiance levels are

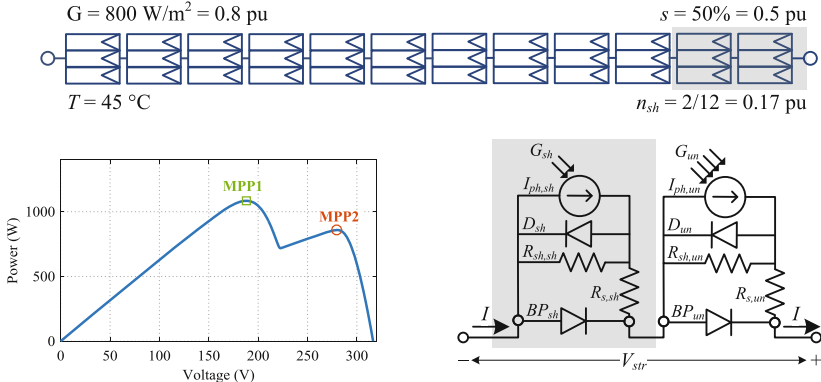


Fig. 1. [TOP] Topology of a PV string consisting of 12 PV modules connected in series, 2 of them being shaded at 50%. [BOTTOM-LEFT] Respective P-V curve indicating the 2 local maxima: MPP1 and MPP2. [BOTTOM-RIGHT] Electrical equivalent circuit.

uniquely identified by the vector $[G, T, s, n_{sh}]$, which in this example equals to $[0.8, 45, 0.5, 0.17]$.

The P-V characteristic at partial shading conditions presents up to two local MPPs (or power peaks), as illustrated in Fig. 1 [BOTTOM-LEFT] for this particular scenario. These two MPPs are studied and characterized as *MPP1* and *MPP2* in [3, 16], as they exhibit different behaviour and dependence on the operating conditions. Only one of them or both (most often) may appear and the power peak that provides the maximum power is denoted as the *global MPP*. This corresponds to either MPP1 or MPP2 depending on the scenario (MPP1 in Fig. 1), so determining its power P_{max} and voltage $V_{P_{max}}$ is not a trivial task.

To generate this P-V curve, one has to adopt an *electrical equivalent circuit*, modelled and simulated in appropriate software (e.g. MATLAB/Simulink). According to the PV modelling theory, the equivalent circuit of this system is described by Fig. 1 [BOTTOM-RIGHT] [1, 3]. Each part of the string, shaded and unshaded, is modelled using the single-diode equivalent and a bypass diode, the respective circuit parameters being 11 in total and strongly dependent on the operating conditions [1]. Calculating the maximum power according to the circuit-based models, involves the laborious steps of (i) extracting these parameters at STC^1 , (ii) translating their values to the actual conditions and (iii) simulating a highly non-linear circuit. These procedures are tedious and time-consuming, thus they are more suited for research, rather than practical applications [2].

2.2 Closed-Form Equations

Among the heuristic methods mentioned in the Introduction, the closed-form equations introduced in [3] and further improved in [17] are considered here, as

¹ *Standard Test Conditions*: 1000 W/m^2 irradiance, 25°C temperature & 1.5 air mass.

they are shown to outperform other explicit mathematical approaches. These are probably the best candidates to be compared against the machine learning models investigated in this paper. For completeness, these expressions are given in Eq. (1) –more details can be found in [3, 17]. Given the conditions [G, T, s, n_{sh}], the *system structural characteristics* (PV module datasheet information V_{mp0} , I_{mp0} , V_{oc0} , $\alpha_{I_{mp}}$, $\beta_{V_{mp}}$, $\beta_{V_{oc}}$, total number of modules N_{tot} , voltage drop on the bypass diode $\Delta V_D = 1.0$ V) and the empirical coefficient $\lambda = 0.06$:

$$\begin{aligned}
 MPP1 : & \begin{cases} V_1 = N_{tot}[(1 - n_{sh})V_{mp}^T + n_{sh}\Delta V_D] \\ I_1 = GI_{mp}^T \\ P_1 = V_1 I_1 \end{cases} \\
 MPP2 : & \begin{cases} V_2 = N_{tot}[(1 - n_{sh})(sV_{mp}^T + (1 - s)V_{oc}^T) + n_{sh}V_{mp}^T] \\ I_2 = sI_{mp}^T[1 + \lambda(1 - n_{sh})] \\ P_2 = V_2 I_2 \end{cases}
 \end{aligned} \tag{1}$$

where $I_{mp}^T = I_{mp0}[1 + \alpha_{I_{mp}}(T - 25)]$, $V_{mp}^T = V_{mp0}[1 + \beta_{V_{mp}}(T - 25)]$ and $V_{oc}^T = V_{oc0}[1 + \beta_{V_{oc}}(T - 25)]$. Although these expressions yield in general fairly acceptable average accuracy, in some cases they exhibit high errors, as observed in [3] and further verified in this paper. This motivates an alternative approach with machine learning models, which, once trained, could give a better estimate of the global MPP in a similarly straightforward and cost-efficient way.

3 Gradient Boosting Models for MPP Prediction

We simulate the behaviour of the circuit of Fig. 1 generating examples of input [G, T, s, n_{sh}] and output [$P_1, V_1, P_2, V_2, P_{max}, V_{P_{max}}$] vector pairs –note that we also include the ‘intermediate’ outputs of the local MPP1 & MPP2 power-voltage pairs, (P_1, V_1) & (P_2, V_2) , respectively. Using these examples, we train regression models to predict the values P_{max} & $V_{P_{max}}$. The goal is to better approximate these quantities than the closed-form estimates of Eq. (1).

In machine learning terminology, this is a *multi-output regression problem*. Normally, we would like to take into account the *covariance* of the targets in our model. In this work, we start with the simpler approach of modelling the individual targets *independently*², i.e. we train *multiple single-output regressors*.

There are many possible choices of regression algorithms to use. As a reliable off-the-shelf learning meta-algorithm, *gradient boosting* [11, 12] was chosen, as its variants have proven very successful in large scale experimental comparisons of learning algorithms [6, 10], industrial applications [5, 19] and competitions [7] alike –often outperforming other powerful *kernel-based* or *deep learning* methods.

More specifically, deep learning methods tend to be outperformed by gradient boosting on tabular data –as is the case here. The problem here is not expected to

² Initial attempts at exploiting interactions between P_{max} & $V_{P_{max}}$ by first predicting the value of one and then using it to predict the other, yielded worse results than assuming independence. Their further investigation is left for future work.

be particularly noisy; another reason to expect boosting to be good at modelling it. Compared to kernel-based methods, like *Support Vector Machines (SVMs)*, that shift the computational burden from the number of features to the number of examples, gradient boosting is computationally preferable in our task, since the feature space is considerably smaller than the number of examples.

Furthermore, gradient boosting constructs models that can be more easily interpreted than those of the aforementioned methods. For example, allowing for better approximations to Eq. (1) to be derived as closed-form formulas. Finally and perhaps more importantly, gradient boosting has relatively few *hyperparameters* to tune compared to the aforementioned methods, the most important being the *ensemble size* M and the *complexity* of its *base learner*. Very recent research [20] suggests that the higher both of these are, the better the ensemble’s generalization behaviour (although the computational cost increases).

Gradient boosting constructs an ensemble of M *additive components* (base learners) in a forward stagewise manner; it allows for the optimization of arbitrary differentiable *loss functions* (here *quadratic loss*). In each stage a base learner (here *regression tree*) is fit on the negative gradient of the given loss function. The final output, i.e. the *estimate* \hat{X} of the target variable in question³ in each case, will be a *weighted linear combination* $\hat{X} = \sum_{m=1}^M a_m f_m(G, T, s, n_{sh})$ of the M base learners’ outputs, each of which –being a regression tree– is a *piecewise linear function* $f_m(G, T, s, n_{sh})$ of the inputs. The *weights* a_m and the additive components f_m are learned from the data.

In the remainder of this section, we shall discuss the three different models we will compare against the closed-form estimation. The models will differ in what target outputs \hat{X} they estimate, and how they combine them to predict the global MPP ($P_{max}, V_{P_{max}}$).

3.1 Direct Modelling of MPP

First we take a direct approach to learning a model for predicting the global MPP ($P_{max}, V_{P_{max}}$). We simply train a regressor to map input vectors $[G, T, s, n_{sh}]$ to P_{max} and another to map them to $V_{P_{max}}$ independently, without making use of the local MPP1 and MPP2. We will henceforth refer to this as the ‘*Direct*’ approach. Note that as the two constituent regression tasks are independent, they can be parallelized to reduce the computational cost.

3.2 Stagewise Modelling of MPP

Next we take a 2-stage approach for predicting the global MPP. We train four regressors: one to map input vectors $[G, T, s, n_{sh}]$ to each of the ‘intermediate outputs’ P_1, V_1, P_2 & V_2 . In other words, we first model the MPP1 and the MPP2 independently (and in each case P_i independently of its corresponding V_i). Then, we predict P_{max} & $V_{P_{max}}$ by

³ As we will see in this section, \hat{X} can be an estimate of P_{max} or $V_{P_{max}}$, or of any of the ‘intermediate outputs’ – P_1, V_1, P_2, V_2 – depending on the method.

$$\text{global MPP} : \begin{cases} P_{max} = \max\{P_1, P_2\} \\ V_{P_{max}} = \{V_{i^*} : i^* = \arg \max_{i \in \{1,2\}} P_i\}. \end{cases} \quad (2)$$

Note that in some scenarios a single power peak appears (MPP1 or MPP2), rather than both. In these cases, we disregard these examples for the purpose of training models to predict P_2 & V_2 or P_1 & V_1 , respectively. By this approach, which we shall call ‘*Stagewise*’, we encode more domain knowledge in our model about the structure of the targets: there exists at least one MPP and at most two, each of which has different characteristics and the global MPP is the maximum of the existing ones. Again, since the four constituent regression tasks are independent, they can be parallelized to reduce computational costs.

3.3 Classifier-Assisted Stagewise Modelling of MPP

As we will see in the experimental section, the *Stagewise* approach achieves very low error on the intermediate estimates of P_1 , V_1 , P_2 & V_2 and also on P_{max} . Yet, its estimates of $V_{P_{max}}$ –even though much better than those of both the closed-form ones and the *Direct* ones– seem to not be on par with those of all other target quantities. This seems to stem from situations in which P_1 & P_2 are very close. When this happens, a small estimation error on either of these can lead us to select the wrong P_i as the P_{max} . The resulting error on P_{max} will still be small (the values of P_1 & P_2 being very close), but by choosing the wrong V_i as the $V_{P_{max}}$, the error on $V_{P_{max}}$ is high.

Motivated by this, the final approach we shall examine is a variant of the *Stagewise* method, where the selection of the global MPP ($P_{max}, V_{P_{max}}$), is not based on the comparison of the predicted values of P_1 & P_2 , but is instead decided by a *binary classifier* trained to predict whether an input vector $[G, T, s, n_{sh}]$ leads to MPP1 or MPP2 being the global MPP. To get the label of each example, it is sufficient to compare if $P_1 > P_2$, in which case MPP1 is the global MPP, otherwise it is MPP2. Based on the classifier’s prediction, we pick the appropriate P_i & V_i pair calculated by the regressors as the P_{max} & $V_{P_{max}}$, respectively. The classification subtask is accomplished by training a gradient boosting ensemble of *decision trees*⁴. We call this method ‘*StagewiseC*’. As with the previous approaches, to reduce computational costs we can parallelize the training of all 5 predictors.

4 Experimental Investigation

4.1 Experimental Setup

We generated 94905 examples from the circuit shown in Fig. 1, considering irradiance $G = [0.1 : 0.05 : 1.0]$ pu, temperature $T = [-5 : 5 : 65]$ °C, shade ratio

⁴ Denoting MPP1 & MPP2 with ‘1’ & ‘-1’, respectively, the classifier’s prediction is of the form $\hat{H} = \text{sign}[\sum_{m=1}^M a_m h_m(G, T, s, n_{sh})] \in \{-1, 1\}$, where $h_m(G, T, s, n_{sh}) \in \{-1, 1\}$ is the prediction of the base learner added on the m -th round and a_m its voting weight, both the learner and a_m being the learned parameters of the model.

$s = [0.1 : 0.1 : 0.9]$ and shade extent $n_{sh} = [0 : 1/36 : 1]$. The examples consist of input vector $[G, T, s, n_{sh}]$ and output vector $[P_1, V_1, P_2, V_2, P_{max}, V_{P_{max}}]$ pairs – where we also included the ‘intermediate’ outputs of MPP1 and MPP2. On each example, we also provide the closed-form estimate for P_1, V_1, P_2 & V_2 , along with the resulting final P_{max} and $V_{P_{max}}$ by evaluating Eq. (1). We compare the estimate of the learned models described in Sect. 3 to the closed-form ones. As ground truth, we consider the true outputs of the circuit.

All learners, unless otherwise specified, were gradient boosting ensembles of size $M = 1000$ trained on a 75% of the data chosen uniformly at random and evaluated on the remaining 25%. Regression trees were chosen as base learners for the regression subtasks, and for the classification step of *StagewiseC*, we used a decision tree. In all cases, the trees had a maximum depth of 3. All remaining hyperparameters were left to the default values of *scikit-learn*⁵.

To evaluate the quality of the estimates (both learned models’ and closed-form ones), for each target quantity X , we calculate on the test set the *normalized root-mean squared error* of the predictions \hat{X} ,

$$NRMSE = \sqrt{\sum_n (X_n - \hat{X}_n)^2} / \mu_X,$$

that uses as normalization factor the mean of quantity X on the test set, μ_X .

To get a more robust measure of central tendency than the NRMSE, we also provide the *median* of the *normalized absolute error* on each example,

$$NAE_n = |X_n - \hat{X}_n| / \mu_X.$$

We also report the *maximum* NAE as a measure of worst-case performance and *percentiles* of the NAE (95th, 99th, 99.9th & 99.99th), the purpose of which is to allow us to observe the frequency of large estimation errors.

We perform 10 train-test splits, shuffling the dataset before each split and report averages and 95% *confidence intervals* for all evaluation measures estimated. Finally, we also provide some characteristic *learning curves* to demonstrate how the quality of estimates of each model changes with the size of the training set.

4.2 Experimental Results

Results for Intermediate Problems. As we saw, the two stagewise models (*Stagewise* & *StagewiseC*) use as intermediate steps the predictions of four regressors (predicting P_1, V_1, P_2 & V_2). In addition, *StagewiseC* predicts whether the global MPP is MPP1 or MPP2 by the use of a classifier. We first present results that showcase the predictive performance of these learners, before moving on to the results on the final P_{max} & $V_{P_{max}}$ predictions. On Table 1 we compare the approximation on P_1 & P_2 of the two stagewise models to that of the closed-form estimates of Eq. (1). On Table 2 we do the same for V_1 & V_2 .

⁵ <http://scikit-learn.org/stable/>.

Table 1. Statistics of the errors of P_1 & P_2 estimates under the closed-form solutions and the two stagewise models. Averages and 95% CIs across 10 runs shown. Results rounded to four decimal places. Lowest values shown in bold.

	P_1		P_2	
	<i>Closed form</i>	Learned model	<i>Closed form</i>	Learned model
NRMSE	0.0388 ± 0.0006	0.0120 ± 0.0002	0.0475 ± 0.0002	0.0098 ± 0.0002
Median NAE	0.0084 ± 0.0000	0.0071 ± 0.0002	0.0214 ± 0.0001	0.0061 ± 0.0001
95th %ile NAE	0.0394 ± 0.0002	0.0244 ± 0.0005	0.1065 ± 0.0007	0.0198 ± 0.0004
99th %ile NAE	0.1413 ± 0.0066	0.0350 ± 0.0012	0.1659 ± 0.0009	0.0275 ± 0.0005
99.9th %ile NAE	0.5193 ± 0.0117	0.0500 ± 0.0010	0.2207 ± 0.0021	0.0365 ± 0.0007
99.99th %ile NAE	0.6608 ± 0.0081	0.0693 ± 0.0063	0.2465 ± 0.0021	0.0440 ± 0.0021
Maximum NAE	0.7171 ± 0.0241	0.0812 ± 0.0084	0.2551 ± 0.0027	0.0464 ± 0.0023

Table 2. Statistics of the errors of V_1 & V_2 estimates under the closed-form solutions and the two stagewise models. Averages and 95% CIs across 10 runs shown. Results rounded to four decimal places. Lowest values shown in bold.

	V_1		V_2	
	<i>Closed form</i>	Learned model	<i>Closed form</i>	Learned model
NRMSE	0.0478 ± 0.0007	0.0041 ± 0.0001	0.0352 ± 0.0001	0.0047 ± 0.0000
Median NAE	0.0127 ± 0.0000	0.0020 ± 0.0000	0.0147 ± 0.0001	0.0020 ± 0.0000
95th %ile NAE	0.0342 ± 0.0001	0.0076 ± 0.0002	0.0797 ± 0.0005	0.0094 ± 0.0001
99th %ile NAE	0.2554 ± 0.0078	0.0118 ± 0.0004	0.1158 ± 0.0008	0.0174 ± 0.0003
99.9th %ile NAE	0.5403 ± 0.0063	0.0269 ± 0.0038	0.1398 ± 0.0004	0.0317 ± 0.0009
99.99th %ile NAE	0.6326 ± 0.0046	0.0756 ± 0.0055	0.1741 ± 0.0112	0.0565 ± 0.0066
Maximum NAE	0.6596 ± 0.0062	0.0880 ± 0.0036	0.2101 ± 0.0081	0.0797 ± 0.0079

As we can see, the learned model estimates for each of the four quantities are considerably better than the closed-form ones. This is not only true as a central tendency (lower NRMSE, lower median NAE), but also holds for the worst case (lower maximum NAE). Inspecting the NAE percentile results given, we can also conclude that only a very small number of the errors on the estimates are high.

Next, on Table 3, we can inspect the confusion matrix of the classifier used by *StagewiseC* for predicting whether the global MPP is MPP1 or MPP2. We can see that it is very accurate on both classes.

Results for Final Predictions. We can expect that these results on the intermediate prediction tasks will translate to better estimates than the closed-form methods for P_{max} & $V_{P_{max}}$ under the stagewise models. Next, in Tables 4 and 5 we present the results of each model for predicting the global MPP.

Dependence on Training Set Size. We saw that the learned models outperform the closed form estimates. But how many examples do the learners need to

Table 3. Confusion matrix for the classifier used by *StagewiseC* for predicting whether the global MPP corresponds to MPP1 or MPP2. Averages and 95% CIs across 10 runs shown. Results rounded to one decimal place. The classifier is very accurate on both classes, having an average accuracy of 99.74% with an average F_1 -score of 0.998 and very low variance across runs. The data is slightly skewed towards MPP1 being the global MPP in approximately 58.92% of the cases.

		Prediction	
		MPP1	MPP2
Truth	MPP1	13945.5 \pm 24.8	33.5 \pm 2.2
	MPP2	27.8 \pm 3.6	9720.2 \pm 24.3

see, before they can do so? In Fig. 2, we provide learning curves for the NRMSE on the final P_{max} & $V_{P_{max}}$ estimates. We conclude that we do not need to use a large training set to exceed the quality of the closed-form estimates. We can also see that beyond –say 5000– training examples, the increase in training set size only marginally improves the quality of the estimates for the *Direct* & *Stagewise* estimators. *StagewiseC* is an exception in two ways: (i) it needs a larger sample size for its estimates of P_{max} to be comparable to the closed-form ones, (ii) after seeing about 10000 training examples, it is already outperforming all other approaches in terms $V_{P_{max}}$ estimation, yet it can take advantage of a larger sample size to produce even better estimates.

4.3 Analysis of the Results

As discussed, all trained models examined produced better estimates than the closed-form ones. Indicatively, inspecting Tables 4 and 5, we find that the reduction in NRMSE over the closed-form estimate for the *Stagewise* model’s estimates in P_{max} is about 69%. Similarly, the reduction in NRMSE over the closed-form estimate for the *StagewiseC* model’s estimates in $V_{P_{max}}$ is about 52%. The results are similar when the goal is not just good average performance, but also to minimize the maximum prediction error: *Stagewise* reduces the closed-form estimates’ maximum NAE in P_{max} by about 59% and *Direct* reduces the closed-form estimates’ maximum NAE in $V_{P_{max}}$ by about 21%. On the intermediate estimates of Tables 1 and 2, we saw that the relative improvement of the learned model estimates over the closed-form estimates is even more impressive, for every quantity and every evaluation measure examined. In short, we get both lowest average and lowest maximum errors by the learned models.

We attributed the relatively large errors in the estimates (both closed-form & learned) of $V_{P_{max}}$ to cases in which P_1 & P_2 are close, and a small error in either results to choosing the wrong global MPP (V_1 & V_2 may significantly differ even when $P_1 \approx P_2$). The models we discussed tackle this issue in different ways: *Direct* models $V_{P_{max}}$ directly, without separately modelling the two MPPs. *Stagewise* estimates P_1 & P_2 with low error –and so does the corresponding V_i

Table 4. Statistics of the errors of P_{max} estimates under the different models. Averages and 95% CIs across 10 runs shown. Results rounded to four decimal places. Lowest –and tied for lowest, in the sense of overlapping CIs– values shown in bold.

	<i>Closed form</i>	Learned models		
		<i>Direct</i>	<i>Stagewise</i>	<i>StagewiseC</i>
NRMSE	0.0300 ± 0.0001	0.0213 ± 0.0003	0.0093 ± 0.0001	0.0308 ± 0.0044
Median NAE	0.0159 ± 0.0001	0.0126 ± 0.0002	0.0056 ± 0.0001	0.0056 ± 0.0001
95th %ile NAE	0.0645 ± 0.0004	0.0431 ± 0.0007	0.0189 ± 0.0003	0.0189 ± 0.0003
99th %ile NAE	0.1038 ± 0.0006	0.0646 ± 0.0009	0.0267 ± 0.0007	0.0269 ± 0.0007
99.9th %ile NAE	0.1369 ± 0.0014	0.0952 ± 0.0025	0.0387 ± 0.0009	0.0403 ± 0.0009
99.99th %ile NAE	0.1530 ± 0.0025	0.1229 ± 0.0080	0.0548 ± 0.0045	1.8218 ± 0.4035
Maximum NAE	0.1616 ± 0.0029	0.1303 ± 0.0080	0.0662 ± 0.0069	2.4909 ± 0.1408

Table 5. Statistics of the errors of $V_{P_{max}}$ estimates under the different models. Averages and 95% CIs across 10 runs shown. Results rounded to four decimal places. Lowest –and tied for lowest, in the sense of overlapping CIs– values shown in bold.

	<i>Closed form</i>	Learned models		
		<i>Direct</i>	<i>Stagewise</i>	<i>StagewiseC</i>
NRMSE	0.0697 ± 0.0013	0.0597 ± 0.0005	0.0507 ± 0.0013	0.0333 ± 0.0016
Median NAE	0.0098 ± 0.0000	0.0121 ± 0.0002	0.0017 ± 0.0000	0.0017 ± 0.0000
95th %ile NAE	0.0735 ± 0.0006	0.1111 ± 0.0023	0.0075 ± 0.0001	0.0072 ± 0.0001
99th %ile NAE	0.2609 ± 0.0063	0.2548 ± 0.0051	0.0221 ± 0.0015	0.0146 ± 0.0002
99.9th %ile NAE	0.9623 ± 0.0048	0.6133 ± 0.0341	0.8539 ± 0.0137	0.6623 ± 0.0708
99.99th %ile NAE	1.1801 ± 0.0377	0.9041 ± 0.0431	1.0789 ± 0.0144	1.1334 ± 0.0228
Maximum NAE	1.2222 ± 0.0007	0.9603 ± 0.0192	1.1512 ± 0.0133	1.1934 ± 0.0173

it assigns as $V_{P_{max}}$. Finally, *StagewiseC* trains an accurate classifier to recognise the global MPP, circumventing the need to compare the estimates of P_1 & P_2 .

The percentile results provided, allow the reader to see that indeed, such large errors appear very rarely under the learned models. For instance, Table 4 shows that only one in 10000 (99.99th %ile) estimates of the *Stagewise* model for P_{max} is expected to have a NAE greater than 5.48%⁶, whereas one in 20 (95th %ile) of the closed-form estimates will have a NAE greater than 6.45%. Similarly, Table 5 shows that only one in 100 (99th %ile) estimates of the *StagewiseC* model for $V_{P_{max}}$ is expected to have a NAE greater than 1.46%, when the same number of closed-form estimates will have a NAE greater than 26.09%.

Which model is to be preferred in practice will depend on the exact guarantees we want for our system and the design limitations imposed. *Direct* produces the smallest maximum error in terms of $V_{P_{max}}$. *Stagewise* outperforms

⁶ More precisely, 95% of the times, 99.99% of the estimates of P_{max} under the *Stagewise* model will have a NAE smaller than some value that lies between 5.03% and 5.93%. In the discussion, we sacrifice this level of mathematical rigour for simplicity.

all other approaches in terms of estimation of P_{max} . *StagewiseC* produces the smallest number of high-value errors on $V_{P_{max}}$. To practitioners, we can propose ‘combining’ *Stagewise* & *StagewiseC*: train all 5 learners used by *StagewiseC*, use the classifier’s prediction for selecting the V_i for $V_{P_{max}}$, but ignore the classifier and directly compare the estimates of P_1 & P_2 to choose the P_{max} as in Eq. (2).

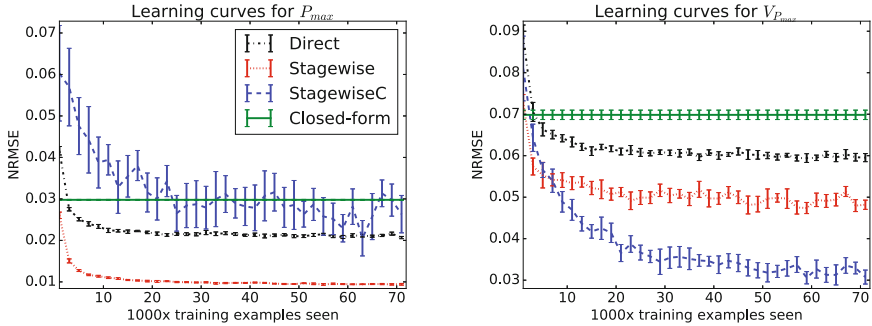


Fig. 2. Learning curves (NRMSE vs. training set size) on the final P_{max} [LEFT] & $V_{P_{max}}$ [RIGHT] estimates under each model. Averages and 95% CIs across 10 runs shown. We investigate training set sizes of 1000 up to the 75% of the total available examples (71178) taken at steps of 1000.

Finally, we also saw in Fig. 2 that even a small fraction of the available training examples (e.g. fewer than 5000) is more than enough for the learned models to considerably improve upon the closed-form estimates. Initial explorations of the ensemble size M and the depth of the trees –omitted due to space limitations– suggest that increasing either leads to better predictive performance for all models albeit at a higher computational cost.

5 Conclusion and Future Work

We introduced 3 gradient boosted tree models to perform PV power estimation under partial shading conditions. Our empirical results show that, compared to the current state-of-the-art closed-form estimates, the errors on the gradient boosting models’ predictions are smaller on average, in the worst case and also the large errors committed are fewer in number. Only a small number of training examples is needed to improve upon the closed-form estimates. Depending on the computational resources, we can opt to parallelize the training of the individual ensembles each model involves or to improve the quality of our estimation at the cost of increased training time by increasing the number or depth of the trees.

A next step would be to take into account the covariance among targets. A more detailed investigation of learning algorithms, hyperparameter optimization and analysis of the resulting computational cost/approximation tradeoffs is also left for future work. So is the derivation of a simpler, interpretable model to

replace the closed-form estimates. The skew in the global MPP distribution could be dealt with cost-sensitive adaptations of boosting [15], improving performance.

The same ideas could also be applied to more complicated scenarios, such as different PV configurations (arrays of modules connected in series-parallel, bridged-linked etc.), multiple irradiance levels where more than two MPPs appear or in the more realistic case of non-uniform temperature across the PV system.

Acknowledgements. This project was partially supported by the EPSRC Centre for Doctoral Training [EP/I028099/1] & the EPSRC LAMBDA [EP/N035127/1] & Anyscale Apps [EP/L000725/1] project grants. N. Nikolaou acknowledges the support of the EPSRC Doctoral Prize Fellowship. E. Batzelis carried out this research at NTUA, Athens, Greece under the support of the ‘IKY Fellowships of Excellence for Postgraduate Studies in Greece-Siemens Program’.

References

1. Batzelis, E., Kampitsis, G.E., Papathanassiou, S.A.: A MPPT algorithm for partial shading conditions employing curve fitting. In: EU PVSEC, pp. 1502–1507 (2016)
2. Batzelis, E.I., Georgilakis, P.S., Papathanassiou, S.A.: Energy models for photovoltaic systems under partial shading conditions: a comprehensive review. *IET Renew. Power Gener.* **9**(4), 340–349 (2015)
3. Batzelis, E.I., Routsolias, I.A., Papathanassiou, S.A.: An explicit PV string model based on the Lambert W function and simplified MPP expressions for operation under partial shading. *IEEE Trans. Sustain. Energy* **5**(1), 301–312 (2014)
4. Brecl, K., Topič, K., Topič, M.: Self-shading losses of fixed free-standing PV arrays. *Renew. Energy* **36**(11), 3211–3216 (2011)
5. Busa-Fekete, R., Kégl, B., Élterő, T., Szarvas, G.: Ranking by calibrated AdaBoost. In: Proceedings of the Learning to Rank Challenge, pp. 37–48 (2011)
6. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: ICML, pp. 161–168 (2006)
7. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system. In: SIGKDD, pp. 785–794 (2016)
8. Deline, C., Dobos, A., Janzou, S., Meydbray, J., Donovan, M.: A simplified model of uniform shading in large photovoltaic arrays. *Sol. Energy* **96**, 274–282 (2013)
9. Dolan, J.A., Lee, E., Yeh, C., Ben-Menahem, S., Ishihara, A.K.: Neural network estimation of photovoltaic IV curves under partially shaded conditions. In: IJCNN, pp. 1358–1365 (2011)
10. Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems? *JMLR* **15**, 3133–3181 (2014)
11. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* **29**, 1189–1232 (2000)
12. Mason, L., Baxter, J., Bartlett, P., Frean, M.: Boosting algorithms as gradient descent. In: NIPS, pp. 512–518. MIT Press (2000)
13. Moballeggh, S., Jiang, J.: Modeling, prediction, and experimental validations of power peaks of PV arrays under partial shading conditions. *Sustain. Energy* **5**(1), 293–300 (2014)

14. Nguyen, D.D., Lehman, B., Kamarthi, S.: Performance evaluation of solar photovoltaic arrays including shadow effects using neural network. *IEEE Energy Convers. Congr. Expo.* **6**(2), 3357–3362 (2009)
15. Nikolaou, N., Edakunni, N., Kull, M., Flach, P., Brown, G.: Cost-sensitive boosting algorithms: do we really need them? *Mach. Learn.* **104**(2), 359–384 (2016)
16. Psarros, G., Batzelis, E., Papathanassiou, S.: Analysis of local MPPs on the P-V curve of a partially shaded PV string. In: *EU PVSEC*, pp. 3383–3389 (2014)
17. Psarros, G.N., Batzelis, E.I., Papathanassiou, S.A.: Partial shading analysis of multistring PV arrays and derivation of simplified MPP expressions. *IEEE Trans. Sustain. Energy* **6**(2), 499–508 (2015)
18. Rodrigo, P., Fernández, F., Almonacid, F., Pérez-Higueras, J.: A simple accurate model for the calculation of shading power losses in photovoltaic generators. *Sol. Energy* **93**, 322–333 (2013)
19. Viola, P., Jones, M.J.: Robust real-time face detection. *IJCV* **57**(2), 137–154 (2004)
20. Wyner, A.J., Olson, M., Bleich, J., Mease, D.: Explaining the success of adaboost and random forests as interpolating classifiers (2017). [arXiv:1504.07676v2](https://arxiv.org/abs/1504.07676v2)

Multi-objective Optimization for Power Load Recommendation Considering User's Comfort

Jaroslav Loebel^(✉), Helmut Posch, and Viera Rozinajová

Fakulta Informatiky a Informačných Technológií, Slovenská Technická Univerzita
V Bratislave, Ilkovičova 4, 842 16 Bratislava 4, Slovakia
{jaroslav.loebel,viera.rozinajova}@stuba.sk, helmutposch4@gmail.com

Abstract. This paper proposes a recommendation system for load shifting of energy consumption for residential consumers. The main goal is to provide to customer a set of energy consumption strategies, which would span from maximum cost saving strategy, to maximum comfort preserving strategy. The discomfort of user caused by load shifting is expressed here as a Euclidean distance between recommended and forecasted consumption. Recommendation is formulated as a multi-objective optimization problem, solved by NSGA-II (non-dominated sorting genetic algorithm II). Evaluation of proposed method is carried out on data from Pecan Street [1], which were preprocessed and aggregated, to form a typical consumption and photovoltaic (PV) generation course for winter and summer day. Albeit no batteries are present in original dataset, we also consider employing the batteries for storing PV generated spare power, with simple heuristics to control charging and discharging the batteries.

Keywords: Load shifting · Multi-objective optimization · NSGA-II

1 Introduction

Load shifting as a part of various demand response programs is a subject of intensive research, both from perspective of utility companies, aiming to reduce the peaks in electricity usage, and from perspective of end users trying to utilize dynamic pricing programs. With emerge of smart and microgrid, incorporating renewable sources and energy storage systems, this issue became even more interesting. This is due to the fact, that this environment provides more means to reduce energy cost.

Smart grid generates a huge amount of data. However, the real value can be obtained only when these datasets will be properly processed. Data analytics offers us means to enhance intelligence of the grid, e.g. consumer clustering, cost and power load optimization or power load prediction are frequently used [6]. One of the main goals of energy consumer is to save on energy expenditure while preserving his demands as much as possible. This task is sometimes quite difficult. We propose a recommendation system based on optimization using

NSGA-II algorithm, whereby we take into consideration not only the final energy cost, but also the comfort of the user.

2 Related Work

Authors of [4] proposed a load scheduling by appliance commitment, namely electrical water heater. User comfort is expressed as linear inequality constraint (for water temperature), which is specified by the user. The optimization objective is to minimize the cost for energy consumption. Price signals were forecasted by adding a white noise to actual historical data from Pacific Northwest Grid-Wise Testbed project [5]. Mixed integer linear programming (MILP) model was formulated for load scheduling in [2], where objective functions expressed cost minimization, maximization of scheduling preferences (where explicitly specified time slots for shiftable loads are required) and maximization of climatic comfort. Three objective functions are joined into one single-objective optimization problem, using normalization and weighting method, for which user can specify his/her preferences by assigning weights to each objective. MILP was then solved using custom heuristic algorithm. Multi-objective nature of load scheduling problem was preserved in [10] and was solved by modified version of NSGA-II algorithm. Modeled optimization problem consisted of two objective functions – one determining the cost for the energy consumption and one determining the end user dissatisfaction. End user must specify preferred time slots for loads, in which shifting may occur. Dissatisfaction is then evaluated as the risk of interruption of energy supply to shiftable load and violating the specified time slots.

Home energy management problem joined with electric vehicle charging was tackled in [8]. The discomfort was formulated as deviation of house temperature from de-sired value, specified by user. The discomfort objective function along with total electricity cost was transformed to single objective optimization problem by simple scalarization.

As can be seen, current methods need user input for expressing the comfort in the optimization problem, or state the comfort as a simple constraint. We propose a multi-objective optimization model employing a Euclidean distance metric, which expresses the user comfort violation without need of his explicit input. Preserving the multi-objective nature of model permits the trade-off exploration of possible strategies.

3 Formulation of Recommendation Model

Our approach assumes that the recommendation of energy consumption schedule is formulated as a multi-objective optimization problem. At first, model without battery is created, addition of battery with simple charge and discharge heuristics is discussed later.

$$\begin{aligned} & \text{minimize } F(X) = (\text{cost}(X), \text{Eucl}(X), \text{diff}(X), \text{smooth}(X))^T & (1) \\ & \text{subject to } x_i \geq 0, \forall i \in \{1, \dots, 0\} & (2) \end{aligned}$$

where x_i is recommended energy consumption at time i , together forming vector X of size N , which is number of time intervals during one day (24 in our case). One simple constraint tells us, that the energy consumption at time i cannot be negative. The $cost(X)$ function is calculating the total price for energy consumed that day, which can be simply formulated as:

$$cost(X) = \sum_{i=1}^N \min(x_i - pv_i, 0) * p_i \quad (3)$$

where pv_i is predicted amount of energy produced from photovoltaics at time i , p_i is price of energy from grid at time i . Prices vary during the day and the value for particular time i depends on the fact, whether it is on/off-peak hour and summer or a winter – actual values are taken from Pacific Power pricing [9] and can be seen in Table 1. In this simplified model, any excessive amount of energy produced from photovoltaics is discarded.

The $Eucl(X)$ expresses Euclidean distance between recommended and predicted consumption and we use it here as comfort indicator. It is based on assumption, that maximum comfort for a user is achieved, when one doesn't need to shift any load, hence the energy consumption is same as predicted one – the distance is zero. Therefore, by minimization of $Eucl(X)$ we maximize the comfort of user.

Next objective function is also a constraint – since the cheapest solution would be not consuming any energy at all, we must ensure, that the total amount of recommended energy will be same, as total amount of predicted energy. This can be done by minimizing the $diff(X)$ function, which is defined as follows:

$$diff(X) = \left| \sum_{i=1}^N x_i - \sum_{i=1}^N a_i \right| \quad (4)$$

Last objective function ($smooth(X)$) aims to minimize large jumps in amount of consumed energy (i.e. the time series is smooth). It is computed as a standard deviation of first order difference of consumption.

This mathematical formulation does not consider battery. In battery considering model, we employ simple heuristics for charging and discharging – when there is surplus amount of energy generated from photovoltaics, we use it for charging and whenever there is energy in battery, we use it to satisfy energy demand. The most widely used batteries for storing the energy from photovoltaics are Li-ion batteries [7]. Charging effectivity of the chosen type is 85% and maximum rate of charging and discharging is 5E, which means that the battery can be fully charged and discharged in 12 min. Since our model employs one hour time steps, we just assume, that battery can be fully charged/discharged in one time step.

4 Evaluation

Optimization model formulated in Chapter 3 was optimized by NSGA-II algorithm [3]. Our goal is to provide a set of recommendations, so the end user can

Table 1. Prices per kWh for on and off peak hours for summer and winter. On peak hours during summer are between 4 pm and 8 pm. On-peak hours during winter are between 6 am and 10 am, and 5 pm and 8 pm. Weekends are always considered as off-peak hours.

Summer		Winter	
Off peak	On peak	Off peak	On Peak
0.03875\$	0.011124\$	0.03875\$	0.08316\$

explore the tradeoff between preserving comfort and shifting load to save cost. The filtered set of Pareto-optimal solutions (i.e. no solution can be improved without worsening the other) is presented to user serving as a guidance for energy consumption on next day.

In our experiment, we considered three settings: household without battery, house-hold with battery with 3 kWh capacity and household with 5 kWh battery capacity. The recommendation is carried out for typical day of summer (April to October) and winter (November to March), which was calculated as an average of all users for given period from Pecan Street dataset [1]. Data for typical day consist of consumed energy and generated energy from photovoltaics and can be seen in Fig. 1. These values serve as an input for model. The energy cost user would pay for these average days is 0.80\$ for winter day and 1.27\$ for summer day. The optimization with NSGA-II algorithm was performed with 200 individuals and 90% crossover probability rate. Number of generations was empirically set to 600. As can be seen in lower right plot in Fig. 1, the growth of hypervolume resembles logarithmic growth, with values dramatically increasing during the first 100 generations and slowly decreasing growth afterwards. The reference point for hypervolume was chosen as the worst objective values from initial population. For evaluation, we considered only solutions, which satisfy the filtration criteria ($diff(X) \leq 0.5$, $Eucl(X) \leq 10$ and $smooth(X) \leq 1$) and we pick the population with highest hypervolume score. Tables 2 and 3 show cost of energy for average winter and summer day, respectively, for consumption strategy, which yields maximum cost savings. Populations with best hypervolume among 300 generations and 600 generations were considered. The best results are (as expected) obtained when employing the large 5 kWh battery, however, for household with no battery, rather significant saving is also achieved. More interestingly, the larger number of generations did not bring improvement in cost saving. This is a consequence of employing the $diff(X)$ and $smooth(X)$ constraints as objective functions, since the hypervolume is computed from all objective functions and not only cost and distance.

Figure 2 shows the visualization of optimization results for average winter day. Upper left plot show all filtered solutions. Marginal solutions (the lowest cost for energy and the lowest distance) are shown in upper right plot. The lower left plot shows the scatter plot of filtered solutions, which (along with consumption profiles) are presented to the user.

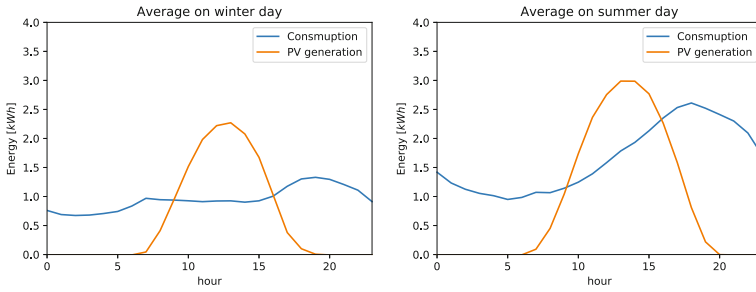


Fig. 1. Average consumption and PV generation of all users

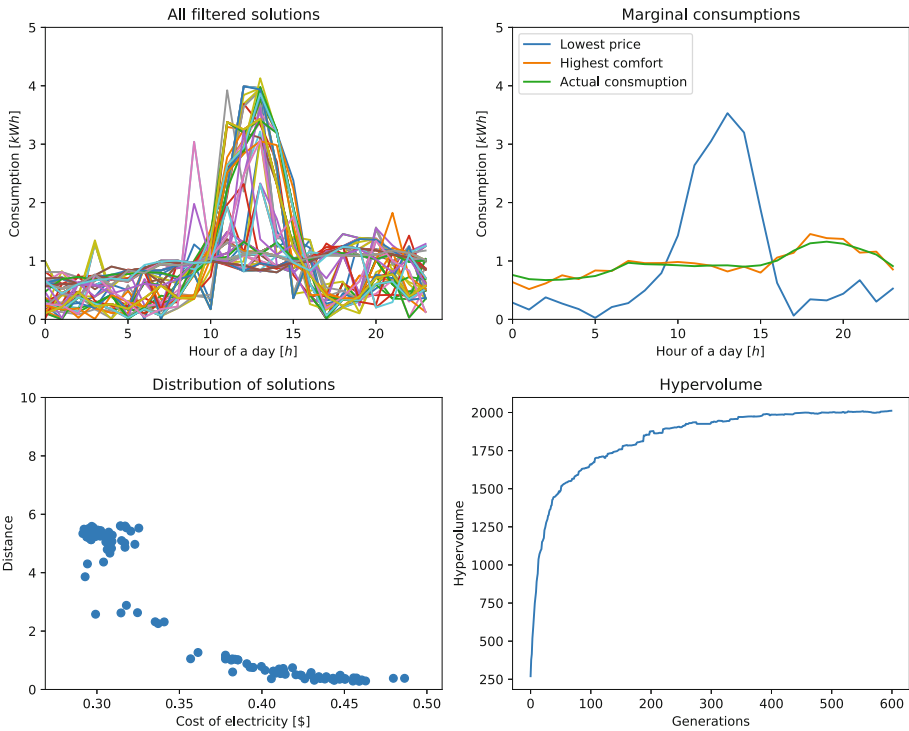


Fig. 2. Visualization of optimization results for average winter day for household with 3 kWh battery.

Table 2. Cost of energy for solution with lowest price on average winter day

Gen	Battery [kWh]	Mean cost [\$]	Std. cost	Saving [\$]	Saving [%]
300	0	0.326	0.027	0.476	59,33
300	3	0.286	0.014	0.516	64,32
300	5	0.282	0.010	0.519	64,83
600	0	0.315	0.011	0.487	60,71
600	3	0.294	0.007	0.508	63,33
600	5	0.279	0.020	0.523	65,20

Table 3. Cost of energy for solution with lowest price on average summer day

Gen	Battery [kWh]	Mean cost [\$]	Std. cost	Saving [\$]	Saving [%]
300	0	0.943	0.078	0.328	25,81
300	3	0.671	0.030	0.600	47,21
300	5	0.565	0.049	0.706	55,55
600	0	0.961	0.024	0.310	24,40
600	3	0.702	0.040	0.569	44,78
600	5	0.575	0.049	0.696	54,77

5 Conclusion and Future Work

We propose a software solution for recommendation of energy consumption for residential user. The main benefits of our solution are the explicit expression of user's comfort through Euclidean distance metric and providing the user with a set of tradeoff solutions from which he can select the most suitable one for the following day. Our method can be used for providing the simple upper bound for maximum savings, which can be achieved by employing PV and battery and is particularly suitable for markets with dynamic day-ahead prices.


In order to deploy our solution in real-life environment, some aspects require more investigation. Every user has its own specific behavior, which is mirrored in his/her energy consumption profile. Accurate predictions for next day would improve accuracy of recommended cost savings and would thus improve the attractiveness of this approach for end users. More advanced charging and discharging strategy could be employed, leading to even more energy cost savings. Cluster analysis of customers load profiles would also be beneficial, mainly for improving the prediction accuracy.

Acknowledgement. This work was partially supported by the Slovak Research and Development Agency under the contract No. APVV-16-0213 and by the Operational Programme Research & Innovation, funded by the ERDF, projects No. ITMS 26240120039 and ITMS 313011B924.

References

1. Dataport from Pecan Street, <https://dataport.cloud/>
2. Agnetis, A., De Pascale, G., Detti, P., Vicino, A.: Load scheduling for household energy consumption optimization. *IEEE Trans. Smart Grid* **4**(4), 2364–2373 (2013)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm. NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
4. Du, P., Lu, N.: Appliance commitment for household load scheduling. *IEEE Trans. Smart Grid* **2**(2), 411–419 (2011)
5. Horst, G.R., Kajfasz, R.: Pacific Northwest GridWise™ Testbed Demonstration Projects Part II. Grid Friendly™ Appliance Project. Appliance 91 (October), 1–123 (2007), http://www.pnl.gov/main/publications/external/technical_reports/PNNL-17079.pdf
6. Martínez-Álvarez, F., Troncoso, A., Asencio-Cortés, G., Riquelme, J.: A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting. *Energies* **8**(12), 13162–13193 (2015). <http://www.mdpi.com/1996-1073/8/11/12361>
7. Mishra, A., Sitaraman, R., Irwin, D., Zhu, T., Shenoy, P., Dalvi, B., Lee, S.: Integrating energy storage in electricity distribution networks. In: Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems, pp. 37–46 (2015), <http://doi.acm.org/10.1145/2768510.2768543>
8. Nguyen, D.T., Le, L.B.: Joint optimization of electric vehicle and home energy scheduling considering user comfort preference. *IEEE Trans. Smart Grid* **5**(1), 188–199 (2014)
9. PacifiCorp: Time of Use Hours & Pricing, <https://www.pacificpower.net/ya/po/otou/ooh.html>
10. Soares, A., Antunes, C.H., Oliveira, C., Gomes, A.: A multi-objective genetic approach to domestic load scheduling in an energy management system. *Energy* **77**, 144–152 (2014)

An Approach for Erosion and Power Loss Prediction of Wind Turbines Using Big Data Analytics

Dina Fawzy, Sherin Moussa^(✉) , and Nagwa Badr

Department of Information Systems, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, Egypt
{dina.fawzy, sherinmoussa, nagwabadr}@cis.asu.edu.eg

Abstract. Due to the huge costs associated with wind energy development, this makes wind farms maintenance and production reliability are of high necessity to ensure sustainability. The continuous evolution of turbines industry has a serious impact on the operation and maintenance costs. Thus, monitoring wind turbines performance and early deterioration prediction are highly required. During the operational life of turbines, some components are persistently exposed to extreme environmental influences that result in their edge erosion. Sensors can be deployed in wind farms to detect such factors, where vast quantities of incomplete, heterogeneous and multi-sourced data are rapidly generated. Hence, wind-related data have been considered as big data that necessitate the intervention of big data analytics for accurate data analysis, which become severely hard to process using traditional approaches. In this paper, we propose the Wind Turbine Erosion Predictor (WTEP) System that uses big data analytics to handle the data volume, variety, and veracity and estimate the turbines erosion rate, in addition to the total power loss. WTEP proposes an optimized flexible multiple regression technique. Experiments show that WTEP achieves high erosion rate prediction accuracy with fast processing time. Thus, it effectively evaluates the accompanied percentage of power loss for wind turbines.

Keywords: Big data analytics · Data mining · Regression analysis · Association rules · Apriori. Principal component analysis · Wind farms reliability · Wind farms maintenance · Erosion. Power prediction

1 Introduction

Wind energy has become a popular source of energy around the world, where its development plants cost huge investments. This requires a keen management of their economic efficiency to ensure higher yields for energy cost reduction [1]. The wind turbine reliability is a critical factor in the success of a wind energy project, which implicates reducing the expensive operation and maintenance (O&M) costs that affect the project's revenue [2]. During the wind turbine's operation, some components, principally the rotor blades, are continuously exposed to certain environmental conditions over time, such as rain, temperature and sand. This results in the deterioration of

the blade's material surface and the increase of its surface roughness if unprotected, leading to its erosion after an average of 2 years of turbines installation and to performance decrease. This requires that in-service maintenance should be performed on the turbine for at least 12 years of operation if it meets its design life, which results in huge maintenance costs [3]. For significant erosion rates (5%–20%), O&M costs are expected to be within \$27–54/MWh. Rain erosion occurs during the processing of turbines in heavy rain. During the high velocity of fallen liquid on a solid target, a high pressure is developed between the solid and liquid, where it varies over many locations [4]. On the other hand, sand erosion exists in the desert environments with the movable dirt and airborne particles affecting turbines' blades, which increase roughness and decrease aerodynamic performance [5, 6]. The high temperature affects wind turbines because the erosion rate increases when the viscosity of liquid reduces [7]. In addition, the increase of wind speed and air density has a positive impact on power production. However, when it exceeds 6 m/s over dry soils, it carries sand and dust towards turbines, leading to erosion. Wind direction is highly effective as well if it is like the sand direction. Whilst a slope that is greater than 20 m affects the angle between the surface and sand/dust, resulting in surface erosion [8]. Sensors can be deployed in the desired location of wind energy plants to monitor such environmental turbines erosion causes, by collecting sensors' data that could be heterogeneous and incomplete massive data [9].

The nature of such wind energy data enforces the desperate need to utilize big data analytics to handle such issues effectively. Big data refers to the collections of so huge and heterogenous datasets that are critically sophisticated to process using customary approaches [10]. This is due to the mainly characterized 4Vs of big data, representing Velocity, Veracity, Variety, and Volume. Big data analytics refers to the usage of advanced analytic techniques against these 4Vs [11]. Wind farm engineers can use big data analytics to manage the risks in order to achieve production goals and recommend activities to address shortfall detected [12]. Thus, the prediction of erosion rate is an efficient way to manage the cost impacts of wind farms through the power usage prediction and the achievement of the supply on demand concept.

In this paper, we introduce the Wind Turbine Erosion Predictor (WTEP) System that uses big data analytics to handle the data volume, variety, and veracity in order to predict turbines' erosion rate. WTEP is built on the top of Trio-V Wind Analyzer system, which is a generic integral system that analyzes the land suitability of a potential location and recommends a distribution layout design, in addition to power prediction using big data analytics prior to wind farms development. WTEP can predict the erosion rate and evaluate its resultant power loss at any spatial region under study based on its environmental factors data rather than other customizable studies. The remaining parts of the paper are organized as follows: Sect. 2 overviews the related works in wind farm reliability, data reduction and power prediction in the wind energy domain. Section 3 presents the proposed system, with a detailed discussion of its architecture. Section 4 explains the experimental approach and the study area. Section 5 discusses the different applied experiments and the associated results. Lastly, Sect. 6 summarizes the conclusion and the future work.

2 Related Work

2.1 Wind Farm Reliability Approaches

Many researches have considered analyzing wind turbines data to maintain wind plants. Most of these studies were done to ensure the reliability of wind farms through extracting the failure history of wind turbines and monitoring their status in order to reduce downtime and increase availability. Authors in [13] monitored the performance of wind farm turbines to detect their downtimes by integrating SCADA system with the turbine's control system and controlling the detected turbines to manage the requirements of power consumption and turbine efficiency. In [14], a platform was developed using the National Reliability Database for turbines' failure detection. Another platform in [15] aimed to discover the hidden patterns in the turbine statuses using the random forest multiclass classification model. SCADA monitoring system was considered in [16] to detect failures by applying an anomaly detection technique. In [17], SCADA data were used to classify the failure events of turbines into severity categories and apply a statistical methodology for each category to decide the wind farm reliability. Since the previous researches have tackled the problem from the engineering perspective, a minimal research effort was dedicated for analyzing the operational and environmental data of wind turbines to raise their performance and reduce the associated maintenance costs. Moreover, most of these studies were poor to process scalable and variable data, since SCADA data are static with a specific format.

2.2 Data Reduction Techniques

Traditional data mining techniques were investigated to fit big data processing. Near Filter Classifier (NFC) upgrades K-Nearest Neighbor (KNN) classification by adding a dimensionality reduction step [18]. It computes the class distribution per every dataset parameter, then sorts the parameters by the calculated value. In [19], Parallel processing was used in the decision tree data mining technique to mine a huge amount of data streams. In addition, "Scalable Advanced Massive Online Analysis (SAMOA)" technique used parallel processing with distributed decision trees for data mining classification over big data [20]. Another upgrade was applied to reduce big data volume using parallel processing by applying K-means on several nodes and combining the results [21]. Although these researches were dedicated to reducing data volume, but they were poor to reach high accuracy that doesn't exceed 60%, with high processing time that reaches 100s with five neighbors [18]. This is in addition to the extra communication time between nodes in the parallel processing approaches, which leads to excessive processing time [19].

2.3 Prediction Techniques for Wind Energy

Several prediction techniques were dedicated for wind energy domain. In [22], the weather prediction used genetic programming. The wind speed and generated power were predicted in [23] using a fuzzy expert system. Artificial neural networks were used in [24] to predict electrical power generated from wind farms. However, such

prediction techniques have just reached 85% accuracy [22] and 20% error rate [24]. The fuzzy system consumes much processing to learn the model that cannot fit big data processing [23].

The contributions in this proposed research can be summarized as follows. (1) We propose the Wind Turbine Erosion Predictor (WTEP) as an integral system for predicting the erosion rate of wind turbines from the data analytical perspective to decrease turbines failure rate. (2) It uses big data analytics to handle wind turbines data volume, variety and veracity, where Double-Reduction Optimum Apriori (DROA) approach is proposed. (3) It presents a new Optimized Flexible Multiple Regression (OFMR) approach to fit big data processing to predict wind turbines erosion rate, taking into consideration the different affecting environmental factors that can be adapted and generalized to wherever the study area is located. Hence, it can fit to evaluate any wind farm irrespective of its location rather than any customized systems to study certain territories, which is one of the main strengths of this proposed system. (4) It predicts the power loss accompanied by the predicted erosion rate.

3 The Proposed Solution

In this section, we present the proposed Wind Turbine Erosion Predictor (WTEP) system. As shown in Fig. 1 representing the system architecture, WTEP is developed in accordance with Trio-V Wind Analyzer system to achieve WTEP functionalities. The study presents the complete work of the proposed system, providing its architecture, the detailed explanation and implementation of all its components, and the associated experimentations. WTEP deals with the data layer managing the factors data of wind farms, in addition to the presentation layer that is connected to the sensors and Google map to manage the user selections and to display the analytical results plotted on the map or generated in reports. WTEP works as shown in Fig. 2. The system user determines the wind farm location and the reduction method to apply on the sensed factors data. WTEP collects the factors data from the sensors in the defined location and then manages their biases and noises using the Variety-Veracity WA Handler [25].

Next, the selected reduction method is applied on the data using the Volume WA Handler. The resultant processed data is then used to analyze the erosion rate and evaluate the associated power loss using Trio-V Power Loss Analyzer in a detailed report, showing each cell with its corresponding erosion and power loss rates. Data velocity, in terms of data processing in the form of a stream, is not handled in this system since it doesn't require real-time processing. Sensors data are accumulated in the data layer, taking into consideration the time representation of data as another data dimension for offline processing and analysis, where the collected data are strongly related to the recording time. Thus, a data stream handler is not required. The main components of WTEP are further explained hereinafter.

3.1 Presentation Layer

This layer provides the User Interface (UI) of the system, which enables the user to determine the wind farm's location and collect the associated environmental factors'

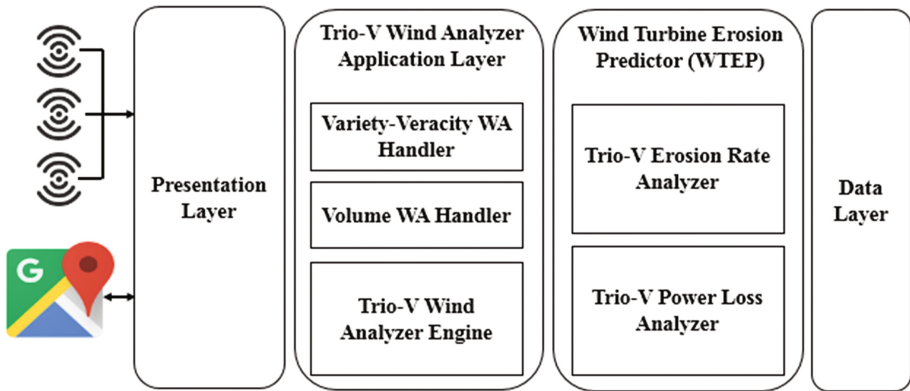


Fig. 1. The proposed system architecture

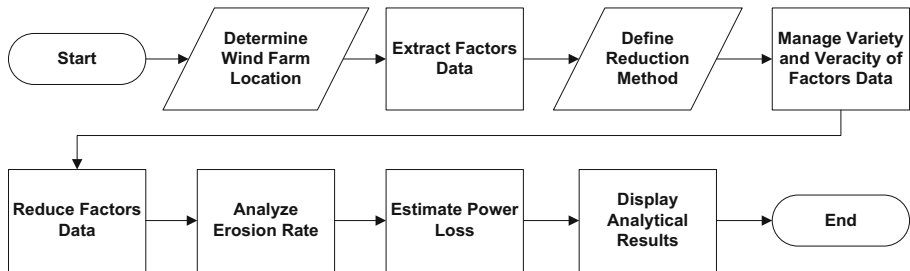


Fig. 2. Wind Turbine Erosion Predictor (WTEP) system flowchart

data from the deployed sensors. It then divides the land into cells of equal size as per a user-defined cell size parameter. In addition, it allows the user to choose a reduction method to manage the huge size of data. Finally, WTEP prediction results are displayed in a detailed report with the suitable graphs per cell, visualizing the expected erosion rate and the corresponding predicted power loss rate.

3.2 Trio-V Wind Analyzer Application Layer

This layer handles the huge Volume, Veracity and Variety (Trio-V) features of the collected environmental factors' data, which are generated from the sensors deployed at the land under study. Then, it evaluates the suitability of this land to establish a wind farm and suggests a distribution layout for the turbines. The main components are explained as follows.

Variety-Veracity WA Handler. This module manages the biases and noise detected in the sensors data while considering its big data nature. It validates the data quality and data inconsistencies before storage into the data layer through several data cleansing processes, including noisy data deletion and filling in missing data with the mean value.

Encoding-decoding processes are considered as well to transform specific factors' data into a certain format to be processed [25].

Volume WA Handler. The deployed sensors generate enormous amounts of data. Thus, this module applies the reduction method that has been selected from the presentation layer. The data layer structure includes different environmental factors to identify each cell, where each factor has excessive amount of data per one cell. WTEP provides several alternative reduction methods merged from different reduction techniques to apply on the cells' factors data. Some of these techniques are responsible for reducing the number of cell factors used for analytics, like Principle Component Analysis (PCA) and Association Rules (ARs) (i.e. column reduction), while others reduce the amount of cell data, like aggregations. PCA is a data reduction technique that uses a mathematical approach to reduce many correlated parameters into a small set of uncorrelated parameters called principal components (PCs). WTEP uses the correlation approach to match the resultant PCs to their corresponding factors in the original dataset by calculating the correlation coefficient between every PC(x) and each factor (y) in the original data using Eqs. (1), (2), (3), and (4) [26]. The factor having the highest correlation coefficient represents the PC.

$$S_{xx} = \sum x^2 - \frac{(\sum x)^2}{n} \quad (1)$$

$$S_{yy} = \sum y^2 - \frac{(\sum y)^2}{n} \quad (2)$$

$$S_{xy} = \sum xy - \frac{(\sum x)(\sum y)}{n} \quad (3)$$

$$\text{CorrelationCoff} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}} \quad (4)$$

Where n represents the number of records for the cell's factors, x represents the resultant PC; and y is the cell's factors data needed to be reduced [26]. The higher the result means that this PC is most correlated to this factor. As for the ARs, we enhanced the original version of the Apriori technique to fit the big data processing by introducing our optimized Apriori algorithm named "Double-Reduction Optimum Apriori" (DROA) to extract the most informative relationships between the factors using the criteria of support and confidence according to Eqs. (5) and (6) [27]. The proposed DROA ARs optimizes the Apriori algorithm to support big data volume by applying two phases before running the basic Apriori; (1) using database scanning time reduction that saves a screenshot of the desired transactions between erosion factors related to a certain area in a supportive map data structure, which decreases the traditional Apriori processing time. (2) Using transactions reduction that reduces transactions by discarding the unsuitable ones that violate erosion values constraints [25]. This allows DROA ARs to work efficiently on a huge number of transactions.

$$\text{Support}_i = \frac{\text{FPi}}{\text{TFP}} \quad (5)$$

$$\text{Confidence } (A \rightarrow B) = \frac{\text{support}(A \cup B)}{\text{support}(A)} \times 100 \quad (6)$$

Where support_i is the support of the i th factor, FPi is the number of times the i th factor is found, and TFP is the total number of factors found. Confidence $(A \rightarrow B)$ represents the confidence of occurrence; if A occurs, then B will occur too. For more processing efficiency, WTEP allows merging several approaches of reduction methods to additionally reduce data. Thus, the reduction alternatives are: Aggregation functions only, Aggregation followed by PCA, Aggregation followed by DROA ARs, PCA followed by Aggregation, or DROA ARs followed by Aggregation. For instance, aggregation only would be sufficient for small datasets, whereas DROA ARs and PCA are more appropriate for huge datasets.

Trio-V Wind Analyzer Engine. This module is the core of Trio-V Wind Analyzer. It uses big data analytic techniques to perform land suitability analysis for wind farms prior to development. Trio-V determines land suitability through evaluating its environmental factors. Upon the positive evaluation, Trio-V Wind Analyzer recommends the optimum wind farm design that avoids the wake effect problem of turbines and maximizes the generated power by suggesting the suitable turbines' specifications and their distribution layout depending on the analyzed factors of the potential location. Accordingly, it then predicts the expected generated power from this recommended design [25].

3.3 Wind Turbine Erosion Predictor (WTEP)

This module explains the main WTEP functionalities in the following sub-modules:

Trio-V Erosion Rate Analyzer. This module is responsible for determining the erosion rate per one turbine for each land cell by evaluating specific environmental factors; rain, sand, wind speed, slope, wind direction, air density, and temperature [4–7, 33]. These environmental factors have variable values over time per year, where severe edge erosion can be caused if certain thresholds were exceeded as clarified in Table 1. WTEP considers the influence of such variance of values on erosion. For example, the dust storms could be very erosive compared to daily wind. However, the continuous direct exposure of everyday wind can even affect turbines erosion. Thus, all variances of the different factors are considered in the data analytics process to determine the erosion rate with an acceptable accuracy. WTEP aggregates all the previous erosion factors data from the collected sensors data for each cell at the potential land under study. Since WTEP processes one turbine at a time, the number of installed turbines per land cell is not considered as a factor in the erosion rate analysis. The Trio-V Erosion Rate Analyzer module estimates the erosion rate using our proposed Optimized Flexible Multiple Regression (OFMR) technique, which is an enhanced flexible form of the original multiple regression technique to support big data volume by considering a dynamic number of predictors, whereas the original multiple regression analysis is a statistical

Table 1. Erosion factors constraints

Factor	Erosion values constraint
Rain	>100 mm
Sand	1–200 μm
Steel temperature	>160°
Wind Speed	>6 m/s
Slope	>20
Wind direction	–
Air density	–

technique for analyzing relationships between factors using multiple predictors in fixed prediction equation parameters [28]. The following model in Eq. (7) shows the multiple linear regression model with K predictor variables.

$$Y = B_0 + B_1X_1 + B_2X_2 + \dots B_KX_K \quad (7)$$

Where parameter B_0 is the intercept of this plane, while “Y” is the unknown value to be predicted, and parameters $B_1, B_2 \dots B_K$ are referred to as regression coefficients [29]. OFMR supports big data volume by considering a dynamic number of predictors. It can build the model depending on the considered erosion factors based on the land under study, which are additionally reduced in the Volume WA handler, rather than building one fixed model based on all factors. Therefore, OFMR ensures more accurate results than traditional multiple regression technique, where the erosion rate “Y” is correlated only to the existing factors from Volume WA handler. OFMR consumes less processing time due to the flexibility in building the model with any number of predictors. It handles the biases and noise detected in the sensors data by ensuring data quality before building the model using the Variety-Veracity Handler. Thus, OFMR manipulates the overfitting problem in the traditional multiple regression model. Moreover, the factors’ values that are less than the erosion constraint thresholds will be ignored by the OFMR regression model. These features made WTEP adapted and generalized to evaluate any wind farm irrespective of its location, taking into consideration the different affecting environmental factors that would be associated by this location.

Trio-V Power Loss Analyzer. Leading edge erosion poses a major threat to the performance of wind turbines. The modeling of power output per one turbine is a trivial approach that assumes static wind parameters. However, a turbine’s status is inconstant, due to the erosion factors and wind parameters like wind speed, and air density that continuously change and affect the turbine’s status [30]. This module allows WTEP to evaluate the power loss rate according to the predicted erosion rates resulted from Trio-V Erosion Rate Analyzer. Power loss prediction is performed by applying a single linear regression technique using the predicted erosion rate value. Single regression analysis explores relationships statistically, containing one predictor as shown in Eq. (8) [31].

$$Y = B_0 + B_1X \quad (8)$$

Where “Y” is the power loss rate; “X” is the resultant erosion rate; B_0 is the intercept of this plane; and B_1 is the regression coefficient.

4 Case Study Area

Egypt climate is affected by several factors, including its position that lies between Africa and Asia [32]. These factors give Egypt a hot and sunny weather, with a very low humidity. The erosion factors value at the main areas in Egypt are presented in Table 2 [33, 34]. As for the wind direction and air density, their values are continuously changing during the year.

Table 2. Erosion factor values at egyptian areas

Area	Sand particles	Rain	Temperature	Wind speed	Slope
Western desert	<170 microns	20 mm	36 °C	5.3 m/s	32 m
North coast	<120 microns	196 mm	20 °C	6.2 m/s	18 m
Red sea	<80 microns	2.3 mm	30 °C	8 m/s	27 m

5 Experimental Results and Evaluation

WTEP has been developed using JAVA, MS SQL Server and APIs to some scientific libraries and external components. Experiments were held to evaluate WTEP from two points of view: the big data processing efficiency and wind analytics accuracy. Hence, the experimentation is categorized into: the erosion and power loss rates prediction accuracy, and the associated processing time versus the different reduction methods to emphasize that the proposed data reduction and prediction techniques are suitable for big data analysis, supported by a comprehensive comparison with the relevant existing state-of-arts. The evaluation was demonstrated on a machine having core i7, 2.70 GHz, 1T hard disk space, and 8 GB RAM. OFMR prediction accuracy is evaluated using the Root Mean Square Error (RMSE) as per Eq. (9) [29]:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2} \quad (9)$$

Where “N” is the number of data points, “ x_i ” the original observed value and “ y_i ” is the predicted value corresponding to the current original data point “ x_i ”. RMSE values vary from 0 to 100 in order to be mapped to percentages, in which the smaller values indicate higher accuracy. RMSE values that are within (0–10) represent an accuracy from 90% and above. A sample of the experimental results of Red Sea area are discussed hereafter, since it is one of the potential areas in Egypt for wind plants. Three dataset sizes are used; small dataset D1 with 100,000 records, medium dataset D2 with

2,2500,000 records, and large dataset D3 with 5,750,000 records. The average temperature is 30 °C with 4 °C variation during winter. The rainfall is low, averaging 2.3 mm per year with average speed 8 m/s, air density equals to 1.2 kg/m³, and slope of 27 m. Red Sea area has occasionally dust storms as well [32]. The values of erosion factors differ depending on the measurement height, representing the height at which the values are detected and recorded. Thus, each dataset is tested for three turbine scale heights; 80, 50 and 30, representing the standard turbines' hub heights in the market. DROA ARs is investigated at confidence and support thresholds: 0.3, 0.5, 0.7 and 0.9, whereas PCA is studied at K-values: 5 and 3. These values have been configured as per many trials of experimental preparation, where their fair representation has been proven to the remaining values.

5.1 Erosion Rate Accuracy vs. Reduction Methods

Previous researches have considered wind farms reliability from the technical fault prediction perspective. In [13], 90% system availability has been achieved using SCADA data monitoring. Random forests data mining was used in [14] to predict turbines' failures with 8.3% error rate. Authors in [15] considered anomaly detection algorithms to detect turbines failures with 90% accuracy. In [24], 88.84% of failures were detected in a detection system of turbine failures using SCADA data. Despite of these previous researches, but they predicted the failures of turbines. To the best of our knowledge, WTEP is the first data analytical system that predicts turbines erosion and power loss rates using big data analytics. Thus, experiments were carried out to evaluate the big data processing efficiency by studying the RMSE results of the erosion rates prediction using OFMR over the different reduction methods. As shown in Fig. 3 for the RMSE results of the three datasets over WTEP reduction methods, the erosion prediction RMSE decreases as the dataset size increases.

Table 3 summarizes RMSE results over D3, representing the largest dataset. The aggregation only has the most accurate results due to the complete number of factors used, then DROA ARs with a reasonable accuracy results, and PCA has the least

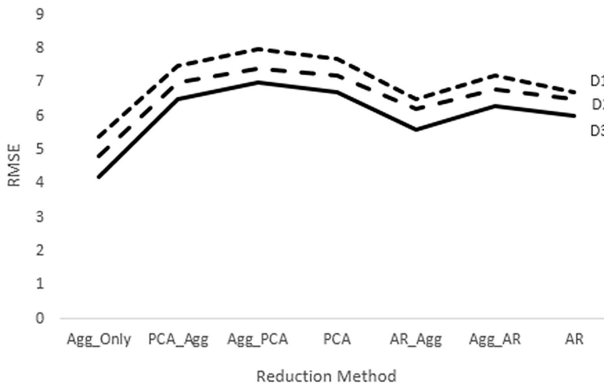


Fig. 3. Erosion prediction RMSE vs. reduction methods

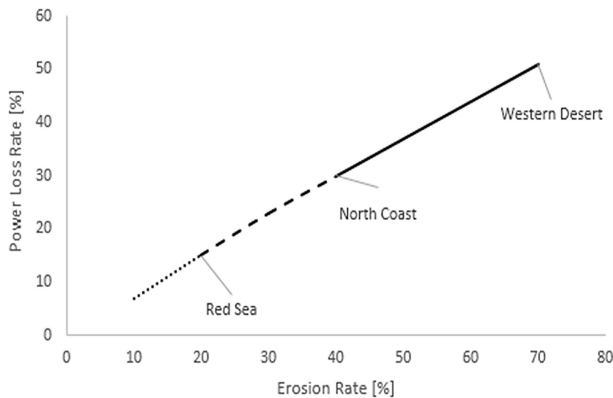
Table 3. Erosion rate prediction at D3

	Agg. Only	ARs+ Agg.	Agg. + ARs	PCA + Agg.	Agg. + PCA
RMSE	4.2	< 5.5	< 6.2	< 6.5	< 7.3

accurate results. Applying DROA ARs or PCA followed by aggregation, the erosion rate prediction error is 10% less than using aggregation first then DROA ARs or PCA, since the erosion rate is calculated from the correlated results generated from DROA ARs or PCA rather than working on all the factors.

5.2 Power Loss Rate vs. Erosion Rate

For the wind analytics evaluation, WTEP have traced the resultant power loss rate over several erosion rate values per three different areas (Western Desert, Red Sea and North Coast) for the largest dataset D3. The higher erosion rate, the more power loss rate as shown in Fig. 4, where each line style represents the erosion rate values interval at a certain area. Erosion rates exceeding 45% represent a major threat to the power production process, as it leads to 30% and more power loss. Figure 3 proves that the erosion rate is high at the Western Desert that reaches 48% and North Coast with 33% due to the increase of sand and fallen rain respectively, whereas a normal erosion rate at Red Sea reaches 17%.

**Fig. 4.** Power loss rate vs. erosion rate

5.3 Processing Time vs. Reduction Methods

Evaluating the big data processing efficiency, WTEP processing time is tested at the different reduction methods for the three datasets as presented in Fig. 5. The processing time increases by enlarging the dataset size. Table 4 shows the results over D3, where

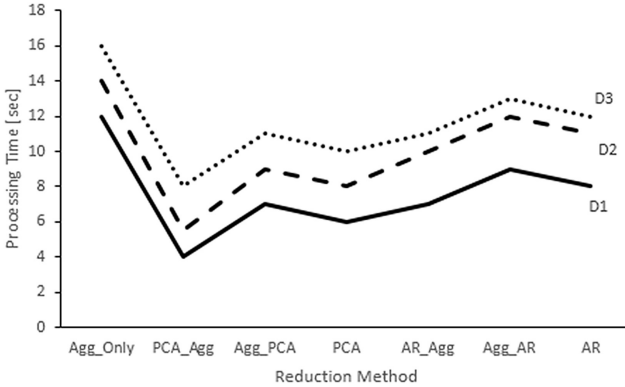


Fig. 5. Processing time vs. reduction methods

Table 4. Processing time over reduction methods

	Agg. Only	ARs + Agg	Agg. + ARs	PCA + Agg.	Agg. + PCA
D3	16	< 11.2	< 12.7	> 8.2	< 11

the largest data can judge the processing time efficiency. The aggregation only consumes the highest processing time due to working on all factors to predict the erosion rate, in contrast to DROA ARs since it works on a less number of factors. The lowest processing time is consumed by PCA. Moreover, using the aggregation first then DROA ARs decreases the number of factors, which reduces the processing time by 20% rather than that of aggregation only that uses all factors in processing. Applying PCA then aggregation, the processing time is 25% less than that of aggregation followed by PCA and 40% less than that of aggregation only. Decreasing K-value by 2, the processing time is reduced by average 2 s. On the other hand, increasing the confidence and support values by 0.2 reduces the processing time by average 3 s, because of reducing the number of factors used for processing.

6 Conclusion

Many researches have considered wind farms reliability evaluation to manage their operation and maintenance costs from the engineering viewpoint. In this paper, we introduce Wind Turbine Erosion Predictor (WTEP) system for predicting the erosion rate of wind turbines from the data analytics perspective to minimize turbines failure rate. WTEP proposes a novel Optimized Flexible Multiple Regression (OFMR) approach for erosion rate prediction that fits big data processing. In addition, it applies a new approach for big data volume handler using Double-Reduction Optimum Apriori (DROA). The Variety-Veracity Handler ensures data quality used for turbines erosion

analysis and power loss prediction. Experiments were performed to evaluate big data processing efficiency and wind analytics at several areas in Egypt, where OFMR reaches >90% in efficient processing time. DROA ARs generates reasonable accurate results in less processing time. The experiments held on the Egyptian locations datasets confirm that the lowest erosion rate is at Red Sea. Our future work is to consider the economic models of wind farm profitability using big data analytics.

References

1. Ackermann, T.: *Wind Power in Power Systems*. Wiley, New York (2005)
2. Walford, C.: *Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs*, United States, Department of Energy (2006)
3. Rempel, L.: Rotor blade leading edge erosion-real life experiences. *Wind Syst. Mag.* **11**, 22–24 (2012)
4. Tobin, E., Young, T., Raps, D., Rohr, O.: Comparison of liquid impingement results from whirling arm and water-jet erosion test facilities. *Wear* **271**, 2625–2631 (2011)
5. Poudel, L., Thapa, B., Shrestha, B., Shrestha, N.: Impact of sand on hydraulic turbine material: a case study of Roshi Khola, Nepal. *Hydro Nepal: J. Water Ener. Environ.* **10**, 60–65 (2012)
6. Fiore, G., Selig, M.: Simulation of damage for wind turbine blades due to airborne particles. *Wind Eng.* **39**, 399–418 (2015)
7. Tabakoff, W.: High-temperature erosion resistance coatings for use in turbomachinery. *Wear* **186–187**, 224–229 (1995)
8. Bakirtzis, A.: A probabilistic method for the evaluation of the reliability of stand alone wind energy systems. *IEEE Trans. Ener. Convers.* **7**(1), 99–107 (1992)
9. Fiore, G., Selig, M.: Simulation of damage progression on wind turbine blades subject to particle erosion. In: *54th AIAA Aerospace Sciences Meeting* (2016)
10. Fawzy, D., Moussa, S., Badr, N.: The evolution of data mining techniques to big data analytics: an extensive study with application to renewable energy data analytics. *Asian J. Appl. Sci.* **4**(3), 756–766 (2016). ISSN: 2321–0893
11. Jadhav, D.: Big data: the new challenges in data mining. *Int. J. Innov. Res. Comput. Sci. Technol.* **1**(2), 39–42 (2013)
12. Richardson, D.: Electric vehicles and the electric grid: a review of modeling approaches, Impacts, and renewable energy integration. *Renew. Sustain. Ener. Rev.* **19**, 247–254 (2013)
13. Kim, H., Kim, C., Yue, L.: A study of the growth of single-phase Mg₀.5Zn₀.5O Films for UV LED. *Int. J. Chem. Nucl. Mater. Metall. Eng.* **8**(7), 674–678 (2014)
14. Hill, R., Jennifer, R., Stinebaugh, A., Briand, D., Benjamin, A.S., Lindsay, J.: *Wind turbine reliability: a database and analysis approach*. Sandia National Laboratories, Albuquerque, New Mexico, 87185 (2008)
15. Verma, A.: *Performance monitoring of wind turbines: a data-mining approach* (2012)
16. Kim, K., Parthasarathy, G., Uluyol, O., Foslien, W., Sheng, S., Fleming, P.: Use of SCADA data for failure detection in wind turbines. In: *ASME 2011, 5th International Conference on Energy Sustainability*, pp. 2071–2079 (2011)
17. Kaidis, C.: *Wind Turbine Reliability Prediction: A Scada Data Processing & Reliability Estimation Tool* (2014)
18. Niu, K., Zhao, F., Zhang, S.: A fast classification algorithm for big data based on KNN. *J. Appl. Sci.* **13**, 2208–2212 (2013)

19. Murdopo, A.: Distributed decision tree learning for mining big data streams. Master of Science Thesis, European Master in Distributed Computing. Barcelona, Spain (2013)
20. De Francisci, M.G.: SAMOA: a platform for mining big data streams. In: Proceedings of the 22nd International Conference on World Wide Web, pp. 777–778 (2013)
21. Cai, X., Nie, F., Huang, H.: Multi-view k-means clustering on big data. In: Twenty-Third International Joint Conference on Artificial Intelligence (2013)
22. Vladislavleva, E., Friedrich, T., Neumann, F., Wagner, M.: Predicting the energy output of wind farms based on weather data: important variables and their correlation. *Renew. Ener.* **50**, 236–243 (2013)
23. Watanabe, T., Fujioka, R.: Fuzzy association rules mining algorithm based on equivalence redundancy of items. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1960–1965. IEEE (2012)
24. Alexiadis, M., Dokopoulos, P., Sahsamanoglou, H.: Wind speed and power forecasting based on spatial correlation models. *IEEE Trans. Ener. Convers.* **14**, 836–842 (1999)
25. Fawzy, D., Moussa, S., Badr, N.: Trio-V wind analyzer: a generic integral system for wind farm suitability design and power prediction using big data analytics. *J. Ener. Res. Technol.* JERT 2017, ASME. <https://doi.org/10.1115/1.4038119>
26. Holland, S.: Principal components analysis (PCA). University of Georgia (2008)
27. Giannotti, F., Lakshmanan, L., Monreale, A., Pedreschi, D., Wang, H.: Privacy-preserving mining of association rules from outsourced transaction databases. *IEEE Syst. J.* **7**(3), 385–395 (2013)
28. Lam, J., Wan, K., Liu, D., Tsang, C.: Multiple regression models for energy use in air-conditioned office buildings in different climates. *Ener. Convers. Manag.* **51**, 2692–2697 (2010)
29. Aranda, A., Ferreira, G., Mainar-Toledo, M., Scarpellini, S., Sastresa, E.: Multiple regression models to predict the annual energy consumption in the Spanish banking sector. *Ener. Build.* **49**, 380–387 (2012)
30. Eminoglu, U., Ayasun, S.: Modeling and design optimization of variable-speed wind turbine systems. *Energies* **7**, 402–419 (2014)
31. Freire, R., Oliveira, G., Mendes, N.: Development of regression equations for predicting energy & hygrothermal performance of buildings. *Ener. Build.* **40**, 810–820 (2008)
32. Eladawy, A., Nadaoka, K., Negm, A., Saavedra, O., Hanafy, M.: Assessment of long term thermal stress on egyptian coral reefs based on remotely sensed sea surface temperature data. *Int. J. Environ. Sci. Dev.* **6**(12), 938–946 (2015)
33. Mortensen, N.G.: Wind atlas for Egypt: measurements, micro-and mesoscale modelling. In: European Wind Energy Conference and Exhibition (2006)
34. El-Barbary, Z., Sallam, D.: Optimizing use of rainfall water in east desert of Egypt. In: Proceedings of 8th International Water Technology Conference, IWTC8, Egypt (2004)

Minimizing Grid Interaction of Solar Generation and DHW Loads in nZEBs Using Model-Free Reinforcement Learning

Adhra Ali^{1(✉)} and Hussain Kazmi²

¹ Technical University of Delft, Jaffalaan 5, 2628BX Delft, The Netherlands
adhra8@gmail.com

² Enervalis, Greenville Campus, Centrum Zuid 1111, Houthalen-Helchteren, Belgium

Abstract. This study applies model-free reinforcement learning (RL) on a case study based in Utrecht province in the Netherlands to optimize for on-site renewable energy. This aims at reducing the interaction of net zero-energy buildings with the grid as a result of an increase of heat pump installations and renewable energy systems (RES) integration. It is believed that this will become increasingly more important since the regulations regarding 2020 and beyond ascribe significant increase in energy efficiency of the built environment. On-site RES self-consumption is therefore a central lead in this research. The project data comprise air source heat pump and solar energy data of 6 different households for the months June to November 2016. The RL algorithm is applied to the different data sets to derive an optimized individual and generalized control strategy. Simulations were carried on, to acquire the resulting energy consumption, self-consumption, and self-sufficiency. The results show an increase of individual self-consumption between 17% and 348% and self-sufficiency between 18% and 72%. This results in an additional monetary benefit for the occupants based on the transition proposals of 2020 for the renewable energy generation net-metering abolishment in the Netherlands. Furthermore, reducing the grid interaction implies benefits for the grid operators in terms of investments required for grid reinforcement.

Keywords: nZEB · Demand response · Smart grid · Reinforcement learning · Solar power · Heat pumps · Self-consumption · Self-sufficiency

1 Introduction

Improvements of the energy efficiency of the built environment in the Netherlands is planned partly to be achieved by increasing the number of nZEB concepts by building new nZEBs as well as refurbishing existing buildings [1]. An nZEB is a nearly zero energy building which consumes very little energy over the course of a year (and preferably meets most of this using renewable energy). Residential nZEBs are dominantly provided by heat pumps (HP) for HVAC and DHW operation to disconnect from the gas network while solar panels are used to generate the required energy to meet the local energy demand. This setup increases the interaction of the residential built environment

with the electricity grid as a result of solar power injection and also increased power consumption as a result of the HP load.

This could be addressed by integrating more storage facilities and reinforcing the grid components such as the transformers and cables which, given the required RES goals of 2020 and beyond, implies significant investments. The weak link in this area of renewable energy generation is the lack of large scale storage capacity due to its relatively high cost. Increasing the size of the grid components is a similarly costly procedure. However, it is believed that the necessity for a costly reform of the energy infrastructure could be reduced and substantially spread over a larger period of time if the variety of decentralized actors could be centrally monitored and managed [2]. The need for centrally monitoring and managing the variety of power sources and sinks in the grid has introduced the concept of Smart Grids [2]. Smart Grids are required to facilitate real-time monitoring and steering tools so as to allow the distribution and transportation capacity of the grid to progress in a more flexible fashion. In the traditional top-down approach of the energy system, the Distribution Systems Operators (DSO) are primarily responsible for the operation and maintenance of the distribution system. However, the need for Smart Grids imposes the need for a change in this role to a more pro-active grid management. This introduces new service opportunities for grid operators. Connecting a large number of nZEB buildings with fluctuating individual energy generation is one such example of locally generated energy that adds to the volatility of the grid power for which grid operators could monitor and manage [3].

Research in this area ranges from fully automated building energy discarding occupant behavior, to smart metering objectives aiming at increasing occupants' energy performance awareness, and control strategies based on electricity prices signals etc. [4–7]. A large part of this research applies rule-based control (RBC) and model-predictive control (MPC) methodologies [7, 8, 10–12]. Generally, MPC performs significantly better than RBC as a result of accurate modelling of the system physics [12]. Both methodologies have benefits and drawbacks related to their application with the model complexity being the main central technical difference [12]. Although both methods have proved to reduce the grid interaction on building level to a certain extent, both methods lack generalizability potential [9]. Hence, when applied on a large scale the system parameters have to be adjusted from system to system. To solve this problem, a learning based methodology is proposed in this paper.

This study, therefore, applies model-free reinforcement learning (RL) to optimize for the self-consumption and self-generation of the DHW load of the HP installation in six different households of a refurbishment project in the Netherlands conducted by a housing corporation. “De Stroomversnelling” project is implemented to refurbish low performing dwellings into nZEB concepts [13]. This is done by applying a high performing façade to the existing one and supplying the houses with solar panels. The houses are fully disconnected from the gas network and operate therefore only on electricity.

2 Methods

2.1 System Setup and Data

An air source heat pump (ASHP) is installed to supply the hot water and special heating. The ASHP has a COP ranging between 3 and 5 for a temperature range of 45 °C and 51.5 °C, depending on the outside temperature. Secondly, a hot water storage vessel of 200 L is installed in the dwellings which is only used for DHW purposes. The third system component is an array of solar panels. The data is measured in 5 min uniform intervals in the form of three different sensors each measuring a different parameter:

1. A hot water flow rate meter
2. A smart meter registering the power consumption of the heat pump at different modes and the power delivered by the PV panels
3. A temperature sensor in the storage vessel.

The data is measured and sent to a central repository in which the baseline control and/or the optimization algorithm is embedded. The individual control action is calculated based on the embedded control strategy and subsequently sent as an action signal to the corresponding ASHP per house.

2.2 Baseline Control and Comfort Standards

The default control strategy is a simple rule-based control algorithm given by Eq. 1 [9]:

$$\mathbf{a}_t = \begin{cases} 1, & \text{if } T_m < T_{th} \\ 0, & \text{if } T_m \geq T_{th} \end{cases} \quad (1)$$

In which \mathbf{a}_t is the action taken by the heat pump: 1 when it is heating and 0 when it is off. Considering the temperature in the vessel, T_m is the temperature measured by the midway sensor in the vessel and T_{th} is the threshold temperature set point [9]. The comfort standards usually dictate that if T_{th} is lower than 45 °C, the ASHP initiates a reheat cycle. Additionally, the upper limit for T_m was set threshold of 51.5 °C to which T_m increases during a reheat cycle. Additional comfort standards determine the water content of the vessel containing the threshold temperatures. A minimum of 50 L with temperature T_m of at least 45 °C, was defined as a minimum requirement to maintain occupant comfort. The PV output is not considered in the default control strategy as the main goal of that control strategy is to comply to the comfort standards. The solar energy delivered is only considered for compensating the energy required from the grid in order to comply to the nZEB concept on annual basis [13].

2.3 Optimization Objectives

The main objective for this study is to shift the DHW energy consumption towards high solar energy production hours in order to reduce the energy required during low

production hours. It is also important to maintain a balanced ASHP consumption to not affect the nZEB concept by either violating the occupant comfort or a large rise in the energy consumption. Two main optimization indicators for this objective, are defined as the relative change in self-consumption and self-sufficiency compared to the baseline conditions. Self-consumption is defined as the energy of the load that was covered by the solar energy expressed in either in kWh in absolute terms or as a percentage of the total generated energy. Self-sufficiency on the other hand, is defined as this same energy covered by the solar energy expressed either in kWh in absolute terms or a percentage of the total load. The following Fig. 1 illustrates the two concepts:

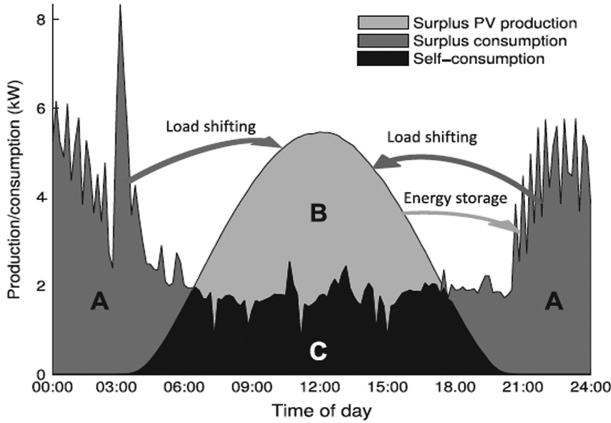


Fig. 1. Schematic overview of the daily net load ($A + C$), net generation ($B + C$), and absolute self-consumption (C) [14]

2.4 Application to Case Study

The control algorithm that is aimed at in this study, is an algorithm that optimized for onsite solar energy consumption. The aim is therefore to utilize the hot water storage vessel similarly to a storage capacity for the PV energy. Hence, it is important to use as much PV energy as possible during the high solar energy production hours of the day. To achieve this, a SARSA(λ) algorithm is applied with a reward function that takes several parameters regulating the vessel state into consideration according to:

$$Q_{(s_t, a_t)} \leftarrow (1 - \alpha)Q_{(s_t, a_t)} + \alpha[r_{t+1} + \gamma Q_{(s_{t+1}, a_{t+1})}] \quad (2)$$

Here, $Q_{(s_t, a_t)}$ represents the state-action value showing how good it is to take action a when being in state s . α represents the step-size parameter which functions as the exponentially moving average parameter. It is especially useful for non-stationary environments for weighting recent rewards more heavily than long-past ones. If α is a number smaller than one for non-stationary environments which indicates that recent updates weight more than previous ones. This transition happens after every nonterminal state.

The $Q_{(s_{t+1}, a_{t+1})}$ of every terminal state are defined as zero. Hence, every terminal state has

an update value of 0. Once the correct modes and their corresponding variables have been filtered, the episodes are simulated accordingly. Each episode consists of the states during which the mode was 0 and ends with the state at which action 1 was taken. From this data, the features needed for the SARSA(λ) code are compiled.

The reward function applied in this algorithm is presented in a flowchart fashion in Fig. 2. This reward will be appended to each action 0 to be taken. Action 1 leads to a terminal state which will lead by definition to a standard update of 0 as described previously. Consequently, if the Q-value of taking action 0 is higher than 0, action 0 is more favorable than taking action 1 and vice versa. A drawback to the binary action-space, is that action 1 could not be punished or rewarded. However, this is solved by amplifying the negative or positive reward for not reheating (action 0) instead.

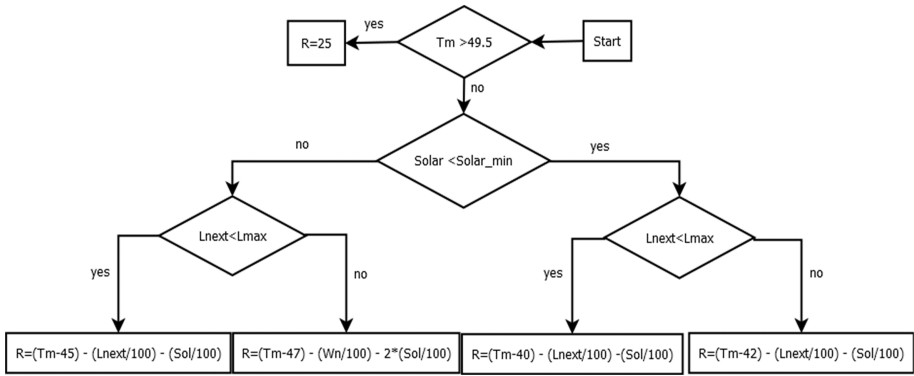


Fig. 2. Flowchart of reward function

To prevent the vessel from constantly reheating when there is high generation of solar energy, a temperature of 2 °C below T_{start} is defined as the threshold temperature for the ASHP to reheat the vessel. This is equal to 49.5 °C when T_{start} is set to 51.5 °C such as in the case in the default strategy. Therefore, two main scenarios are created this way, one describing a midpoint temperature of higher than 49.5 °C and one lower. Temperatures higher than 49.5 °C receive a constant high reward of 25 points to ensure an idle state. On the other hand, to fully exploit the solar energy when available, if the temperature falls below this threshold and the solar energy available is higher than that is needed for rising the temperature of the vessel to 51.5 °C again, negative rewards are favoured to enhance reheating.

When the temperature falls below 49.5 °C and the solar energy available is higher than what is needed to increase T_m by 2 °C, the algorithm will append an increasingly negative reward to temperatures below 49.5 °C which depend on the amount of water consumed. For this purpose, the water consumption threshold is set according to the comfort standards already defined. If the water consumed at time t is higher than this threshold L_{max} , a reward as a function of both T_m and the water consumption is appended according to:

$$R = (T_{next} - T_{min}) - \left(\frac{L_{next}}{100}\right) - \left(\frac{Sol}{100}\right) \quad (3)$$

In which L_{next} represents the cumulative volume of water consumed and T_{min} represents the minimum midpoint temperature for that scenario. The solar and water parameters are divided by 100 to be normalized to the temperature range.

In each of the scenarios, the agent is punished to a different extent by adjusting T_{min} so as to create a linear decrease in the reward for decreasing temperatures and increasing solar energy and DHW consumption. Also, in case of high solar energy generation in combination with a high DHW consumption, the weight of the solar parameter is increased to enhance reheating for these scenarios. Hence, reheat cycles are preferred when there is a sufficient amount of solar energy and T_m is more than 2 °C below T_{start} . In the scenario of temperatures above 49.5 °C, the appended reward is always positive. This reward function is then applied to SARSA(λ) update which results in state-action pair values determining how favourable it is to reach a certain state at a particular action. The derived optimized control strategy is then applied to the vessel state model learned by the study of [9].

The comparison between the baseline control output and the optimized control output will be based on the same solar energy and water consumption profile derived from the data of June to September 2016 for the houses. This serves to compare the potential of reducing grid interaction with the Sarsa(λ) control in the same context.

3 Results

3.1 Q-Value Analysis

The Q-value function (indicating how good action ‘0’ is) for the summer months for a house are shown in Fig. 3. A negative value (blue) suggests the start of a reheat cycle whereas a positive (green/orange) value favors the idle state. It is evident from the figure that temperatures higher than 51.5 °C always favor the idle state. Temperatures between 45 °C and 51.5 °C favor the idle state when there is no to low solar energy production and decreases in value as a function of the total water consumed and the solar energy produced.

The most negative Q_0 values (most left corner) represent the least favorable states for the idle state. This shows that the algorithm learns to favor reheating when there is high solar energy generation. This effect is amplified when there is DHW consumption in the dwelling that causes the temperature to drop significantly. The Q-values in the rightmost corner of the plot show the most stable states which represent the states with high temperatures above 51.5 °C. The influence of the reward is visible this plot as the Q-values follow the strategy set in the reward function. Deriving from these results, it can be concluded that the algorithm learns the desired behavior to enhance the heating cycles during high solar energy generation events while considering the occupant DHW use.

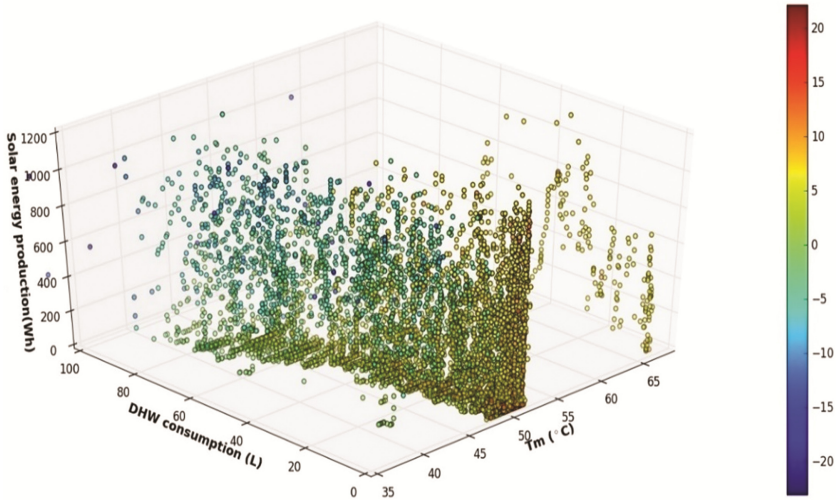


Fig. 3. 3D representation of the Q_0 values of house 1 summer months' data (Color figure online)

3.2 Linear Model Analysis

The reward function built in the algorithm is based on a linear function which suggests that the output of the algorithm should vary accordingly. Therefore, linear regression is applied to approximate the function of the algorithm. The same features as the algorithm were taken as the variables of the linear function. The comparison between the Sarsa Q -values and the linear Q -values are shown in Fig. 4. The clearest differences lay in the transition from positive to negative values. The values of the Sarsa algorithm show sharp incline and decline in certain points in time whereas the transition of the linear model progresses in a smoother fashion. This explains the increase of the difference shown, as an example, in the encircled area in Fig. 4 in which it clear that when the algorithm's output declined sharply, the linear output continued to increase steadily following the increase in solar energy production. Additionally, the fluctuations due to the solar energy

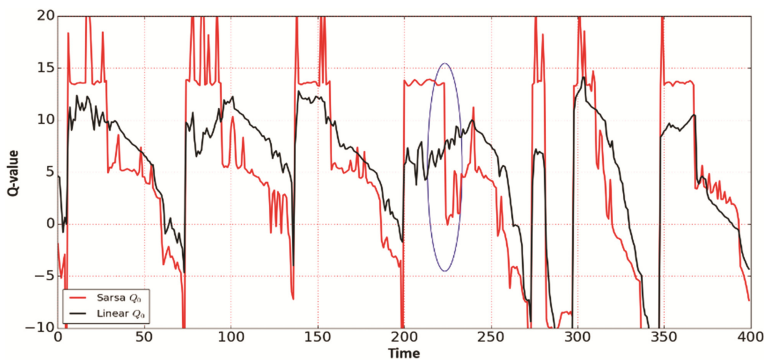


Fig. 4. Sarsa Q -values vs. linearly approximated Q -values

fluctuation are damped significantly in comparison with the algorithm. This results also in a slight delay in the transition to negative values. However, this does not pose a drawback to the model as the delay remains in the comfort level temperatures of higher than 45 °C. In fact, the damping effect of the model could be considered as an improvement to the algorithm since it is less responsive to sudden fluctuations in the solar energy production.

3.3 Optimization Results

The results of the energy consumption and the average daily number of reheat cycles for the six houses are shown in Table 1.

Table 1. Complete numerical results of the optimization control strategy for Jun-Sep

2 House nr.	Baseline total energy consumption (kWh)	Δ Times turned on	Δ Total energy withdrawn from grid (kWh)	Δ Self-consumption (kWh)	Self-sufficiency baseline and optimized (%)	Δ Comfort violation (nr of times)
1	133	162	-24	16	71%	-363
		(+36%)	(-63%)	17%	89%	(-88%)
2	94	19	-33	17	58%	-45
		(+6%)	(-82%)	31%	91%	(-53%)
3	124	123	-37	20	58%	-287
		(+28%)	(-70%)	28%	85%	(-84%)
4	44	44	-11	10	67%	-3
		(+23%)	(-73%)	35%	91%	(-72%)
5	53	135	-38	51	28%	-132
		(+67%)	(-99%)	348%	100%	(-70%)
6	91	217	-37	32	39%	-164
		(+74%)	(-67%)	89%	79%	(-73%)

The interaction of the ASHP load with the grid decreases significantly as a result of the optimization control strategy. Overall, it is evident from the results, that the energy withdrawn from the grid is reduced primarily due to the optimized timing of the reheat cycles during high energy generation hours. The total energy injected to the grid by the solar panels (indicated by the self-consumption column) is therefore reduced accordingly. The highest increase in self-consumed energy is 348% larger than for the baseline control strategy for house number 5 and the lowest for 15% for house number 1. This supports the significance of the consumption pattern of the building occupant on the optimization potential furthermore.

The combined total amount of enhanced onsite solar energy consumption for these houses accounts for 165 kWh in the period of June to September. This would have alternatively been injected to the grid causing large peaks in the grid during high generation hours. Moreover, the number of comfort violations (T_m lower than 45 °C and water

consumption is greater than 0L) is decreased significantly as a result of the optimization for all houses. This could be attributed to significant rise in the number of reheat cycles in order to maintain vessel midpoint temperatures of higher 49.5 °C during high generation hours as set in the reward function. Therefore, the reduction of the comfort violation is considered to account for part of the increase in total average daily required energy.

Perhaps the most notable effect of the optimization is visible in the self-sufficiency parameter. The optimization resulted in a remarkable increase in the self-sufficiency for all houses. House 5 even reached an increase of 70% in self-sufficiency to reach a total self-sufficiency of 99% of the load. The lowest increase is evident for house number 1 with an increase of 17% which resulted in a total self-sufficiency of 89%. The large difference could partly be explained by the high baseline self-sufficiency of house number 1 of 71% against that of only 28% for house 5. Intuitively, the lower the initial self-sufficiency, the larger the potential for optimization. Secondly, house 1 has significantly higher DHW consumption than the other houses which depletes the DHW storage tank much faster forcing it to reheat additionally during the low energy generation hours.

4 Discussion

4.1 Generalizability

The cluster of houses of this case-study comprise identical refurbished social housing dwellings. Therefore, the system size and parameters are all identical which will naturally result in similar control behavior. The only highly divergent variable on daily basis is the occupant behavior. Nevertheless, the algorithm learns the corresponding control strategy based on the individual data input. Therefore, this control strategy could also be applied to clusters with different and also divergent system parameters provided that the features used in this case study are available. In other words, the algorithm is generalizable and could be applied to any system that could provide these three parameters. The optimization potential will vary from one system to another based on the system parameters such as the insulation and the size of the water vessel, solar energy generation, type of dwelling, and the number of occupants of the dwelling.

4.2 Implications for the Occupants

The current Dutch net metering compensation scheme means selling excess solar energy is beneficial in all cases before 2020. Power sold to the grid is compensated against the total consumed power on annual basis. The excess power sold to the grid beyond the range of the total consumed power, is also compensated for a fair price determined by the DSO. This scheme is therefore stimulating consumers to invest in solar energy. As a result, consumers do not have to pay tax on their energy bill. This tax is estimated to account for 40 million Euros annually [15] which is funded by the Dutch tax system. It is therefore argued that this tax is paid for by tax-payers without solar panels [15]. Moreover, the scheme is not beneficial for grid operators on a long-term scale as a result

of the need for grid reinforcement. It is therefore planned to diminish this scheme by 2020.

All of this means that while currently reducing the power injected to the grid is not beneficial, it will become so according to the plans for the transition plan in 2020 [16]. Two main transition proposals are stated in the report state a reduced value for the excess sold power to the grid. The first proposal states a buying value of €0.22/kWh against a selling value of €0.06/kWh. The second proposal includes an additional feed-in tariff of €0.075 to the selling price. In both cases, on-site solar energy consumption will become more beneficial than selling to the grid. It is estimated by [16] that the monthly cost for nZEBs will increase, as a result of diminishing the net-metering scheme, by €40 given the current prices of solar technology and heat pumps and €30 considering the expected price reduction of these installations. Table 2 presents the total reduction of the summed energy injection and uptake for all six houses in the period of June to December, against the monetary value in both scenario 1 and 2 of the proposal of [16]. Deducting the money missed due to the reduction of solar energy selling from the gained money due to the reduction of electricity uptake, results in a total benefit between € 4- 24 in the first scenario and €1-9 in the second for the six houses for the period of June to December.

Table 2. Overview of reduction of energy uptake and injection and the corresponding monetary benefit for the two scenarios proposed in [16] to replace the net-metering policy in the Netherlands in 2020

House nr	Total reduced injection (kWh)	Total reduced uptake (kWh)	Δ Monetary benefit scenario 1 (€)	Δ Monetary benefit scenario 2 (€)
1	79	89	14.84	8.915
2	98	118	20.08	12.73
3	102	137	24.02	16.37
4	37	30	4.38	1.605
5	164	104	13.04	0.74
6	115	108	16.86	8.235

4.3 Implications for Grid Operators

The optimization results discussed in the previous sections show a decrease of energy injection to the grid of a maximum of 25 kWh on average for the houses in summer and a minimum of 10 kWh for the period of September to December. The reduction of injected power to the grid translates directly to a reduction in the necessary cost for grid reinforcement. The required grid reinforcement cost comes from the size of the transformers and cables required to accommodate a large increase solar energy generation due to an increase in nZEB concepts. Additional benefits for the grid operators relate to the real-time grid monitoring and steering role of the grid operators.

4.4 Learning Mechanism of RL

The main argument to use RL for this optimization is the learning aspect and generalizability of the method. The constantly changing solar and occupant behavior patterns create a disadvantage for fixed optimization approaches such as MPCs and rule-based control methodologies. To test and proof the learning mechanism of the RL algorithm used, average occupant behavior and monthly solar energy profiles were created from the individual profiles of the six houses. These profiles were then applied to the algorithm to simulate the results for each of the houses. The number of reheat cycles as a result of the averaged profiles optimization show remarkable results for being substantially amplified for houses with a low energy consumption profile. The large difference of the number of heating cycles between the individual and averaged profiles indicates the strong influence of the individual occupant behavior on the performance of the algorithm. This emphasizes the significance of the learning behavior of the algorithm as opposed to rigid approach of rule-based control optimization algorithms.

5 Conclusions and Recommendations

Deriving from the results and discussion, using RL could provide means for improving the performance of nZEB to comply to the label standards and in the same time reduce the pressure on the grid as a result of the increasing electrification of the built environment without posing a threat to the occupant comfort. Providing these means for the utility grid not only reduces the need/cost for grid reinforcement, but allows for more renewable energy technologies to be deployed in the residential built environment.

The results show an increase of individual self-consumption between 17% and 348% and self-sufficiency between 18% and 72% by applying DHW load shifting. This indicates a significant potential to reduce grid interaction to boost the implementation of RES in the residential built environment for the sake of increasing the share of renewables and also to improve the efficiency of the built environment by implementing more nZEBs. Additionally, considering the planned abolishment of the net-metering policy for solar power in 2020 in the Netherlands, reducing the grid interaction increases the monetary benefit for the residents. Moreover, this approach improves the monitoring potential of grid operators to avoid grid curtailment as a result of the increased electrification of the building loads and locally generated renewable energy. However, if the transient grid stability is aimed at as a main factor by the DSO, then the instantaneous power injection should be included as an additional variable and prioritized in the algorithm.

References

1. Rijksoverheid website, Onderzoeken Energie Gebouwde Omgeving, <https://www.rijksoverheid.nl/onderwerpen/onderzoeken-over-bouwen-wonen-en-leefomgeving/onderzoeken-energie-gebouwde-omgeving>. Accessed 07 Apr 2017

2. Van den Oosterkamp, P., Koutstaal, O., van der Welle, A., de Joode, J., Lenstra, J., van Hussen, K., Haffner, R.: The role of DSOs in a Smart Grid environment. ECORYS, Amsterdam/Rotterdam (2014)
3. Lawrence, T.M., Boudreau, M., Helsen, L., Henze, G., Mohammadpour, J., Noonan, D., Pateeuw, D., Pless, S., Watson, R.T.: Ten questions concerning integration smart buildings into the smart grid. In: *Building and Environment*, vol. 108, pp. 273–283. Elsevier (2016)
4. Yang, R., Wang, L.: Multi-objective optimization for decision-making of energy management in building automation and control. In: *Sustainable Cities and Society*, vol. 2, pp. 1–7. Elsevier (2012)
5. Klein, L., Kwak, J., Kavulya, G., Jazizadeh, F., Becerik-Gerber, B., Varakantham, P., Tambe, M.: Coordinating occupant behavior for building energy and comfort management using multi-agent systems. In: *Automation in Constructions*, vol. 22, pp. 525–526. Elsevier (2012)
6. Lund, P., Lindgren, J., Mikkola, J., Salpakari, J.: Review of energy system flexibility measures to enable high levels of variable renewable electricity. In: *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 785–807. Elsevier (2015)
7. Sossan, F., Kosek, A.M., Martinenas, S., Marinelli, M., Bindner, H.W.: Scheduling of domestic water heater power demand for maximizing PV self-consumption using model predictive control. Technical University of Denmark, Lyngby (2013)
8. Arteconi, A., Hewitt, N.J., Polonara, F.: Domestic demand-side management (DSM): Role of heat pumps and thermal energy storage (TES) systems. In: *Applied Thermal Engineering*, vol. 51(1–2), pp. 155–165. Elsevier (2013)
9. Kazmi, H., D’Oca, S., Delmastro C., Lodeweyckx, S., Corgnati, S.P. Generalizable occupant-driven optimization model for domestic hot water production in NZEB. In: *Applied Energy*, vol. 175, pp. 1–15. Elsevier (2016)
10. Ijaz Dar, U., Sartori, I., Georges, L., Novakovic, V.: Advanced control of heat pumps for improved flexibility of Net-ZEB towards the grid. In: *Energy and Buildings*, vol. 69, pp. 74–84. Elsevier (2014)
11. De Coninck, R., Baetens, D., Saelens, D., Woyte, A., Helsen, L.: Rule-based demand side management of domestic hot water production with heat pumps in zero energy neighborhoods. *J. Build. Perform. Simul.* 7, 271–288 (2014)
12. Ruelens, F.: Residential Demand Response Using Reinforcement Learning. Arenberg Doctoral School, Faculty of Engineering Science, KU Leuven, Leuven (2016)
13. Portaal homepage, <http://www.portaal.nl/stroomversnellingsoesterberg.aspx>. Accessed 02 July 2016
14. Pruissen, O.P., Kamphuis, I.G.: Grote concentraties warmtepompen in een woonwijk en gevolgen elektriciteitsnetwerk. Energy research Centre of the Netherlands (ECN), Petten (2010)
15. Rijksdienst voor Ondernemend Nederland, Saldering, zelflevering en tariefkorting, <http://www.rvo.nl/onderwerpen/duurzaam-ondernemen/duurzame-energie-opwekken/duurzame-energie/saldering-en-zelflevering>. Accessed 26 June 2016
16. Merosch, de effecten van en oplossingen voor aanpassing van salderingsregeling op NOM-woningen in 2020, <http://www.merosch.nl/download/CAwDEAwUUkBFXw==&inline=0>

Improving Time-Series Rule Matching Performance for Detecting Energy Consumption Patterns

Maël Guillemé^{1,2}(✉), Laurence Rozé¹, Véronique Masson¹, Cérés Carton², René Quiniou¹, and Alexandre Termier¹

¹ INSA, INRIA/IRISA, Université Rennes 1, Rennes, France
{mael.guillemé, laurence.roze, veronique.masson,
rene.quiniou, alexandre.termier}@irisa.fr

² Energiency, Rennes, France
ceres.carton@energiency.com

Abstract. More and more sensors are used in industrial systems (machines, plants, factories...) to capture energy consumption. All these sensors produce time series data. Abnormal behaviours leading to over-consumption can be detected by experts and represented by subsequences in time series, which are patterns. Predictive time series rules are used to detect new occurrences of these patterns as soon as possible.

Standard rule discovery algorithms discretize the time series to perform symbolic rule discovery. The discretization requires fine tuning (dilemma between accuracy and understandability of the rules). The first promising proposal of rule discovery algorithm was proposed by Shokoohi et al., which extracts predictive rules from non-discretized data. An important feature of this algorithm is the distance used to compare two sub-sequences in a time series. Shokoohi et al. propose to use the Euclidean distance to search candidate rules occurrences. However this distance is not adapted for energy consumption data because occurrences of patterns should have different duration. We propose to use more “elastic” distance measures. In this paper we will compare the detection performance of predictive rules based on several variations of Dynamic Time Warping (DTW) and show the superiority of subsequenceDTW.

1 Introduction

Nowadays, only around 60% of energy resources purchased by industrial enterprises are used to create added value. The remaining 40% are considered to be lost. To trace these losses, companies are increasingly equipped with sensors. Detecting dysfunctions from time series recorded by these sensors becomes a crucial part for reducing energy consumption.

Dysfunctions can be associated with specific patterns in time series (dysfunction signatures related to specific shapes of the time series). Thus, diagnosing the abnormal behavior of an observed system, machine, or plant, could be achieved

by locating patterns related to dysfunctions in time series. Our goal is to get typical patterns from experts (a pattern is a sub-sequence of a time series selected by the expert) and to return the smallest prefix to detect new occurrences of this typical pattern in a time series efficiently.

Predictive rules discovery is adapted to this task. A predictive rule has two parts: one is called antecedent and the other consequent. It means that if the antecedent is recognized, the consequent will occur before a delay. These predictive rules are easily understandable for the expert. Our goal is to discover such predictive rules in time series for detecting power consumption problems.

Classical predictive rule discovery algorithms discretize time series data [3, 7, 19]. But discretization is a difficult task and requires to choose the appropriate discretization method and to fine tune its parameters. To avoid this problem, in [15], Shokoohi et al. propose to extract predictive rules directly from time series without discretizing data. Shokoohi et al.’s approach can be summarized as follows. First, recurrent sequences are selected in a pre-processing step. Second, each of these sequences is split into two parts: an antecedent and a consequent. The split positions are chosen arbitrarily to yield a set of candidate rules. Next, for each rule, its occurrences, as pairs (antecedent, consequent), are retrieved in the time series. This is what we call *time series rule matching*. Then, these occurrences are used to compute a score for the rule, via a score function inspired by MDL [12] (Minimum Description Length). Finally the rule with the best score is returned.

Shokoohi et al.’s algorithm makes use of the Euclidian Distance for *time series rule matching*. However, the Euclidian Distance does not cope with distortions on the time axis (called here time *elasticity*) which is often the case with power consumption data.

We study the use of different extensions [11, 17, 18] of Dynamic Time Warping (DTW), a well-known distance measure [2], able to handle time *elasticity*, i.e. distortion in time. In this paper, we propose one of them, subsequenceDTW [11], to improve *time series rule matching*. We compare its performance (prediction earliness and precision) of time series rules elaborated with two different distance measures, the Euclidean distance and DTW.

Section 2 defines important concepts such as predictive rules and rules occurrences. Section 3 presents some related works. Section 4 presents our method for *time series rule matching*. In Sect. 5, we present detailed experiments for evaluating the performance of *time series rule matching* on a manually annotated energy-consumption time series, according to three distance measures: the Euclidean distance used in [15], the DTW, and subsequenceDTW.

2 Background

Sensors are used to capture energy consumption and to produce time series data. We consider here a time series T as defined in [5].

Definition 1. *time series*

A time series T is an ordered sequence of n real-valued variables $T = (t_1, \dots, t_n), t_i \in \mathbb{R}$. Values t_i are uniformly spaced in time.

An expert gives a typical pattern to be searched. Such a pattern is a sub-sequence of a time series. In order to detect the pattern as soon as possible in time series, a predictive time series rule is built from the expert pattern.

Definition 2. *predictive time series rule*

A predictive time series rule R for a pattern $p = \langle p_1, \dots, p_k \rangle$ is a pair (R_a, R_c) . The antecedent is $R_a = \langle p_1, \dots, p_i \rangle$. The consequent is $R_c = \langle p_{i+1}, \dots, p_k \rangle$. It means that if the antecedent is recognized, the consequent will occur before a delay.

From a time series, an expert is able to extract a set of sub-sequences considered similar. In energy consumption data, two similar sub-sequences can last longer or shorter. A pattern is shown in Fig. 1a and one of its occurrences is shown in Fig. 1b. As one can see, the occurrence is not identical to the searched pattern, in duration and in *shape*, but the expert still consider them as similar.

The rules discovery algorithm is based on the similarity of two sub-sequences. Different parameters are needed: a distance measure D between two sub-sequences, a constant th (threshold of similarity), and a *maxlag* (maximum delay between the antecedent and the consequent of a rule).

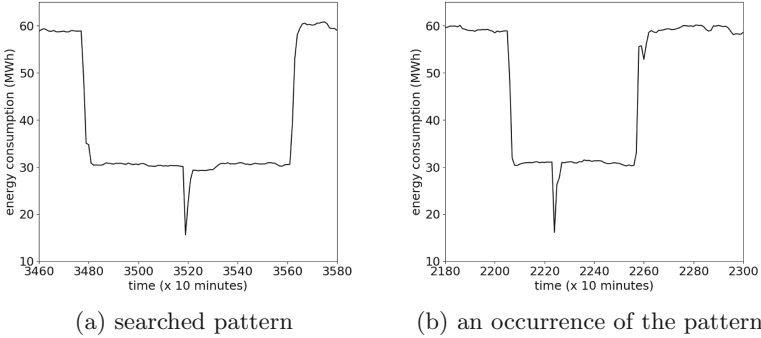


Fig. 1. Example of an occurrence (b) lasting shorter than the searched pattern (a) and having its bottom spike slightly shifted to the left.

Definition 3. *Similarity of two sub-sequences*

Two sub-sequences s_1 and s_2 are similar if and only if $D(s_1, s_2) \leq th$.

Definition 4. *Set of occurrences of a sub-sequence in a time series*

In a time series $t = \langle t_1, t_2, \dots, t_n \rangle$, a set of occurrences $O = \{o_i^{t_i}\}_{i \in START}$ of a sub-sequence $s = \langle s_1, s_2, \dots, s_k \rangle$ is the set of all sub-sequences o_i similar to s in t . $START[1..K]$ is the index of the occurrences where K is the length of O .

$o_i^{t_i} = \langle t_i, \dots, t_{i+l_i} \rangle$ is the i^{th} sub-sequence of s in t of size l_i starting at the time t_i with $l_i \in [1, n]$.

Definition 5. *Set of non overlapping occurrences in a time series*

A set of non overlapping occurrences of a sub-sequence in a time series t is a set of occurrences O such as $\forall i, j \in \text{START}, i < j \Rightarrow t_j > t_i + l_i$.

The former definitions are related to occurrences of sub-sequence. But, in rule discovery, these definitions can be extended to the case of occurrences of time series rules.

Definition 6. *Set of rule occurrences*

Let $R = (R_a, R_c)$ a time series rule, t a time series and maxlag a positive constant. A set of rule occurrences of R in t is a set O_R where $O_R = \{(a_1^{ta_1}, c_1^{tc_1}), \dots, (a_m^{ta_m}, c_m^{tc_m})\}$. Let $O_a = \{a_1^{ta_1}, \dots, a_m^{ta_m}\}$ and $O_c = \{c_1^{tc_1}, \dots, c_m^{tc_m}\}$ two sets of sub-sequence occurrences, O_R verify the following properties:

- O_a is a set of non overlapping occurrences of R_a in t .
- O_c is a set of non overlapping occurrences of R_c in t .
- $\forall i \in [1, m], 0 < tc_i - ta_i + la_i < \text{maxlag}$ with la_i the length of $a_i^{ta_i}$.
- $O_a \cup O_c$ is a set of non overlapping occurrences.

3 Related Work

Most approaches for discovering rules in real-valued time series rely on discretization of time series in order to apply symbolic rule discovery methods [1]. A classical method, in [3], proposes to apply K-means clustering on time series to obtain symbolic data. This preprocessing step has been commonly used in several other rule discovery methods, as in [6–8]. However, clustering sub-sequences in time series can be unsuitable in some situations [9]. For example, a single sub-sequence can occur several times in a cluster with narrow delays. This is called *trivial matches* [10].

In [14, 19], Piecewise Linear Aggregation (PLA) is used as discretization method, but Shokoohi et al. stated in [15] that this method is not adapted to *rule-based prediction*. Recently, Shokoohi et al. propose a discovery rule method which does not rely on a symbolic approach. This method is based on *time series rule matching*, consisting in finding rule occurrences in a time series (see Sect. 2). Algorithms of time series rely on a distance measure for comparing series.

Distance Measures. Many distance measures exist, but here we focus on measures that are able to handle similarity with distortion in time (see Sect. 2). In [4], Ding et al. compare various *elastic* distance measures. The Dynamic Time Warping (DTW) is one of the best methods because it is accurate and requires few parameters.

DTW is a method for speech recognition, which was brought to the data mining community in [2]. DTW is based on a dynamic programming approach

to align two sequences and computes the optimal distance between them. Several extensions of DTW exist: FastDTW [13] speeds up its computation, OBE-DTW [18] and ψ -DTW [17] relax constraints on the endpoints of compared series. Another extension, subsequenceDTW (subDTW) [11] finds several occurrences of a series in a time series in a single pass. Section 4.2 provides further details on DTW and subDTW.

4 Distance Measures for Time Series Rule Matching

This section describes how a *time series rule* is matched on a time series. Then, we present further details of two existing elastic distance measures: DTW and subDTW.

4.1 Time Series Rule Matching

Time series rule matching allows to find the occurrences of a *time series rule* in a time series. First a *time series rule* R is generated from the searched pattern s in the same way as in [15]. The searched pattern s is split into two parts: the first part is the antecedent R_a and the second part is the consequent R_c (see Sect. 2).

Time series rule matching is divided into two steps:

- step 1: identify all the antecedent occurrences R_a in the time series
- step 2: for each antecedent occurrence found in step 1, search the associated consequent R_c between this antecedent and the next one

Two steps are needed to retrieve first the sets of non-overlapping occurrences of the antecedent R_a . In both step 1 and 2, the search requires to compare antecedent or consequent with a sub-sequence of the time series. Different parameters are needed: a *maxlag* (the maximum length allowed between antecedent and consequent), a distance measure, a distance threshold, and for many distance measures, a sliding window to browse all the sub-sequences of the time series. Intuitively, the distance threshold should only authorize small distance values, because the smaller the distance is, the more similar the compared series are.

The *maxlag* is an expert knowledge and the choice of the distance measure D will be discussed later in Sect. 4.2. A single distance measure is used for the search of antecedent and consequent occurrences. However, since the antecedent and the consequent can have different sizes, a specific threshold distance and a specific window size are required for each of them. We called th_a (th_c) the distance threshold and w_a (w_c) the window size for the search of the antecedent R_a (consequent R_c).

To avoid asking a user for too many values, th_a , th_c , w_a and w_c are automatically computed from p_{th} and p_{window} . p_{th} is the given percent of the distribution of possible distance values between R_a (R_c) and sub-sequences of time series. p_{th} is easier to set than raw distance values. p_{window} is a percent of the length of the

series searched (respectively R_a and R_c). This parameter depends of the distance measure D . For example if D is the Euclidean distance the value of p_{window} must be 100% because Euclidean distance can only be calculated between series of the same size.

Algorithm 1 shows the *time series rule matching*. Step 1 is described by line 4, step 2 from line 5 to line 11.

Algorithm 1. Time series rule matching

Input: distance measure D , time series rule R , time series ts , $maxlag$ between antecedent and consequent, parameter for the window sizes p_{window} , parameter for the thresholds p_{th}

Output: O_R the set of occurrences of R in ts

```

1:  $O_R, O_a \leftarrow \emptyset$  //  $O_a$  is the set of antecedent occurrences found
2:  $w_a, w_c \leftarrow get\_window\_sizes(R, p_{window})$ 
3:  $th_a, th_c \leftarrow get\_thresholds\_distance(R, p_{th}, w_a, w_c)$ 
4:  $O_a \leftarrow get\_antecedent\_occurrences(R, ts, D, w_a, th_a)$ 
5: for all  $occ_a \in O_a$  do
6:    $consequent \leftarrow get\_consequent(R, ts, D, maxlag, w_c, th_c)$ 
7:   if  $consequent\_found$  then
8:      $occ_R \leftarrow create\_rule\_occ(occ_a, consequent)$ 
9:      $O_R.add(occ_R)$ 
10:  end if
11: end for
12: return  $O_R$ 

```

During step 1, the window of size w_a is slid at every successive position of the input time series. At each position, the sub-series included in the window is compared to the antecedent R_a . If the distance is less than the threshold th_a the position gives a candidate match. During this matching, we need to ensure that no *trivial matches* are returned [10] (see Sect. 3).

During step 2, for each antecedent occurrence found in step 1, the best candidate match of R_c is retrieved. The best candidate must not overlap the next antecedent occurrence. The delay between the current antecedent and this consequent must not exceed the $maxlag$.

As defined in Sect. 2, during the search of the antecedent and consequent occurrences, the distance measure must be able to compare series with distortion in duration. The next section presents two existing distances measures that are able to solve this problem: DTW and subDTW.

4.2 Distance Measures

The definition of DTW, given in [17], is used. To compute the optimal non-linear alignment between a pair of time series X and Y , with lengths n and m respectively, DTW typically bounds to some constraints: boundary condition, monotonicity condition and continuity condition. Many alignments satisfy all

the conditions. DTW performs a dynamic programming algorithm to compute the alignment between X and Y with minimum cost (DTW distance). The time and space complexities are $O(nm)$.

Here, X is the rule antecedent R_a or consequent R_c whereas Y is a sub-sequence of a time series. To get a set of Y , DTW requires a sliding window browsing the time series. However, this raises a new problem, how to well configure the size of the window? Indeed, a window that is too small could miss occurrences longer than the window, whereas a window that is too long, could cover several occurrences.

One extension of DTW, subDTW [11], proposes to solve the setting of the window. It is a distance measure, which offers the aligning property of DTW without the boundary condition. Moreover, it does not need any window. For a more detailed presentation refers to [11]. The integration of subDTW to our *time series rule matching* algorithm allows to remove w_a and w_c during the search of the antecedent and consequent occurrences.

5 Experimental Results

The experiment consists of comparing three distance measures (the Euclidean distance, DTW and subDTW) to match 20 generated rules in an energy-consumption time series that are annotated by an expert. This expert knowledge is provided by the French start-up Energiency, a provider of software analyzing energy-consumption to improve energy performance. In Sect. 5.2, the ability of each distance measure to retrieve the annotated rule occurrences is evaluated. In Sect. 5.3, the accuracy of distance measures is given with their related false alarm rate.

5.1 Experimental Setup

The experiments are performed on a real energy-consumption time series, called TS . Its frequency is ten minutes and its length is 26253 data points, which represents 6 months of energy consumption. The monitored system is an industrial plant composed of several machines. This kind of time series is commonly observed by experts.

An expert is asked to pick a sub-sequence of TS related to a interesting phenomenon to predict. Figure 2a shows a sequence of an abrupt decrease, a steady state, a down spike, a steady state and an abrupt return to the initial steady state. As presented in Sect. 4.1, a set of *time series rules* is first generated by splitting the sequence at n equidistant points. A rule is generated for each split point. To avoid extreme rules (for example, a time series rule with an antecedent with only one data point), the minimum length of the antecedent and consequent parts is set at 5% of the length of the sequence picked by the expert. In our experiment, the number of divisions is set to 19, that yields a set of 20 rules, and one of these rules is illustrated in Fig. 2b.

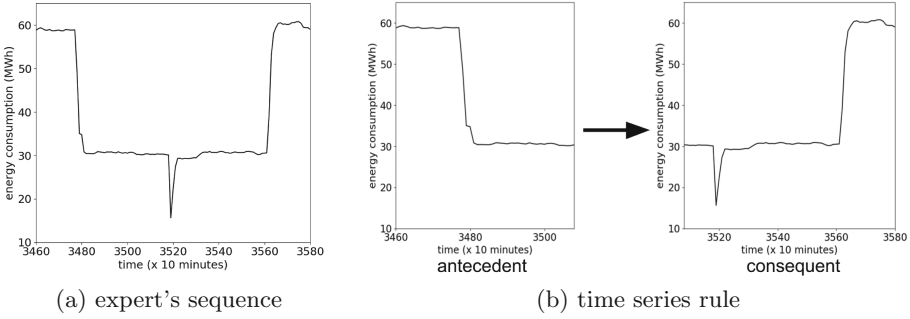


Fig. 2. Example of a rule (b), generated from the 8th split point in the expert's sequence (a).

For the experiment, the time series is manually annotated by an expert, according to the set of rules. As illustrated in Fig. 3, an annotation is an interval (Fig. 3a). For the expert, a rule occurrence must start in this interval to be considered as a match (Fig. 3b). In *TS*, 48 intervals are identified for each rule generated from the 1th, 2th, 3th, 19th and 20th split points. 65 intervals are found for each rule generated from the other split points. For each experiment, several values are tested for *maxlag* ($\{0h, 20h, 40h, 120h\}$) and *p_{th}* ($\{5\%, 10\%, 15\%, 20\%\}$). For the Euclidean distance, *p_{window}* is set to 100%, and for DTW, the values of *p_{window}* vary in $\{50\%, 100\%, 125\%\}$.

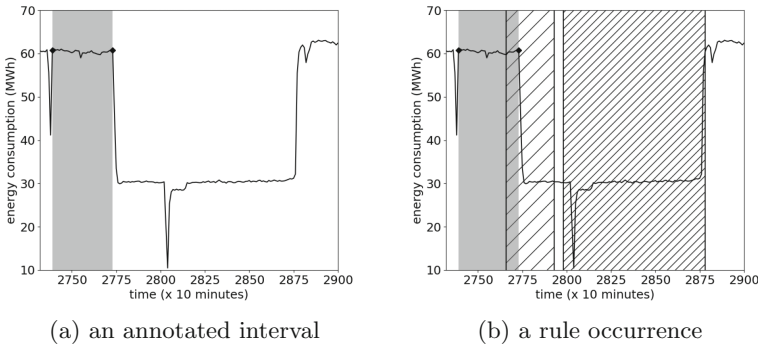


Fig. 3. Correct matching example of an annotated occurrence (light hatching for antecedent, heavy hatching for consequent).

5.2 Performance of Rule Matching

In this experiment, the ability of each distance measure to retrieve the annotated rule occurrences is evaluated for the 20 rules generated. Let *Actual True* (*T*) be the set of rule occurrences annotated in the time series *TS* by the expert, and

Predicted True (PT) the set of rule occurrences found by the rule matching algorithm where $PT = \{e|e \in TP \cup FP\}$, *True Positive TP* = $\{e|e \in PT \cap T\}$ and *False Positive FP* = $\{e|e \in PT \setminus T\}$. Precision Pr and recall Rc are then computed as follows:

$$Pr = \frac{|TP|}{|PT|} = \frac{|TP|}{|TP \cup FP|} \quad Rc = \frac{|TP|}{|T|}$$

Figure 4a shows the precision versus the recall for the three distance measures: the Euclidean distance, DTW and subDTW. The results of each distance measure are surrounded by a convex hull (covering the results for the 20 rules). Surrounding the results allows to compare more easily the global performance for each distance measure. It should be noted, that a small convex hull implies a small variation in performance among the 20 rules.

The results confirm that the Euclidean distance is not adapted to retrieve rule occurrences with time distortion. Indeed, the Euclidean distance compare series point-to-point without relying on an alignment algorithm. Moreover, only series with the same size can be compared. The precision of DTW is at least as high as the precision of subDTW, but the convex hulls show that the performance of DTW varies to an extended rate according to the rule. SubDTW has a better recall than DTW, thus more annotated rule occurrences are found.

Figure 4b shows the impact of the size of the sliding window on the performance of DTW. When the size of the window increases, the recall decreases. Whereas if the size of the window decreases, then the recall increases but the precision decreases. These results confirm that most of the annotated occurrences of the rules in TS are shorter than the rule. Moreover, if the size of the window is too large, no consequences are found in TS because the antecedent occurrence overlaps the entire rule occurrence. Defining the correct size of the window is critical for the performance of DTW. Several sizes of window can be tested but the execution time is strongly impacted.

This problem does not concern subDTW, because a window is not required. The recall of subDTW is higher than the recall of DTW for any size of the window. The use of subDTW to perform *time series rule matching* is best suited to time series with distortion in duration for rule occurrences.

Note that, after the experiments, the values of p_{th} is set to 20% and $maxlag$ is set to 120 h, which represents 5 days. The $maxlag$ is high because the time series are associated to an industrial environment, where breakdowns and maintenance can take several days.

5.3 False Alarms Rate

The next experiment focuses on the accuracy of the distance measure, according to the false alarm rate measured by the confidence. Let *Incomplete Predicted True (IPT)*, the set of the antecedent occurrences found by the *time series rule matching* algorithm $IPT = \{e|e \in TP \cup FP \cup A\}$ with A the set of antecedent occurrences found without consequent. The confidence Cf can be computed as follows:

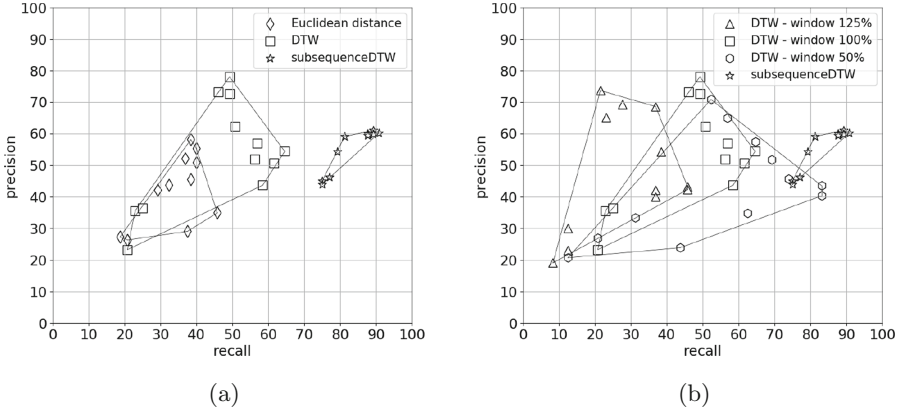


Fig. 4. Precision/recall plots for Euclidean, DTW and subDTW distances (a) and DTW with several window sizes (b).

$$Cf = \frac{|PT|}{|IPT|} = \frac{|TP \cup FP|}{|TP \cup FP \cup A|}$$

Figure 5 shows the confidence of each distance measure for the 20 rules generated. SubDTW has a higher confidence than DTW and the Euclidean distance, especially between the 4th split point rule and the 10th split point rule. Thus, subDTW triggers less false alarms.

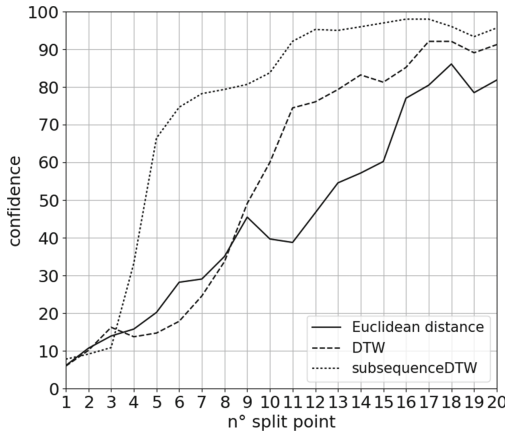


Fig. 5. Confidence plot for Euclidean, DTW and subDTW distances

Table 1 presents the confidence, the recall and the precision for the 4th and the 5th split point rules. These split point rules have a small antecedent allowing

Table 1. Recall, precision and confidence for 4th and 5th split point rules. Bold numbers highlight the best result in each column.

Distance measure	4 th split point rule			5 th split point rule		
	Recall	Precision	Confidence	Recall	Precision	Confidence
Euclidean distance	38%	58%	15%	40%	55%	20%
DTW (100%)	46%	73%	13%	49%	78%	14%
DTW (50%)	52%	70%	13%	57%	65%	18%
subDTW	89%	61%	33%	88%	60%	66%

to predict sooner the searched pattern. SubDTW keeps a good ratio between precision, recall and confidence for rules with a small antecedent.

6 Conclusion and Future Research

We have investigated *time series rule matching* in a rule discovery algorithm in energy consumption time series. In this context, *time series rule matching* must compare sub-sequences with distortion in time. We evaluate three existing distance measures on a real use case: the Euclidean distance used in [15], DTW a well-known elastic measure and subDTW, an extension of DTW which does not require a sliding window to browse the time series. The results confirm that the Euclidean distance is not adapted to find rule occurrences with different duration. Whereas, DTW and subDTW can handle the task. However, subDTW allows to find occurrences of rule with smaller antecedent without losing performance. Finding occurrences of rule with small antecedent means predicting energy consumption problems “as soon as possible”. Furthermore, subDTW does not need to set a sliding window.

There are lot of avenues for future work. We primarily focused on the constraint brought by using a sliding window when taking into account the duration of the occurrences in *time series rule matching*. However, as with all the methods relying on a distance measure, setting the right threshold is a very important and difficult task. That is even more important in our case, because we work from the shape which is highly dependent of the expert judgment. That’s why we propose to explore interactive learning, to learn the threshold by asking the expert to evaluate a sub-set of occurrences found, then to iterate until an acceptable threshold was found.

We should emphasize that *time series rule matching* is only a step of the rule discovery algorithm. In [15], *time series rule matching* is used to find occurrences of candidate rules. The score of each candidate rule is computed, from these occurrences, by a score function inspired by MDL. This score function works only for rule occurrences with the antecedent and consequent of the same duration of the candidate rule. With our new *time series rule matching* which find rule occurrences with different length, this condition doesn’t hold anymore. Hence, there is a need to find a new score function.

In the presented rule discovery algorithm, a whole rule is generated from a sequence given by an expert. However, the given sequence could be only used for the antecedent or consequent part of the rule. Then, the algorithm has to find in time series the other part of the rule.

A last avenue can be to extend rule discovery to a set of time series as proposed in [15]. The distance measures have to be adapted to multiple time series, but there is already a proposal in [16].

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: ACM SIGMOD Record, vol. 22, pp. 207–216. ACM (1993)
2. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop, Seattle, WA, vol. 10, pp. 359–370 (1994)
3. Das, G., Lin, K., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: KDD 1998, pp. 16–22 (1998)
4. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proc. VLDB Endow. **1**(2), 1542–1552 (2008)
5. Esling, P., Agon, C.: Time-series data mining. ACM Comput. Surv. (CSUR) **45**(1), 12 (2012)
6. Harms, S.K., Deogun, J., Tadesse, T.: Discovering sequential association rules with constraints and time lags in multiple sequences. In: Hacid, M.-S., Raś, Z.W., Zighed, D.A., Kodratoff, Y. (eds.) ISMIS 2002. LNCS (LNAI), vol. 2366, pp. 432–441. Springer, Heidelberg (2002). <https://doi.org/10.1007/3-540-48050-1-47>
7. Hetland, M.L., Sætrom, P.: Temporal rule discovery using genetic programming and specialized hardware. In: Lotfi, A., Garibaldi, J.M. (eds.) Applications and Science in Soft Computing. Advances in Soft Computing, vol. 24, pp. 87–94. Springer, Heidelberg (2004). <https://doi.org/10.1007/978-3-540-45240-9-13>
8. Jin, X., Lu, Y., Shi, C.: Distribution discovery: local analysis of temporal rules. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, pp. 469–480. Springer, Heidelberg (2002). <https://doi.org/10.1007/3-540-47887-6-47>
9. Keogh, E., Lin, J., Truppel, W.: Clustering of time series subsequences is meaningless: implications for previous and future research. In: Third IEEE International Conference on Data Mining, ICDM 2003, pp. 115–122. IEEE (2003)
10. Lin, J., Keogh, E., Lonardi, S., Patel, P.: Finding motifs in time series. In: Proceedings of the 2nd Workshop on Temporal Data Mining, pp. 53–68 (2002)
11. Müller, M.: Information Retrieval for Music and Motion, vol. 2. Springer, Heidelberg (2007)
12. Rissanen, J.: Modeling by shortest data description. Automatica **14**(5), 465–471 (1978)
13. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. **11**(5), 561–580 (2007)
14. Sang Hyun, P., Wesley, W., et al.: Discovering and matching elastic rules from sequence databases. Fundamenta Informaticae **47**(1–2), 75–90 (2001)
15. Shokoohi-Yekta, M., Chen, Y., Campana, B., Hu, B., Zakaria, J., Keogh, E.: Discovery of meaningful rules in time series. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1085–1094. ACM (2015)

16. Shokoohi-Yekta, M., Hu, B., Jin, H., Wang, J., Keogh, E.: Generalizing dtw to the multi-dimensional case requires an adaptive approach. *Data Min. Knowl. Disc.* **31**(1), 1–31 (2017)
17. Silva, D.F., Batista, G.E.A.P.A., Keogh, E., et al.: On the effect of endpoints on dynamic time warping. In: *SIGKDD Workshop on Mining and Learning from Time Series, II*. Association for Computing Machinery-ACM (2016)
18. Tormene, P., Giorgino, T., Quaglini, S., Stefanelli, M.: Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artif. Intell. Med.* **45**(1), 11–34 (2009)
19. Wu, H., Salzberg, B., Zhang, D.: Online event-driven subsequence matching over financial data streams. In: *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pp. 23–34. ACM (2004)

Probabilistic Wind Power Forecasting by Using Quantile Regression Analysis

Mehmet Baris Ozkan¹(✉), Umut Guvengir¹, Dilek Küçük¹, Ali Unver Secen¹,
Serkan Buhan¹, Turan Demirci¹, Abdullah Bestil¹, Ceyda Er¹,
and Pinar Karagoz²(✉)

¹ TÜBİTAK MAM Energy Institute, Ankara, Turkey
mehmet.ozkan@tubitak.gov.tr

² Middle East Technical University, 06531 Ankara, Turkey
karagoz@ceng.metu.edu.tr

Abstract. Effective use of renewable energy sources, and in particular wind energy, is of paramount importance. Compared to other renewable energy sources, wind is so fluctuating that it must be integrated to the electricity grid in a planned way. Wind power forecast methods have an important role in this integration. These methods can be broadly classified as *point wind power forecasting* or *probabilistic wind power forecasting* methods. The point forecasting methods are more deterministic and they are concerned with the exact forecast for a particular time interval. These forecasts are very important especially for the Wind Power Plant (WPP) owners who attend the energy market with these forecasts from day-ahead. Probabilistic wind power forecasting is more crucial for the operational planning of the electricity grid by grid operators. In this methodology, the uncertainty in the wind power forecast for WPPs are presented within some confidence. This paper presents a probabilistic wind power forecasting method based on local quantile regression with Gaussian distribution. The method is applied to obtain probabilistic wind power forecasts, within the course of the Wind Power Monitoring and Forecast Center for Turkey (RİTM) project, which has been realized by TÜBİTAK MAM. Currently, 132 WPPs are included in the project and they are being monitored in real-time. In this paper, the results for 15 of these WPPs, which are selected from different regions of the country, are presented. The corresponding results are calculated for two different confidence intervals, namely 5–95 and 25–75 quantiles.

Keywords: Wind power forecasting · Quantile regression · Probabilistic forecast

1 Introduction

Due to the well-known disadvantages of using fossil-fuels as energy sources, such as unsustainability and environmental pollution, the importance of the renewable sources has increased in recent years. Wind is one of the most important

renewable energy resources with its high potential. However, compared to other renewable energy sources, it has high variability and unstability. Therefore, reliable wind power forecasting is an important research problem.

The wind power forecasting methods reported in the literature can be mainly classified as *point forecasting* and *probabilistic forecasting* methods.

Point forecasting targets at exact forecasts for each time interval under consideration. These forecasts are quite important for Wind Power Plant (WPP) owners since they use these forecasts in the energy market and their economical returns depend on the accuracy of these forecasts [1]. There are several point forecasting methods presented in the literature and these forecasts are mainly based on statistical, physical and hybrid approaches.

Probabilistic wind power forecasting is more crucial for the electrical grid operators who manage the planning of the grid in the country. In this type of forecasting, some confidence interval such as (5–95 percentiles, 10–90 percentiles, and 25–75 percentiles) are given as forecasting results as shown in the sample forecast graph in Fig. 1. In this graph, the green line shows the point forecast and the gray lines show different confidence interval forecasts. According to these confidence intervals, the grid operators can carry out their day-ahead energy planning operations, as these probabilistic forecasting results help the operators get an overall idea about wind power generation in the country. Therefore, these probabilistic forecasts must be reliable enough for the safety of the electrical grid with WPPs.

In this paper, we present a probabilistic wind power forecasting method based on local quantile regression with Gaussian distribution. The proposed method has been applied to 15 WPPs that are within the scope of a large-scale wind power monitoring and forecast project (the RITM project), which currently covers 132 WPPs countrywide. The evaluation results attained using the real data for these 15 WPPs are quite satisfactory and promising.

The rest of the paper is organized as follows: Sect. 2 presents the related literature on the probabilistic wind power forecasting and Sect. 3 provides brief information about the (RITM) project as the proposed forecasting method is developed and applied within the scope of this project. Information regarding the data set is given in Sect. 4 and the proposed forecasting method is described in Sect. 5. The evaluation results are presented in Sect. 6 and finally, Sect. 7 concludes the paper with a summary and future work based on the current study.

2 Related Work

Probabilistic wind power forecasting is based on presenting an upper and a lower bound for the future probability of one or more events. This kind of approach is commonly used in various areas from economy to health or energy to stock market applications [2]. The aims of these models are to determine the safety intervals for the forecast and to present more reliable forecasts with less error rates.

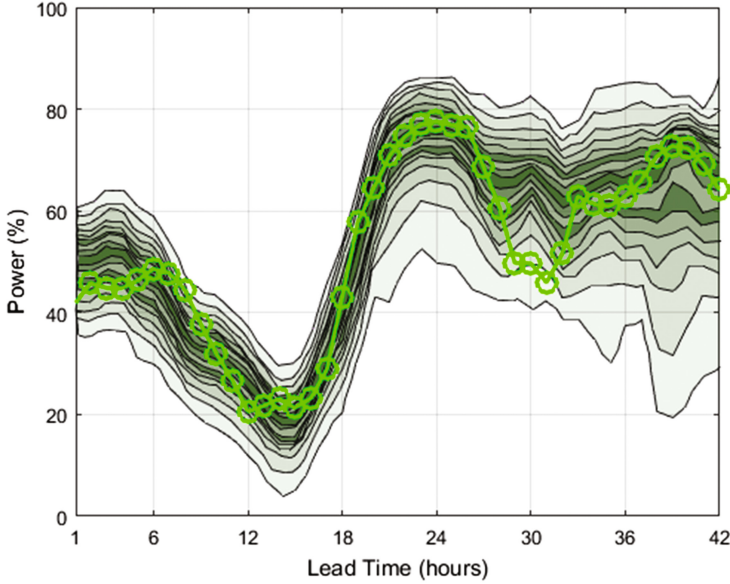


Fig. 1. A sample probabilistic wind power forecast graph

Most of the probabilistic wind power forecast models in the literature are based on quantile regression techniques [3]. In [4], Nielsen and Madsen first consider models for the mean of a random variable and later they consider models for the quantiles [4]. They implement their models for 25% and 75% quantiles. In the implementation of their model, they use *quantreg* function of the statistical tool, R [5]. According to their results, the upper and lower bounds fail in 500 out of 11000.

Juban and et al. [6] propose a probabilistic model with kernel density estimation. They test their model on three WPPs in France and their data resolution is 3-hour. According to their results, the proposed model has better results compared to continuous probabilistic evaluation and their model can be improved by optimizing the value of the smoothing parameters for different regions of the input hyperspace.

In [7], Sideteros and Hatziargyriou propose a probabilistic wind power forecast model by using neural networks and artificial intelligence techniques. Weather forecast are initially modeled by they start to using Self Organized Maps (SOM) and these forecasts are classified into three Radial Base Function Neural Networks (RBFNN) which have point forecasting information. In the final step, they use Gaussian Basic Functions at the hidden layer with generalized cross-validation and Bayesian interpolation. The model is tested on the data of two different WPPS along one year under the confidence intervals of 5–95, 10–90, 25–75 and 40–60. According to their evaluation results, authors report that their model can perform satisfactorily and very robust under different weather conditions and for different terrains.

Carpiana et al. propose a probabilistic wind power forecast approach which is based on Markov chain models [8]. The proposed method is reported to perform for short-term horizon, without requiring restrictive assumptions on the wind power probability distribution. They compare their obtained results from first and second order Markov chain models with respect to persistence model. According to their observations, the second order improves the forecast performance by reducing the prediction error.

A novel hybrid model is presented by Yan et al. in [9]. In their study, the authors present a hybrid deterministic–probabilistic method where a temporally local “moving window” technique is used in Gaussian process to examine the estimated forecasting errors. They have tested their models in two WPPs which are located in the USA and Ireland. The results are compared with each other, with respect to Mean Absolute Error (MAE) rate metric. The authors have demonstrated that the proposed hybrid deterministic–probabilistic method reduces the computational complexity during the learning and inference process compared to the standard Gaussian process. According to them, this factor is the most advantageous part of the proposed approach compared to classical Gaussian process model. Also, the proposed approach is shown to be effective for overcoming the time-varying characteristic at the two WPPs by producing a smaller forecasting error than those of the other models [9].

Three main statistical models in that area namely local quantile regression, local Gaussian Model and the Nadaraya–Watson estimator are compared in Bremnes work [19]. All of these models have several advantages and disadvantages. The Nadaraya–Watson estimator is generally preferred in very short term (0–6 h) probabilistic wind power forecasting. In this study, a hybrid model that uses both local quantile regression and Gaussian distribution is followed since it is more suitable for the short term (0–48 h) probabilistic wind power forecasting. Proposed model is mainly based on local quantile regression analysis but also in the determination of the bound coefficients Gaussian distribution is used.

3 The General Architecture of the RİTM System

In order to effectively monitor and forecast the wind power in Turkey, a large-scale project, namely, Wind Power Monitoring and Forecast Center for Turkey (Ruzgar İzleme ve Tahmin Merkezi (RİTM)) has started in 2012 [10]. The main aim of the project is to help the operators who control the electrical grid and carry out the necessary planning operations on the grid, by providing reliable day-ahead short-term and very short-term forecasts and monitoring the WPPs in real-time. Additionally, the point wind forecasts produced within the scope of the project are considerably important for the WPP owners because according to the accuracy of their day-ahead forecasts, they make profit or they lose in the daily energy market. They submit their 48-hour forecasts (in 1-hour resolution) to the energy market in day-ahead and their returns depend on these forecasts.

The general architecture of the overall monitoring and forecast system implemented within the scope of the project is presented in Fig. 2 [12]. In this architecture, the most important information source is the weather forecasts as they

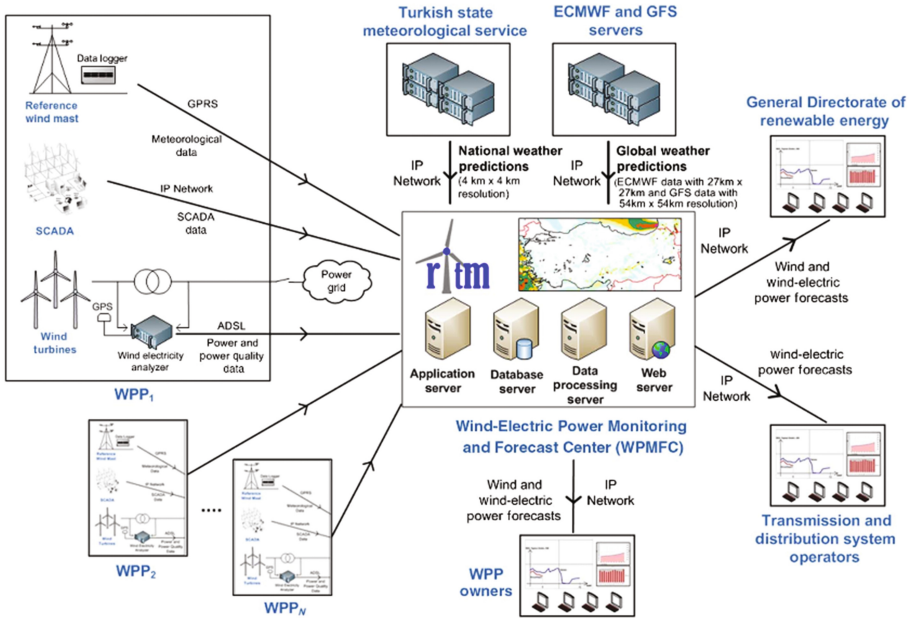


Fig. 2. The general architecture of the RITM system [12]

are crucial for producing the power forecasts. Therefore, in this large-scale system, the weather forecasts are obtained from three different sources, namely, Turkish Met Office (MGM) [13], European Centre for Medium-Range Weather Forecasts (ECMWF) [14] and Global Forecast System (GFS) [15] and the final power forecasts are obtained from the combination of these models. Apart from the weather forecasts, the real production values are also measured in the WPP areas by the power quality devices installed and are sent to the system center in every three seconds.

As the point forecasting facility of the project, a new data mining based forecast model, namely, Statistical Hybrid Wind Power Forecast Technique (SHWIP) has been implemented and it has been operation in the center [11]. By executing the forecast models the outputs are visualized by the user interface software and they are monitored by the WPP owners and the grid operator organization which regulates the energy sources in the country in real-time.

For the probabilistic forecast model described in the paper, the inputs are the intermediate point forecasts of the SHWIP model for three different weather forecast sources and the historical power data of the WPP for the most recent one-month period.

4 Data Set Description

The proposed probabilistic wind power forecast method is applied on the cluster-based intermediate point forecasts [11] obtained by using the individual weather forecasts of the aforementioned three sources, namely, MGM, GFS and ECMWF. In order to construct the model, historical real production values of the WPPs are used in the training stage. While constructing the model, the training period is selected as 30 days.

Table 1. Data sets used in the model

Name	Unit	Resolution
Historical Real Power Data	Megawatt	Hourly(30 days)
Historical MGM Clustered Power Forecast Data	Megawatt	Hourly(30 days)
Day-Ahead MGM Clustered Power Forecast Data	Megawatt	Hourly(48 h)
Historical GFS Clustered Power Forecast Data	Megawatt	Hourly(30 days)
Day-Ahead GFS Clustered Power Forecast Data	Megawatt	Hourly(48 h)
Historical ECMWF Clustered Power Forecast Data	Megawatt	Hourly(30 days)
Day-Ahead ECMWF Clustered Power Forecast Data	Megawatt	Hourly(48 h)

The data structure used in constructing the model is presented in Table 1. The unit for all of the data types is Megawatt and they are in hourly resolution. The historical data sets consist of 30×24 hourly data for the most recent 30 days when the model is executed. The day-ahead forecast tuples are the 48-hour point forecast tuples of the day for which the probabilistic power forecasts are produced.

5 Proposed Method

The aim of the proposed method is to find each hour interval forecast independently by using the historical forecast and power data. Initially, the maximum distance between the current hour forecast and historical forecast data is found as shown in Fig. 3. Then the u matrix of size $48 \times 30 \times 24$ is calculated as given in Eq. 1. Finally, the weight vector w of size $48 \times 30 \times 24$ is calculated according to the weight function given in Eq. 2 [8] for each 48 h independently by using the u vector. In all of the formulas, i index ranges from 1 to 48 and j index ranges from 1 to 30×24 .

$$u_{ij} = \frac{TrainingForecast_j - Forecast_i}{maxDistance_i} \quad (1)$$

$$w_{ij}(u_{ij}) = \begin{cases} (1 - u_{ij}^3)^3, & \text{if } u_{ij} < 1 \\ 0, & \text{if } u_{ij} = 1 \end{cases} \quad (2)$$

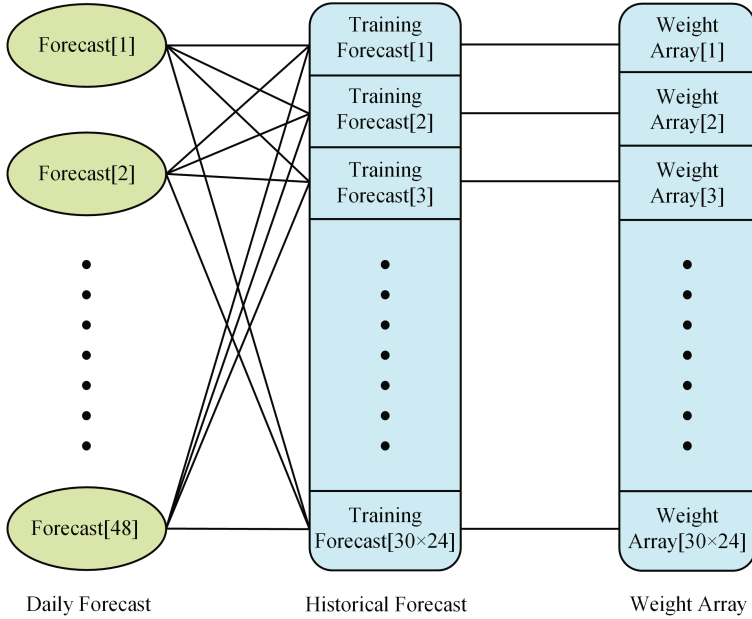


Fig. 3. Determination of weight array

$$\sigma_i = \sum_{j=1}^{30 \times 24} (Power_j - Training_j)^2 \times w_{ij} \tag{3}$$

For each element in the training forecast, u value is determined according to Eq. 1 and the u value used in the weight function, which is given in Eq. 2 to calculate the weight array in the size of 30×24 for each 48 test hour. In the following step, by using this weight array and historical power data, the variance of the each hour is determined as given in Eq. 3.

$$bound5_i = Forecast_i - BOUND5_{COEFF} \times \sqrt{\sigma_i} \tag{4}$$

$$bound25_i = Forecast_i - BOUND25_{COEFF} \times \sqrt{\sigma_i} \tag{5}$$

$$bound75_i = Forecast_i + BOUND25_{COEFF} \times \sqrt{\sigma_i} \tag{6}$$

$$bound95_i = Forecast_i + BOUND5_{COEFF} \times \sqrt{\sigma_i} \tag{7}$$

In the final step, by using the total variance and bound coefficients, the interval forecast for that particular hour is calculated as stated in Eqs. 4, 5, 6 and 7. While determining the bound coefficients, Gaussian distribution is assumed [16] and Bound5 coefficient is selected as 1.6449 and Bound25 coefficient is selected as 0.6745.

The process described above is conducted for each 48-hour forecast tuple independently and at the end of the process, two-day interval forecasts are determined for 5–95 and 25–75 confidence intervals.

6 Evaluation Results

This section presents the evaluation results of the applied method on the WPPs. The results are given for 15 WPPs, which are selected from different regions of Turkey. For testing the reliability of the model, at least 93% of the real power

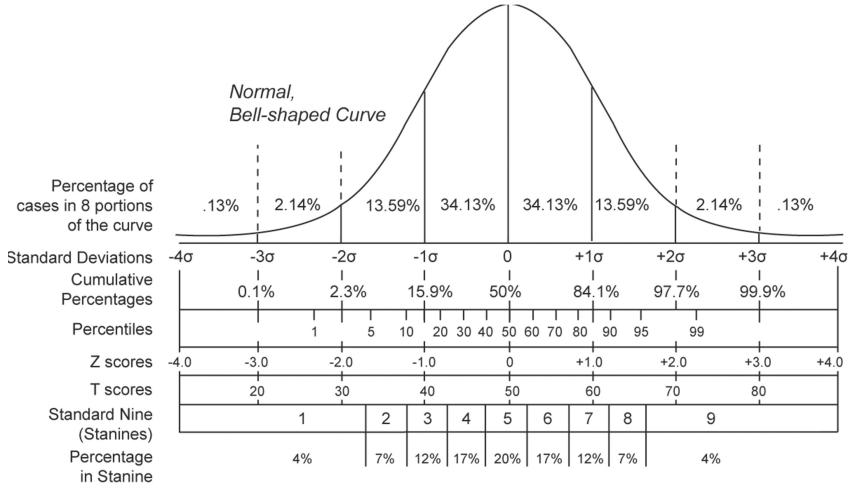


Fig. 4. Gaussian distribution with sigma levels

Table 2. The evaluation results

WPP	5–95 percentile(%)	25–75 percentile(%)	Region	Capacity (MW)
WPP1	91.69	74.39	Marmara	114
WPP2	92.38	73.01	Marmara	60.5
WPP3	93.33	77.41	Marmara	142.5
WPP4	91.57	73.5	Marmara	93
WPP5	90.36	76.55	Marmara	50
WPP6	90.96	68.4	Aegean	50
WPP7	91.46	73.84	Aegean	240
WPP8	90.59	74.53	Aegean	120
WPP9	91.68	76.86	Aegean	47.5
WPP10	91.91	77.19	Mediterranean	48
WPP11	92.68	72.37	Mediterranean	76
WPP12	90.87	79.48	Mediterranean	42.5
WPP13	87.6	69.47	Black Sea	42
WPP14	82.46	67.97	Central Anatolia	168
WPP15	93.34	84.45	Southeastern Anatolia	25

value of the WPP must be in 5–95 percentile and 69% of the power value must be in 25–75 confidence intervals according to the bell curve given in Fig. 4 [17, 18].

The corresponding evaluation results are presented in Table 2 together with several other characteristics regarding the WPPs such as their installed capacities. As the test period, the four-months region between January 1, 2017 and May 1, 2017 is selected.

According to results presented in Table 2, in 12 of 15 WPPs, more than 70% of the real production value in the test region is in 25–75 confidence interval. Similarly in 13 of 15 WPPs, production value remains in the forecast band of 5–95 more than 90% of the cases. The best results are obtained in WPP3 and the worst results are obtained for WPP14. For these WPPs, sample time series charts are presented in Fig. 5 and Fig. 6, respectively. In these figures, the green line represents the real production data and gray intervals denote the 25–75 and 5–95 interval forecasts.

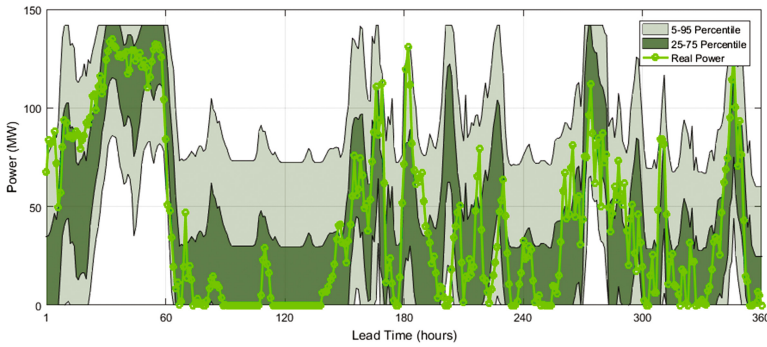


Fig. 5. Probabilistic wind power forecasts for WPP3

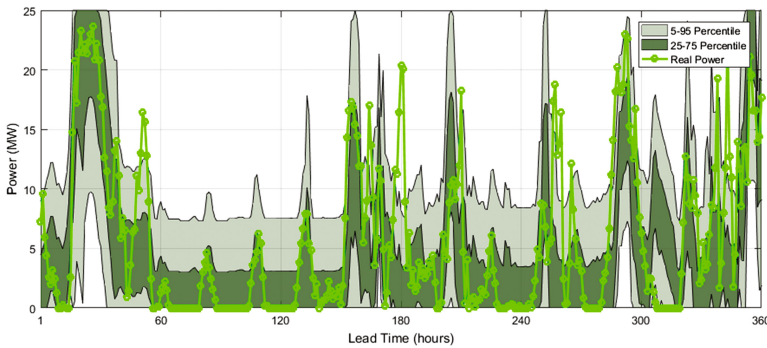


Fig. 6. Probabilistic wind power forecasts for WPP14

7 Conclusion

In order to benefit from the renewable energy sources effectively, there is a need for good planning. Especially if this source is so variable such as wind, then a reliable forecast system is inevitable for the convenient planning the energy grid. In this work, the probabilistic wind power forecasting method implemented and applied within the course of the RİTM project is presented. According to the evaluation results, the method attains reliable results especially for the 25–75 confidence intervals and the results for the 5–95 intervals are also applicable. The training duration for constructing model is selected as 30 days. Future work based on the current study includes investigating the effects of different training durations on the model performance. In addition, apart from the historical power, the effects of other variables such as temperature and pressure in the WPP area can be investigated.

Acknowledgment. This work is conducted within the scope of RİTM project (with number 5122807), which is directed by Energy Institute of TÜBİTAK MAM. We would like thank all of the partners of the RİTM project especially to Renewable Energy General Directorate of Turkey (YEGM).

References

1. Hodge, B.-M., et al.: The value of improved short-term wind power forecasting. In: National Renewable Energy Laboratory (NREL), Golden, CO (2015)
2. Kou, P., Gao, F., Guan, X.: Stochastic predictive control of battery energy storage for wind farm dispatching: using probabilistic wind power forecasts. *Renew. Energy* **80**, 286–300 (2015)
3. Bremnes, J.B.: Probabilistic wind power forecasts using local quantile regression. *Wind Energy* **7**(1), 47–54 (2004)
4. Nielsen, H.A., Madsen, H., Nielsen, T.S.: Using quantile regression to extend an existing wind power forecasting system with probabilistic forecasts. *Wind Energy* **9**(1–2), 95–108 (2006)
5. <https://www.r-project.org/>
6. Juban, J., Siebert, N., Kariniotakis, G.N.: Probabilistic short-term wind power forecasting for the optimal management of wind generation. In: *Power Tech, 2007 IEEE Lausanne* (2007)
7. Sideratos, G., Hatziaergyriou, N.D.: Probabilistic wind power forecasting using radial basis function neural networks. *IEEE Trans. Power Syst.* **27**(4), 1788–1796 (2012)
8. Carpinone, A., et al.: Very short-term probabilistic wind power forecasting based on Markov chain models. In: *IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)* (2010)
9. Yan, J., et al.: Hybrid probabilistic wind power forecasting using temporally local Gaussian process. *IEEE Trans. Sustain. Energy* **7**(1), 87–95 (2016)
10. <http://www.ritm.gov.tr/root/index.php>
11. Ozkan, M.B., Karagoz, P.: A novel wind power forecast model: statistical hybrid wind power forecast technique (SHWIP). *IEEE Trans. Ind. Inform.* **11**(2), 375–387 (2015)

12. Terciyani, E., et al.: Enhanced nationwide wind-electric power monitoring and forecast system. *IEEE Trans. Ind. Inf.* **10**(2), 1171–1184 (2014)
13. <http://www.mgm.gov.tr/>
14. <http://www.ecmwf.int/>
15. <http://weather.unisys.com/gfs/>
16. Brown, M.B., Forsythe, A.B.: Robust tests for the equality of variances. *J. Am. Stat. Assoc.* **69**(346), 364–367 (1974)
17. Herrnstein, R.J., Murray, C.: *Bell curve: Intelligence and class structure in American life*. Simon and Schuster, New York (2010)
18. <https://en.wikipedia.org/wiki/Standardscore>
19. Bremnes, J.B.: A comparison of a few statistical models for making quantile wind power forecasts. *Wind Energy* **9**(1–2), 3–11 (2006)

Identifying Representative Load Time Series for Load Flow Calculations

Janosch Henze^{1(✉)}, Tanja Kneiske³, Martin Braun^{2,3}, and Bernhard Sick¹

¹ Intelligent Embedded Systems, University of Kassel, Kassel, Germany
{[jhenze](mailto:jhenze@uni-kassel.de),[bsick](mailto:bsick@uni-kassel.de)}@uni-kassel.de

² Energy Management and Power System Operation,
University of Kassel, Kassel, Germany
martin.braun@uni-kassel.de

³ Fraunhofer IWES, Kassel, Germany
{[tanja.kneiske](mailto:tanja.kneiske@iwes.fraunhofer.de),[martin.braun](mailto:martin.braun@iwes.fraunhofer.de)}@iwes.fraunhofer.de

Abstract. Power system analyses increasingly use annual time series for temporal and spatial assessment of operational and also planning aspects. These analyses are often limited due to the computational time of the large amount of load flow calculation. By introducing algorithms which are capable of generating shorter and representative time series of measured load or power generation time series, the calculation time for load flow calculations can be reduced. We present a method which is capable of extracting features from the time series and use those features to create a representative time series. Furthermore, we show that our method is capable of maintaining the most important statistical features of the original time series by applying a Fisher-Pitman Permutation test.

Keywords: Representative time series · Feature extraction · Clustering · Load flow · Grid expansion planning

1 Introduction

The amount of collected data due to smart grids is vast. Data producing devices within smart grids range from power generators, e.g., wind turbines or solar power facilities, to households with smart meters. Each of these devices generates a steady flow of data which can be processed to determine the future power production, to plan grid expansions, simulate future energy markets, or just to assess the dynamics of the power grid itself. The time resolution of these data can vary. Typically, in the industry data are collected in intervals of 15 min. However, also a resolution of 1 min or even of seconds, could be imagined in a future application which increases the amount of data by a factor 15 or even 900.

An aggregation of data does not necessarily reflect the statistical characteristics of the time series data. Depending on the application, an aggregation has to maintain different statistical characteristics of the input data. Applications such as the planning of battery charging strategies, the planning future power grid

expansions, or simulations of the power market, the input data need to reflect the most important statistical characteristics of the data for the given task. For example, grid expansion planning has to focus on extreme values and their probabilistic characteristics, whereas simulation of the power market needs to have a view on all possible result variations within the grid including outliers.

In this article, we propose a method to create a shorter representative time series from a long input time series. The goal is to reduce the runtime of following analyses, like load flow calculations. The method analyzes load time series, calculates features of segments of the time series and then clusters the extracted feature vectors. By sampling from the clusters, we can select similar time series segments and create a representative time series of the input time series. Furthermore, we show that the created representative time series has similar statistical properties as the input time series when used in a load flow calculation.

The remainder of the article starts with a discussion of the related work in Sect. 2 followed by a description of the segmentation and feature extraction algorithms used to determine the representative time series in Sect. 3. Afterward, we compare the results of load flow calculations using the representative load time series and the original load time series in Sect. 4. The paper concludes with a discussion of key results and an outlook on future work in Sect. 5.

2 Related Work

Creating representations of a time series to reduce the calculation for further analyses is an important research topic. Processing large amounts of data is time-consuming, by pre-processing the data to create representative time series with a lower amount of data and the same statistical characteristics, it is possible to improve the computation time of subsequent algorithms. As a good time series representation depends on the application, different algorithms have been developed.

In [1], time overlaps are used during the segmentation of annual data for parallel simulation runs. Showing that using weekly segments with an overlap from one day to the previous week improves parallel simulation performance.

A proposed general approach preserving different statistical properties of time series is given by [2]. The authors provide a method using log-spectrum smoothing to conduct inference on time series sampled by different methods, allowing them to reduce the amount of data and increasing the performance. In [3], an indication is given regarding the amount of data that needs to be preserved to create representative time series, which capture temporal variability to preserve statistical properties of the time series.

Basic algorithms to create representative time series include algorithms like k-means for clustering similar time segments as discussed in [4] to approaches that use linear optimization to find similar segments as used in [5].

The k-means clustering approach in [4] allows generating clusters to represent the gross electricity demand and net demand for wind output. The authors were able to reduce the runtime of their subsequent simulations by using these clusters.

The optimization based approach in [5] identifies representative time series, by weighting representative time series based on their performance in load flow applications, e.g., grid expansion planning.

Related work focusing on clustering energy and load data include [6–8]. The authors of [6] use clustering methods, such as k-means or hierarchical clustering, to identify typical loads of electricity customers. [7] uses a feature extraction process, where simple features such as mean and standard deviation are extracted to identify similar customers and use their respective load curves to optimize their load. In [8], a feature extraction approach similar to the one proposed in [7] is used to identify unusual energy consumption in buildings.

Creating a representation of a time series, especially for load time series, is an active field of research. So far, feature-based approaches have been used to analyze household related data to identify unusual behavior. The impact of feature based clustering on load flow applications has not been analyzed in detail so far.

3 Proposed Method

This section explains the process of obtaining representative time series from an evenly spaced input time series, i.e., equidistant in time. Representative time series T_{rep} segments of the original univariate time series $T = \{t_0, t_1, \dots, t_n\}$. To create representative time series we propose the following consecutive steps as shown in Fig. 1. The steps include preprocessing as explained in Sect. 3.1, followed by segmentation and feature extraction in Sect. 3.2, and finally the creation of the representative time series by sampling from a created cluster representation followed by the postprocessing as explained in Sect. 3.3.

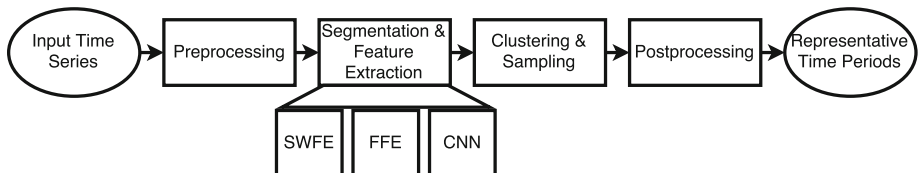


Fig. 1. A flow chart of the proposed algorithm for the generation of representative time series using feature vectors. The algorithms for feature extraction are Sliding Window Feature Extraction (SWFE), Fast Feature Extraction (FFE) and a Convolutional Auto Encoder Neural Network (CNN).

3.1 Preprocessing

The preprocessing consists of two substeps, the first step is optional and removes the seasonal influence and the second step normalizes the time series. Both preprocessing steps are applied on each time series to allow an independent handling

of each time series. The preprocessing parameters are saved to allow to transform the representative time series to a similar level as the input time series.

An input time series which is longer than a year most likely has seasonal influences. Hence, the first pre-processing step removes these seasonal influences. The time series is resampled to a resolution of one week using the mean of the data points. By subtracting these resampled values from the corresponding weekly values, the seasonal influence on the time series is removed.

The second preprocessing step normalizes the time series T by applying the standard z-score.

$$Z = \frac{T - \mu}{\sigma} \quad (1)$$

Equation 1 shows the standard z-score normalization, where Z is the normalized time series, μ is mean value of T and σ is the standard deviation of T both are calculated for the whole time series. The normalized time series $Z = \{z_0, z_1, \dots, z_n\}$ has a mean of 0 and a standard deviation of 1.

3.2 Segmentation and Feature Extraction

After the preprocessing, the normalized time series is segmented and the feature vectors are extracted. For some algorithms, the segmentation is dependent on the feature extraction algorithm. Hence, the segmentation method is described with the corresponding feature extraction algorithm. In our work, we propose three different feature extraction algorithms.

1. Simple Sliding Window feature extraction (SWFE),
2. Fast Feature Extraction (FFE) and
3. feature extraction based on Convolutional Autoencoder Neural Networks (CNN).

Each of these algorithms produces a feature vector for each analyzed segment. The feature vectors for each time series are clustered in a subsequent step.

Simple Sliding Window Feature Extraction. The segmentation of the SWFE algorithm is shown in Fig. 2a. The algorithm uses a fixed sliding window of size m . Hence, the first window starts at time step $t_1 = 0$ and reaches $t_2 = t_1 + m$. Afterwards, the window is shifted by distance of s . Hence, the second window then starts at $t_3 = t_1 + s$ and reaches $t_4 = t_3 + m$. In our application s is set to m , so the time series is always shifted by m and therefore creates non-overlapping segments.

For each window, the mean, variance, minimum, maximum, and the median are extracted as features and stored for clustering. Furthermore, the start position and the window length m are stored, to easily allow to obtain the corresponding time series segment to a feature vector.

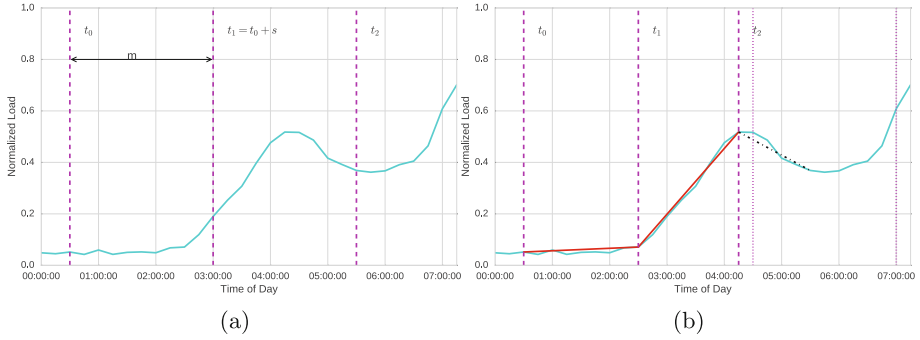


Fig. 2. Schematic segmentation for the different feature extraction algorithms: SWFE and CNN (a), FFAL (b). In (c) the solid red line shows the representation within the growing window. The dotted vertical line shows the current sliding window, which is used to select segmentation points. The black dash-dotted line shows the current polynomial fit of the growing window.

Fast Feature Extraction. The FFE uses a sliding window and a growing window technique [9]. Figure 2b depicts the mechanisms of the FFE algorithm. The sliding window acts as explained in Sect. 3.2 with $s = 1$. The growing window starts with a window size of 1 and is iteratively increased. A polynomial approximation with a degree of k is fitted in both windows. The coefficients of the orthogonal expansion of the approximating polynomial define the feature vector for the FFE algorithm. As explained in more detail in [9], the coefficients of the polynomial approximation represent characteristics of the analyzed time series, e.g., mean or curvature.

Segmentation is performed if the fit of the polynomial approximation reaches a certain threshold, e.g., the approximation error is too high, or the curvature reaches a user-specified limit. At the segmentation point, the coefficients of the polynomial approximation are stored alongside with the approximation error, the current window length of the growing window, and the segmentation point. Afterward, the gathered information resets, and the algorithm tries to find the next segment [10]. In our implementation, the sliding window approximation is used to find appropriate segmentation points, and the growing window approximation is used to determine the feature vector.

Convolutional Auto Encoder Neural Networks. An autoencoder (AE) learns the relationship between input data and output data through a bottleneck. The input data are the same as the output data, allowing the autoencoder to learn a view on the data at the bottleneck which is similar to principal component analysis. By splitting the autoencoder into an encoder and decoder part, it is possible to transfer the data into a low-dimensional representation with the encoder and decode this representation using the decoder.

A convolutional neural network (CNN) learns a filter function on the input training data, in our case different time series segments. Thus, allowing to extract different features of the analyzed subsampled time series. Furthermore, similar subsampled time series produce similar features as input for the subsequent layer. By using convolutional inputs and combine them with the structure of the autoencoder we obtain a feature extraction algorithm.

The CNN AE algorithm uses the same sliding window method as the SWFE. The inputs for the training, as well as the outputs of the neural network, are the data within the current sliding window. After the training of the neural network, the network is separated into encoder and decoder parts. The feature vectors are obtained by taking the output from the encoder, i.e., a low dimensional representation of the data at the bottleneck of the CNN AE. Those extracted feature vectors are then used for clustering.

The CNN AE algorithm has the obvious drawback that the neural network needs to be trained before it can be applied to extract feature vectors. The training increases the run time and the amount of data needed.

3.3 Creation of Representative Time Series

To create representative time series using a set of feature vectors we need to cluster the feature vectors. Similar feature vectors will be clustered to represent similar elements within the input time series.

To create a representative time series from these clusters, we sample N data points of those clusters and reconstruct the associated time series segment. Reconstructing time segments of the SWFE or FFE feature vectors is done by using the associated segmentation point and length which are used to obtain the associated time series segment from the normalized input time series. The CNN AE reconstructs the input data by applying the sampled feature vectors as input to the decoder. Using the feature vectors as input to the decoder allows calculating the associated time series segment. Afterward, all samples are concatenated to create the output time series, and the preprocessing steps are reversed, i.e., reversing the steps of the standard z-score normalization and adding the corresponding seasonal influence.

The sampling of feature vectors influences the representation of the time series. Depending on the subsequent analysis different sampling strategies might be selected. E.g., if the analysis focuses on obtaining information about extreme values, such as minimum and maximum, outliers or elements of the convex hull of the cluster should primarily be sampled.

4 Evaluation

We evaluated the performance of the representative time series by applying them to load flow calculation to determine explanatory power in applications such as grid expansion planning. We first describe our evaluation method in more detail in Sect. 4.1 and show our results in Sect. 4.2.

4.1 Method

To evaluate the method we proposed in Sect. 3, we created 100 representative time series for 100 different household load time series for all proposed feature extraction algorithms. The representative time series and the input time series were analyzed using load flow calculations. The results of those load flow calculations were validated using the Fisher-Pitman [11] Permutation test.

The input load time series were artificially created time series with a length of a year and a resolution of 15 min. The load profile generator [12], we used to create the input time series, uses detailed information on German households to create load time series as close as possible to the current housing situation in Germany.

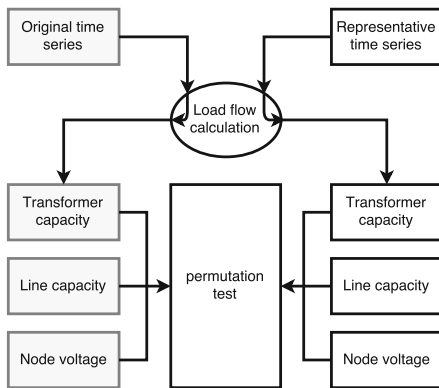


Fig. 3. Overview of the validation process.

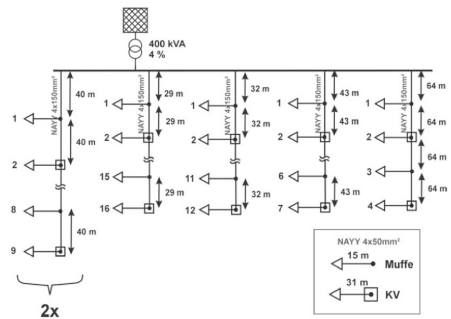


Fig. 4. Generic distribution grid used for the validation.

Figure 3 illustrates the validation process. The validation determines if the representative time series has the same statistical characteristics as the input time series. The validation evaluates the grid parameters calculated by the load flow analysis and not the representative time series itself. Hence, we can identify if the representative load time series lead to similar statistical characteristics as the input load time series

All algorithms use a sliding window size of 12 h to segment the time series. The sliding windows had no overlap, except for the FFE. As clustering algorithm, we used HDBSCAN [13], a hierarchical version of the density based clustering algorithm DBSCAN. From the resulting clusters of feature vectors, we sampled 1000 samples of 12 h to form a representative load time series. During the sampling, each feature vector was drawn uniformly from the cluster member, and outliers were not taken into account. Furthermore, the cluster size was taken into account to obtain the same size ratio in the representative time series as in the clustered data. E.g., if two clusters are found with 80 and 20 members

and we would draw 10 samples, 8 of them would be for the first and 2 from the second.

The input load time series and the representative load time series were used in a load flow calculation of a generic Kerber distribution grid [14] as shown in Fig. 4. The generic distribution grid consists of one transformer, five feeders with 57 nodes and 57 lines. For each line, node and the transformer a time series of the capacity of the component is calculated using the load flow calculation. The load flow calculation was performed using the Python based open source software PandaPower [15].

The transformer capacities, line capacities, and node voltages time series are checked for similar statistical features using the Fisher-Pitman permutation test [11]. To decide if both time series are similar the Fisher-Pitman test should result in high p-values ≥ 0.15 and ≤ 0.85 . In our evaluation, we choose to assess the minimum, maximum, mean, median and standard deviation to determine the similarity of the input time series with the representative time series.

4.2 Results

During our evaluation, we calculated p-values for the minimum, maximum, mean, median and standard deviation for transformer capacities, line capacities, and node voltages. Figure 5 shows an exemplary boxplot of the calculations of the SWFE for the line capacities.

Table 1 presents an overview of the results. All algorithms, except the CNN, show promising behavior to create representative time series. The SWFE is capable of maintaining information about the maximum values and the standard

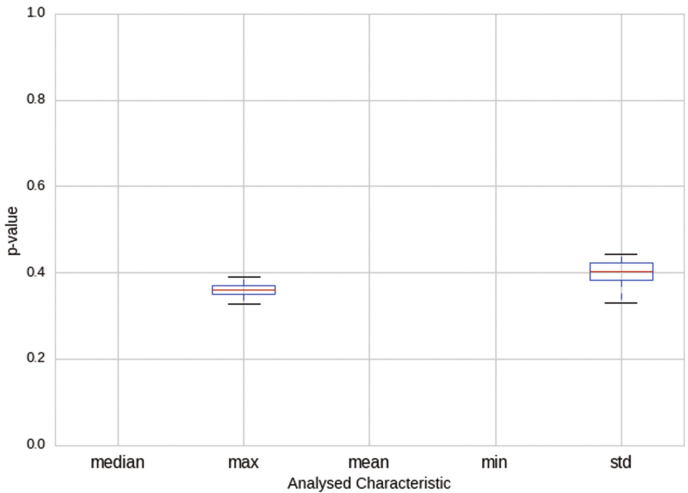


Fig. 5. Exemplary boxplot of the p-values for the line capacity. Calculated using the Fisher Pitman test for the SWFE.

Table 1. Overview of p-values per algorithm. The means of all load flow results are shown with the corresponding standard deviation.

Algorithm	p value for	Transformer	Line	Voltage
SWFE	median	0.000(\pm 0.000)	0.000(\pm 0.000)	1.000(\pm 0.000)
	max	0.293 (\pm 0.250)	0.293 (\pm 0.249)	0.980(\pm 0.140)
	mean	0.000(\pm 0.000)	0.000(\pm 0.000)	1.000(\pm 0.000)
	min	0.020(\pm 0.140)	0.020(\pm 0.140)	0.707 (\pm 0.250)
	std	0.366 (\pm 0.353)	0.365 (\pm 0.353)	0.335 (\pm 0.345)
FFE	median	0.019(\pm 0.131)	0.019(\pm 0.129)	0.981(\pm 0.129)
	max	0.983(\pm 0.060)	0.982(\pm 0.059)	1.000(\pm 0.000)
	mean	0.232 (\pm 0.418)	0.232 (\pm 0.417)	0.767 (\pm 0.418)
	min	0.000(\pm 0.000)	0.000(\pm 0.000)	0.018(\pm 0.059)
	std	1.000(\pm 0.000)	1.000(\pm 0.000)	1.000(\pm 0.000)
CNN	median	0.000(\pm 0.000)	0.000(\pm 0.000)	1.000(\pm 0.000)
	max	1.000(\pm 0.000)	1.000(\pm 0.000)	1.000(\pm 0.000)
	mean	0.000(\pm 0.000)	0.000(\pm 0.000)	1.000(\pm 0.000)
	min	0.000(\pm 0.000)	0.000(\pm 0.000)	0.000(\pm 0.000)
	std	1.000(\pm 0.000)	1.000(\pm 0.000)	1.000(\pm 0.000)

deviation of the load flow results. The FFE is capable of maintaining information about the mean of the load flow results. The high standard deviations of the results are due to the averaging over all calculated load flow results. Due to different losses and line length within the power grid, these values tend to vary. The CNN algorithm is not capable of reproducing any of the characteristics calculated by the load flow calculation.

The runtime could successfully be reduced from 638 s using the 35040 data points of the original time series to 76 s using the 1000 data points of the representative time series. The runtime was averaged over 10 load flow calculations on a single core of an Intel Xeon E5-2683 @ 2.1 Ghz.

5 Conclusion and Outlook

In this paper, we presented a method to obtain representative time series from an input time series using feature vectors of time series segments. We then evaluated the results of a load flow calculation for the representative time series and the input time series. By applying Fisher-Pittmann test on the results of the load flow calculation, we were able to assess the similarity of both time series.

As shown in Sect. 4, only the SWFE, and FFE are capable of recreating load flow results similar to the original input time series. The CNN method fails to recreate the load flow results. The shortcomings of the SWFE is to obtain similar median, mean and minimal values for transformer and line capacities.

The minimum value of the node voltage can be achieved using the SWFE. The FFE is only capable of obtaining similar mean values and has a low score on all other characteristics. The CNN feature extraction is not able to reproduce any load flow results. This might be due to the layout of the CNN which uses 48 input neurons and has a bottleneck of 4 neurons which might not be good enough to reproduce the necessary information of the input time series.

An additional explanation for the low scores in areas such as the mean, min or median values might be the applied sampling strategy or the size of the representative time series. We tried to have a broad general view of the data and sampled uniformly on the clusters. A better strategy would be to adjust the sampling to the feature extraction algorithm as well as the specific applications which have to be solved using the load flow calculations, e.g., grid expansion planning.

Furthermore, we might be able to obtain more information about the original time series by increasing the size of the representative time series. If this is done, it has to be taken into account that by increasing the size of the representative time series the runtime of the load flow calculation will increase as well.

The additional runtime for the SWFE and the FFE to extract the representative time series adds to the total runtime but can be considered less in comparison to the achieved speedup of about 90%. The CNN, on the other hand, needs to learn a representation of the data, which makes the algorithm unusable for single time usage. If applied to real world data, the CNN should be pre-trained. This pre-training allows for a pre-adaption to load time series. During their application, the fine tuning can be done on the data for whom the representative time series need to be calculated.

Our future work includes the adoption of the presented methods to multivariate time series to better capture the correlation between different household loads. This adoption may allow improving the load flow analysis for renewable energies further. All of the methods are capable of performing multivariate feature extraction; this would result in an increased size of the feature vector for clustering. Hence, another dimensionality reduction algorithm might need to be applied before clustering. Furthermore, we want to develop sampling strategies, which allow selecting specific features for grid planning tasks, e.g., grid expansion or stability calculations. Other work will include an analysis of generating representative time series by using a function, which maps actual load flow results back to the time series itself. This allows learning a function which maps similar load flow results to a specific time series.

Acknowledgment. This work was created within the PrIME (03EK3536A) project and funded by BMBF: Deutsches Bundesministerium für Bildung und Forschung/German Federal Ministry of Education and Research.

References

1. Barrows, C., Hummon, M., Jones, W., Hale, E.: Time Domain Partitioning of Electricity Production Cost Simulations. Technical report, National Renewable Energy Laboratory (NREL) (2014)
2. Nason, G.P., Powell, B.: Should we sample a time series more frequently?: decision support via multirate spectrum estimation. *J. R. Statist. Soc. A* **180**(2), 1–30 (2017)
3. Merrick, J.H.: On representation of temporal variability in electricity capacity planning models. *Energy Econ.* **59**, 261–274 (2016)
4. Green, R., Staffell, I., Vasilakos, N.: Divide and conquer? k-means clustering of demand data allows rapid and accurate simulations of the British electricity system. *IEEE Trans. Eng. Manag.* **61**(2), 251–260 (2014)
5. Poncelet, K., Hoschle, H., Delarue, E., Virag, A., D’haeseleer, W.: Selecting representative days for capturing the implications of integrating intermittent renewables in generation expansion planning problems. *IEEE Trans. Power Syst.* **32**(3), 1936–1948 (2017)
6. Tsekouras, G.J., Kotoulas, P.B., Tsirekis, C.D., Dialynas, E.N., Hatzigiargyriou, N.D.: A pattern recognition methodology for evaluation of load profiles and typical days of large electricity customers. *Electr. Power Syst. Res.* **78**(9), 1494–1510 (2008)
7. Räsänen, T., Kolehmainen, M.: Feature-based clustering for electricity use time series data. In: Kolehmainen, M., Toivanen, P., Beliczynski, B. (eds.) ICANNGA 2009. LNCS, vol. 5495, pp. 401–412. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04921-7_41
8. Li, X., Bowers, C.P., Schnier, T.: Classification of energy consumption in buildings with outlier detection. *IEEE Trans. Ind. Electron.* **57**(11), 3639–3644 (2010)
9. Gensler, A., Gruber, T., Sick, B.: Fast feature extraction for time series analysis using least-squares approximations with orthogonal basis functions. In: 22nd International Symposium on Temporal Representation and Reasoning, pp. 29–37. IEEE (2015)
10. Fuchs, E., Gruber, T., Nitschke, J., Sick, B.: Online segmentation of time series based on polynomial least-squares approximations. *IEEE Trans. Patt. Anal. Mach. Intell.* **32**(12), 2232–2245 (2010)
11. Berry, K.J.: The fisher-pitman permutation test: an attractive alternative to the F Test. *Psychol. Rep.* **90**(2), 495 (2002)
12. von Appen, J., Haack, J., Braun, M.: Erzeugung zeitlich hochaufgelöster Stromlastprofile für verschiedene Haushaltstypen. In: IEEE Power and Energy Student Summit, Stuttgart (2014)
13. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
14. Kerber, G.: Aufnahmefähigkeit von Niederspannungsverteilsnetzen für die Einspeisung aus Photovoltaikkleinanlagen. Dissertation, Technische Universität, München (2011)
15. Thurner, L., Scheidler, A., Dollichon, J., Schfer, F., Menke, J.H., Meier, F., Meinecke, S., et al.: pandapower - Convenient power system modelling and analysis based on PYPOWER and pandas. Technical report, University of Kassel and Fraunhofer Institute for Wind Energy and Energy System Technology (2017)

Scalable Gaussian Process Models for Solar Power Forecasting

Astrid Dahl^(✉) and Edwin Bonilla

University of New South Wales, Sydney, Australia
astrid.dahl@student.unsw.edu.au

Abstract. Distributed residential solar power forecasting is motivated by multiple applications including local grid and storage management. Forecasting challenges in this area include data nonstationarity, incomplete site information, and noisy or sparse site history. Gaussian process models provide a flexible, nonparametric approach that allows probabilistic forecasting. We develop fully scalable multi-site forecast models using recent advances in approximate Gaussian process methods to (probabilistically) forecast power at 37 residential sites in Adelaide (South Australia) using only historical power data. Our approach captures diurnal cycles in an integrated model without requiring prior data detrending. Further, multi-site methods show some advantage over single-site methods in variable weather conditions.

1 Introduction

Solar power forecasting is motivated by several areas of application, including grid management, load shifting (demand management) and energy storage management. As small scale residential solar penetration grows, challenges to forecasting power for multiple distributed small scale sites, in particular forecasting with incomplete site information and noisy power data, become of interest.

Challenges in this context include nonstationarity in the data,¹ and developing useful probabilistic forecasts. Many forecasting methods assume it is possible to detrend power data prior to stochastic modelling in order to ‘flatten’ the data and remove diurnal cyclical trends associated with cycles in solar radiation. However, methods to do so rely on comprehensive site information [7], or site history as in [5, 17]. Overall, existing methods have high data demands, constraining their usefulness for new or unseen sites.

For certain applications it is desirable to work with a probabilistic distribution of forecasts that quantifies forecast uncertainty. Statistically-based methods, such as vector autoregressive (VAR) models, typically allow probabilistic forecasts however are constrained in their application to unflattened data and sites for which no training data is available. Machine learning methods such as

¹ Stationarity here refers to the property that distribution parameters remain stable (and finite) over time.

neural networks (ANNs) are more widely applicable but do not generally allow probabilistic forecasts. Gaussian process models are advantageous in this regard, providing a flexible, nonparametric forecast approach that is also probabilistic in nature.

Transfer learning over distributed sites may assist in addressing site data limitations as well as improve prediction of weather-related power fluctuations. The literature suggests cross site data can be helpful in modelling cloud conditions to improve site level forecasts, as in [3, 12, 23], with evidence that cross site information in a dense network can be relevant from timescales of a few minutes, as in [27], to multi-hour horizons in a widely distributed network, as in [3].

A key constraint often associated with multisite approaches is scalability to large numbers of sites. Within the Gaussian process literature, several approximate methods have been developed that support stochastic parameter optimisation, thus maintaining scalability to large datasets and feasibility for real world application.

The current study considers the problem of short term (less than 30 min) power forecasting for large distributed networks of residential rooftop solar systems. We apply sparse variational Gaussian process (GP) approaches for probabilistic forecasting across multiple solar sites in Adelaide, Australia. Our aim is to test whether scalable GP methods can be applied to short term distributed forecasting to provide useful, probabilistic forecasts at the site level with limited site history and information.

1.1 Related Work

The literature around solar forecasting is extensive, including studies that investigate both solar irradiance and power forecasting over multiple forecast horizons (a few minutes to multiple days) using approaches that range from physics-based models to statistical and machine learning methods. Studies to date also examine multiple inputs including irradiance or power measurements, ground and satellite based weather data and meteorological forecasts. Several reviews [10, 14, 18, 25] provide a thorough coverage of recent methods.

In the sphere of short term forecasting, stochastic models utilising only historical power data have been shown to perform relatively well in the past [18], although recent advances suggest highly accurate forecasts can be produced by including comprehensive climate data [16]. Predominant statistical methods include adaptively estimated VAR, autoregressive integrated moving average (ARIMA) and generalised autoregressive conditional heteroskedasticity (GARCH) models, for example as in [3, 9]. These methods allow for weather-related nonstationarity and at shorter horizons (up to one hour) have been found to be competitive in forecasting clearness indices (i.e. flattened irradiance data) [9, 15]. In a number of cases models are applied in a multisite setting as in [3, 5, 11, 27].

The major machine learning methods explored for short term solar forecasting are neural networks and support vector machines (SVMs). Recent examples of ANNs for short term horizons include [12, 20]. ANNs have been explored in

a multisite setting in irradiance forecasting [25], although at time of writing no examples were identified of multivariate prediction at horizons less than one hour ahead.

Gaussian process and related models have been explored to a limited extent in solar forecasting. [9, 15] include univariate GP models applied to clearness indices as comparative models. [4] also uses a GP model to forecast clearness index values over an irradiance field in 30 min increments. The authors apply probabilistic principal components dimension reduction to improve feasibility of real time adaptive GP modelling over multiple locations (by reducing the number of ‘locations’ for which a GP is estimated), and further assume independence between models, thus respecifying the multivariate problem as several univariate problems. However, even with these adaptations, scalability is still constrained by non-stochastic optimisation of the exact GP as described in Sect. 2.

Several studies use a closely related method, kriging, to predict clearness indices in a multisite setting [2, 22, 23, 26]. Building on [22, 26] develops one-hour-ahead clearness index forecasts using one month of hourly data from a group of 10 meteorological stations in Singapore. In [23], the authors ‘nowcast’ clearness index values for 25 sensor locations covering an approximately 30 km radius area in Osaka.

2 Theory

Gaussian process (GP) models provide a flexible nonparametric Bayesian approach to machine learning problems such as regression and classification [21] and have proved successful in various application areas involving spatio-temporal modeling [8]. Formally, a GP is a prior over functions for which every subset of function values $f(\mathbf{x}_1), \dots, f(\mathbf{x}_n)$ follows a Gaussian distribution. We denote a function drawn from a GP with mean function $\mu(\mathbf{x})$ and covariance function $\kappa(\cdot, \cdot)$ by $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), \kappa(\mathbf{x}, \mathbf{x}'))$.

One of the most widely used GP models is the standard regression setting with a zero-mean GP and i.i.d. Gaussian noise:

$$y_t \sim \mathcal{N}(f(\mathbf{x}_t), \sigma_y^2) \text{ with } f(\mathbf{x}_t) \sim \mathcal{GP}(\mathbf{0}, \kappa(\mathbf{x}_t, \mathbf{x}_{t'})), \quad (1)$$

where \mathbf{x}_t denote features at time t and σ_y^2 is the noise variance.

Given a set of observations $\{(\mathbf{x}_t, y_t)\}_{t=1}^N$, we wish to learn a model in order to make predictions at a new datapoint \mathbf{x}_* . Given the likelihood and prior models in Eq. (1), the predictive distribution over $f(\mathbf{x}_*)$ is a Gaussian with mean and variance given by:

$$\mu_* = \kappa(\mathbf{x}_*, \mathbf{X})(\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \mathbf{y}, \quad \sigma_* = \kappa(\mathbf{x}_*, \mathbf{x}_*) - \kappa(\mathbf{x}_*, \mathbf{X})(\mathbf{K} + \sigma_y^2 \mathbf{I})^{-1} \kappa(\mathbf{X}, \mathbf{x}_*),$$

where \mathbf{X} and \mathbf{y} denote all the training features and outputs, respectively; \mathbf{K} is the covariance matrix induced by evaluating the covariance function at all training datapoints; and \mathbf{I} is the identity matrix.

Although computing the exact predictive distribution above is appealing from the theoretical perspective and in a small-data regime, these computations become unfeasible for large datasets as their time and space complexity are $\mathcal{O}(N^3)$ and N^2 respectively.

Much of the research efforts in GP models have been devoted to this issue [19] with significant breakthroughs achieved over the last few years [13, 24]. Indeed, here we study the *variational* approach to inference in GP models, which relies upon reformulating the prior via the so-called *inducing* variables [24].

2.1 Scalable Gaussian Process Regression via Variational Inference

Full details of the variational approach to scalable GP regression is out of the scope of this paper and we refer the reader to [6, 13, 24] for further reference. Here it suffices to explain that we introduce a set of M inducing variables $\mathbf{u} = (u_1, \dots, u_M)$, which lie in the same space as the original function values and are drawn from the same GP prior. For these inducing variables we have their corresponding inputs $\mathbf{Z} = (\mathbf{z}_1, \dots, \mathbf{z}_M)$, where each \mathbf{z}_j is a D -dimensional vector in the same space as the original features \mathbf{x} .

The variational approach to GP inference involves a reformulation of the prior via the inducing variables and the proposal of an approximate posterior over these using $q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{S})$, which is estimated via the optimization of the so-called evidence lower bound (ELBO):

$$\mathcal{L}_{\text{elbo}}(\mathbf{m}, \mathbf{S}) = \text{KL}(q(\mathbf{u})||p(\mathbf{u})) - \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})], \quad (2)$$

where $\text{KL}(q||p)$ denotes the Kullback-Leibler divergence between distributions q and p ; $p(\mathbf{u}) = \mathcal{N}(\mathbf{0}, \kappa(\mathbf{Z}, \mathbf{Z}))$ is the Gaussian prior over the inducing variables; $\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})]$ is the expectation of the conditional likelihood (given in Eq. (1)) over $q(\mathbf{f}) = \int_{\mathbf{u}} q(\mathbf{u})q(\mathbf{f}|\mathbf{u})d\mathbf{u}$; and $q(\mathbf{u})$ the approximate posterior given above. Using simple properties of the Gaussian distribution it is possible to show that Eq. (2) can be solved analytically and, more importantly, $\mathcal{L}_{\text{elbo}}$ decomposes as a sum of objectives over the training data. This readily allows the application of stochastic optimization methods rendering the time and space complexity of the algorithm as $\mathcal{O}(M^3)$ and $\mathcal{O}(M^2)$, respectively, hence independent of N and applicable to very large datasets.

2.2 Gaussian Processes for Solar Power Forecasting

A key advantage of GP models is their flexibility to express potentially nonlinear relationships and nonstationary processes through various kernel forms. GP models have the capacity to account for nonstationarity associated with diurnal cycles through appropriate kernel functions. Further, their nonparametric nature allows models to flexibly reflect variable volatility i.e. nonstationarity associated with weather effects.

In the present study, we propose several Gaussian process model specifications for application to the residential solar forecasting problem where site

information is unknown. Kernels are structured to capture both cyclical and autoregressive processes in the power data. We compare results under both ‘site-independent’ approaches, where Gaussian process models are applied to sites individually, and multi-site approaches, where forecasting for multiple sites is performed via collaborative Gaussian process models.

Site-Independent Models. Consider the timeseries of power observations for a single site p , denoted y_p , at times $t = 0, \dots, N$. As in (1), let

$$y_{pt} = f_p(\mathbf{x}_{pt}) + \epsilon_{pt}, \quad f_p(\mathbf{x}) \sim \mathcal{GP}(0, \kappa_p(\mathbf{x}_{pt}, \mathbf{x}_{ps})) \quad (3)$$

$\epsilon_{pt} \sim iid\mathcal{N}(0, \sigma_{y_p}^2)$. Under the GP specification, observed power y_{pt} is a function of a latent Gaussian process, $f_p(\mathbf{x}_{pt})$, plus idiosyncratic noise ϵ_{pt} . The covariance between power at time t and time s , $s \neq t$ is thus given by the kernel function $\kappa_p(\mathbf{x}_{pt}, \mathbf{x}_{ps})$. The likelihood function is given by $y_{pt}|f_{pt} \sim \mathcal{N}(f_{pt}, \sigma_{y_p}^2)$.

In the site-independent setting, the feature vector \mathbf{x}_{pt} is comprised of two main elements: a time index t and a set of lagged power observations at pre-specified five minute intervals denoted \mathbf{g} . In order to forecast power at $t + \delta$ for δ steps ahead, lag features are current observed power and past observed power at 5 and 10 min prior. Thus $\mathbf{x}_{pt} = (t, \mathbf{g}_{pt})$, $\mathbf{g}_{pt} = (y_{pt-\delta}, y_{pt-\delta-1}, y_{pt-\delta-2})$. Lags were selected in line with previous studies that find immediate lags are relevant for short term forecasting (see e.g. [26]).

Additional, ‘extended site-independent’ models are estimated using an augmented set of lag features. The feature vector is extended to include power observations of nearby sites, that is $\mathbf{g}_{pt} = (y_{pt-\delta}, y_{pt-\delta-1}, y_{pt-\delta-2}, y_{-pt-\delta}, y_{-pt-\delta-1}, y_{-pt-\delta-2})$, where y_{-p} denotes all sites near to site p . Utilising cross-site features in the form of lags allows separate site model estimation and has been applied in several studies including [3, 12]. We define ‘near’ as being within a 10 km radius.²

Kernel functions for site-independent and extended site-independent models are comprised of several separable kernel elements. A periodic kernel is applied to the time index to capture daily cyclical trends in output and is defined as

$$\kappa_{Per.}(t, s) = \theta \exp \left[-0.5 \left(\frac{\sin \left(\frac{\pi}{T} (t - s) \right)}{l} \right)^2 \right] \quad (4)$$

where θ governs cycle amplitude, T denotes cycle period (fixed at one day), and lengthscale, l , governs rate of decay in covariance as the time-span between observations increases.

A linear kernel is applied to lag features $g_i \in g$ to capture short term variations from the regular diurnal trend:

$$\kappa_{Lin.}(\mathbf{g}_{pt}, \mathbf{g}_{ps}) = \sum_i \sigma_i g_{pti}, g_{psi} \quad (5)$$

² A fixed radius is applied to provide local regularisation, which has been found to reduce overfitting in multisite settings [11, 27]. The 10 km threshold aims to limit ‘neighbours’ to sites most likely to be relevant given historic local windspeed.

where σ_i are in effect weight coefficients. The overall kernel structure for all site-independent models is:

$$\kappa_p(\mathbf{x}_{pt}, \mathbf{x}_{ps}) = \kappa_{Per.}(t, s) \kappa_{Lin.}(\mathbf{g}_{pt}, \mathbf{g}_{ps}). \quad (6)$$

Multi-site Models. Values for proximate sites would be expected to covary, due to both synchronous diurnal cycles in unflattened data and shared weather systems. Some efficiency would thus be expected from exploiting the shared covariance structure through collaborative learning.

Two separate multi-site GP model structures are estimated for site-level power forecasting. The first is a pooled structure, where (standardised) site data are used in a joint specification with shared kernel parameter values. The second structure is the linear coregionalisation model or LCM. This structure assumes site observations covary through a lower dimension set of shared latent processes.

For each multi-site model structure, two alternative kernel specifications are explored. These four model specifications are detailed below.

Pooled Model. The pooled, or ‘joint’, model is a pooled Gaussian process model where all site observations share a common kernel that includes an additional kernel element defining a spatial covariance factor.

The first pooled model kernel (‘Joint Model 1’) is defined as a multiplicative, separable spatiotemporal kernel added to a shared linear kernel applied to lagged power values. Feature vector \mathbf{x} is extended to include $\mathbf{h} = (\textit{latitude}, \textit{longitude})$ i.e. $\mathbf{x}_{pt} = (t, \mathbf{g}_{pt}, \mathbf{h}_p)$. A radial basis function (RBF) kernel is applied to $h_i \in \mathbf{h}$ to capture spatial dependencies, thus for sites p and q ,

$$\kappa(\mathbf{x}_{pt}, \mathbf{x}_{qs}) = \kappa_{Per.}(t, s) \kappa_{RBF}(\mathbf{h}_p, \mathbf{h}_q) + \kappa_{Lin.}(\mathbf{g}_{pt}, \mathbf{g}_{qs}). \quad (7)$$

where

$$\kappa_{RBF}(\mathbf{h}_p, \mathbf{h}_q) = \sigma^2 \exp\left\{-\frac{1}{2} \sum_{i=1}^2 ((h_{pi} - h_{qi})/l_i)^2\right\}. \quad (8)$$

In the RBF kernel, σ^2 governs maximum covariance between points h_p and h_q , and lengthscale, l_i , governs the rate of decay in covariance as distance between observations along the relevant axis increases.

The second joint model (‘Joint Model 2’) is similarly specified however replaces the shared linear kernel with separately parameterised linear kernels for each site. Specifically, $\kappa_{Lin.}(\mathbf{g}_{pt}, \mathbf{g}_{qs})$ becomes

$$\kappa_{Lin.,p}(\mathbf{g}_{pt}, \mathbf{g}_{qs}) = \sum_i \sigma_{pi} g_{pti}, g_{qsi}, \quad \kappa_{Lin.,p} = 0 \quad \text{for } p \neq q \quad (9)$$

Coregional Model. The linear coregional model (LCM) assumes y_p is a function not of a single latent process $f_p(\mathbf{x}_p)$ but a linear combination of several independent latent Gaussian processes. Covariance between sites arises from these shared latent processes. Weights defining the linear combination for a given site are site-specific,³ $f_p(\mathbf{x}) = \sum_{j=1}^Q w_{pj} u_j(\mathbf{x})$.

We assume three latent processes $u(x)_j, j = 1, \dots, 3$ in the first LCM model ('LCM Model 1') and two latent processes in the second model ('LCM Model 2'). Each latent process has an associated kernel, κ_j , giving rise to a shared covariance structure across sites driven by both kernel elements and weight matrices.

Let $\mathbf{B}_j = \mathbf{W}_j \mathbf{W}'_j + \kappa_j$ where \mathbf{W}_j is a $p \times 1$ matrix of weights w_{pj} , and κ_j is a diagonal matrix of isotropic noise. We define $\kappa_1 = \kappa_{Per.}(t, s)$ and $\kappa_2 = \kappa_{RBF}(t, s)$ respectively as periodic and RBF kernels applied to time indices. The third latent process kernel is defined as $\kappa_3 = \kappa_{Lin.}(\mathbf{g}_t, \mathbf{g}_s)$. The shared kernel structure in LCM Model 1 is thus given by:

$$\mathbf{K}(f_p(\mathbf{x}_{pt}), f_q(\mathbf{x}_{qs})) = \sum_{j=1}^3 [\mathbf{B}_j]_{pq} \kappa_j(\mathbf{x}_{pt}, \mathbf{x}_{qs}). \quad (10)$$

The second coregional model is similar to the above, however again the linear kernel component is treated slightly differently. In LCM Model 2, $Q = 2$ with κ_1 and κ_2 defined as above, and lag features are included in a separate kernel component defined as in (9). Thus

$$\mathbf{K}(f_p(\mathbf{x}_{pt}), f_q(\mathbf{x}_{qs})) = \sum_{j=1}^2 [\mathbf{B}_j]_{pq} \kappa_j(\mathbf{x}_{pt}, \mathbf{x}_{qs}) + \kappa_{Lin.,p}(\mathbf{g}_{pt}, \mathbf{g}_{qs}). \quad (11)$$

The specification in (11) allows a slightly more expressive parameterisation of the linear kernel than (10).

Benchmark Models. Without clear sky normalisation and under the assumption of short site history, there are few existing models for comparison. One feasible benchmark prevalent in the literature is the persistence model,⁴ which forecasts the next observation as the current observation i.e. $y_{t+1} = y_t$. Persistence models are estimated for each site separately.

In addition, the site-independent Gaussian process models serve as a benchmark. These are approximately equivalent to linear Bayesian regression models assuming a standard Gaussian prior distribution over regression coefficients. As such they are closely related to VAR models as applied to flattened data.

3 Experiments

The analysis makes use of a sample of 37 residential photovoltaic systems installed within an approximately 10 by 15 km 'box' in the central Adelaide area.

³ A useful exposition of coregional models can be found at [1].

⁴ The persistence model in the present study is applied to unflattened data.

Most sites have an installed capacity of 2 to 5 kW. The dataset is comprised of 5-minute average power readings over a 30 day period in January 2017 (specifically 30 days ending 28 January 2017). Days were defined as 7 am to 7 pm, yielding a total of 144 observations for a site over a day. This accounts for a total of 159,840 observations, which is clearly unfeasible for standard (non-scalable) GP models.

The goal of the experiments is to test whether GP models estimated under a sparse variational framework can be applied to forecast distributed power output at the site level for multiple distributed sites. In particular, whether (a) combined kernel forms can be used to model nonstationary data characteristics, and (b) collaborative learning can improve forecast accuracy or reduce data requirements compared to independent site forecasts.

The four multisite models set out above are used to forecast output for each site. These are compared to results under the site-independent and persistence models. Models are trained for forecasting horizons from five to thirty minutes at five minute intervals. The forecast target in each case is five minute average power at that horizon. Models were trained using the first 60% of observations (18 days). Forecasts were then generated for a test set of the following 40% of observations (12 days) for each site.

All models are estimated via the sparse, variational approach described in Sect. 2. Inducing points are initialised at cluster centroids and optimised within the model. To illustrate the scalability of the approach, we use 2300 inducing points for joint models, or approximately 2.4% of the data dimension. Maintaining the same ratio, 63 inducing points per site were used for individual models.

3.1 Accuracy Metrics

Forecast accuracy is assessed for each site for each model using three measures: mean absolute error (MAE) in kilowatts, standardised mean squared error (SMSE) and standardised mean log loss (SMLL), as defined in [21]. Specifically,

$$SMSE = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} \left(\frac{y - \hat{y}_i}{\sigma_{y_{te}}} \right)^2 \quad (12)$$

$$MAE = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} |y - \hat{y}_i| \quad (13)$$

$$SMLL = \frac{1}{N_{te}} \sum_{i=1}^{N_{te}} (nlpd_i - nll_i), \quad \text{where} \quad (14)$$

$$nlpd_i = \frac{1}{2} \left[\ln(2\pi) + \ln \sigma_{\hat{y}_i}^2 + \left(\frac{y_i - \hat{y}_i}{\sigma_{\hat{y}_i}} \right)^2 \right],$$

$$nll_i = \frac{1}{2} \left[\ln(2\pi) + \ln \sigma_{y_{tr}}^2 + \left(\frac{y_i - \mu_{y_{tr}}}{\sigma_{y_{tr}}} \right)^2 \right].$$

Subscripts te and tr refer to training and test sets respectively and \hat{y} denotes the predicted value of y . SMSE is standardised by reference to test set variance $\sigma_{y_{te}}^2$. Values less than one indicate the model improves on a simple mean forecast. SMLL measures the (negative log) likelihood of the test data under the model, denoted $nlpd$, relative to (negative log) likelihood under the trivial normal distribution with parameters $(\bar{y}_{tr}, \sigma_{y_{tr}}^2)$, denoted nll . More negative metric values indicate better relative performance of the model.⁵

3.2 Results

Forecast Accuracy. Results at the site level suggest the site-independent model performs as well as or better than the joint (pooled) model in terms of average site accuracy (Fig. 1). SMSE for both the site independent and joint models ranges from 0.05–0.12 over 5–30 forecast horizons, however MAE and SMLL are consistently improved under the site-independent model over all forecast horizons e.g. MAE of 0.14–0.26 kW versus 0.17–0.29 kW under site and joint models respectively. The LCM specifications perform poorly on all measures relative to the joint and basic site-independent models. At all forecast horizons, the better performing models (joint and site-independent) are more accurate than the persistence benchmark.

Additional expressiveness in the kernel due to the more flexible linear lag kernel structure does not significantly improve forecast accuracy in the joint or LCM models, and in some cases tends to contribute to higher forecast variability across sites (Fig. 1). Interestingly, the extended site-independent model performs very poorly relative to other models, however forecast error remains fairly stable over 10–30 min horizons. This result may indicate overspecification of this (very flexible) kernel structure.

Estimation of Daily Power Curve. It is difficult to evaluate the current approach as an alternative to those that require flattening the data without a direct (flattened) benchmark for the given dataset. However, examining forecast accuracy on clear⁶ days provides some insight into how the approach accounts for clear sky curves. Table 1 summarises forecast accuracy under the site-independent and joint model 1 specifications for clear (or mostly clear) and cloudy days in the test set, which each represent 50% of the test data.

Forecast accuracy appears competitive on clear days, with mean MAE across sites of 50 Watts on clear days at the five minute horizon, rising to 130 Watts at the 30 min horizon. Given mean power for the full dataset set across clear and cloudy days of 2.1 kW, MAE represents around 2.4 (6.2)% of mean power at

⁵ Note that SMLL does not apply to the non-probabilistic persistence model.

⁶ Clear days are defined as those where daily global horizontal irradiance (GHI) was more than 90% of mean maximum daily GHI for the month of January. Measurements are from the Adelaide (West Terrace) Australian Bureau of Meteorology weather station. GHI for clear (cloudy) days ranges from 93–97 (36–90)% of the mean January maximum.

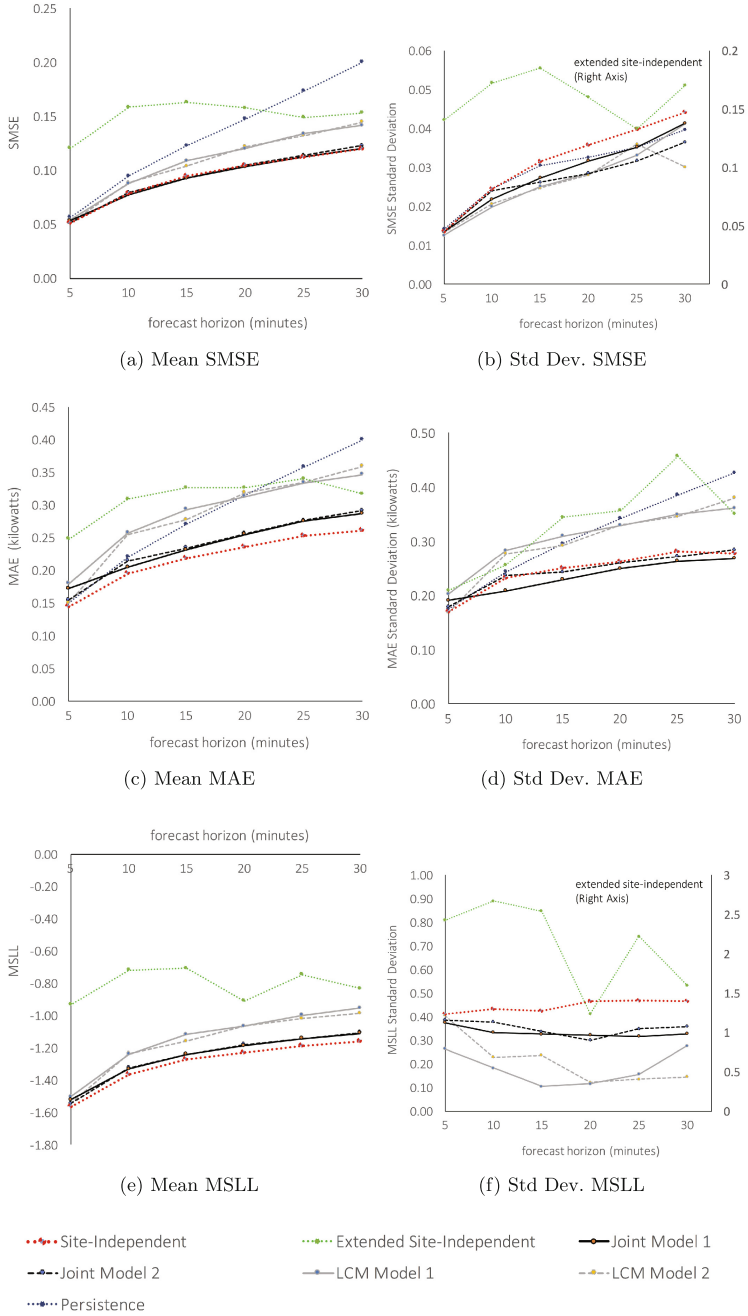


Fig. 1. Site forecast mean error and error variability

Table 1. Mean site forecast accuracy on clear versus cloudy days

Model	Horizon	Clear days			Cloudy days		
		MAE (kW)	SMSE	MSLL	MAE (kW)	SMSE	MSLL
joint model 1	5	0.091	0.003	-2.01	0.253	0.051	-1.03
site-independent	5	0.050	0.002	-2.10	0.238	0.052	-1.04
joint model 1	10	0.100	0.005	-1.77	0.308	0.066	-0.89
site-independent	10	0.078	0.005	-1.88	0.311	0.077	-0.85
joint model 1	15	0.122	0.007	-1.65	0.340	0.076	-0.83
site-independent	15	0.096	0.006	-1.77	0.341	0.085	-0.78
joint model 1	20	0.150	0.009	-1.56	0.359	0.080	-0.81
site-independent	20	0.109	0.007	-1.69	0.361	0.093	-0.77
joint model 1	25	0.171	0.011	-1.49	0.378	0.085	-0.79
site-independent	25	0.124	0.008	-1.63	0.380	0.100	-0.75
joint model 1	30	0.184	0.012	-1.44	0.389	0.087	-0.78
site-independent	30	0.133	0.009	-1.59	0.390	0.103	-0.73

the 5 (30) min horizon. Similarly, SMSE of 0.002 to 0.009 (for site-independent results) over 5 to 30 min horizons implies that average mean squared error is less than one percent of total power variation on clear days.

Considering transfer learning more generally, it is relevant to note the better performance of the joint model on cloudy days, which contrasts with the better performance of the site-independent models on clear days (Table 1). On all measures, the best performing joint model performs consistently better during variable weather, while the opposite is true for sunny weather periods (noting accuracy is significantly diminished for both models on cloudy days). This suggests ‘negative’ transfer effects with respect to forecasting diurnal cycles, while forecast errors are somewhat moderated during cloudy periods by the joint model.

4 Discussion

The scalable, approximate Gaussian process methods appear to have significant potential in the distributed forecasting setting. We are able to produce probabilistic site level forecasts using a flexible, nonparametric method in a large scale setting. Further, the approach seems to incorporate diurnal cycles within an integrated model successfully without exogenous site information.

Gaussian process based models produce a strong level of accuracy on sunny days for forecasts out to the 30 min horizon. Overall, however, accuracy of models during cloudy conditions appears low. Given the absence of feature data beyond location, time and output, however, it is possible accuracy can be substantially improved (as in [16]) via inclusion of weather or other external data, including site features where available.

Overall accuracy of forecasts is not improved by jointly estimated models (pooled and coregional) compared to site-independent models. Performance in cloudy versus clear weather, however, illustrates that there may be potential for transfer learning benefits during more variable weather.

One possible factor affecting model performance is the spatial covariance kernel, which is a stationary function resulting in sites equally distant along a fixed axis being assigned an equal covariance regardless of current weather direction. Ideally, a spatial kernel would more specifically reflect current cloud velocity. Further, the stationary kernel assigns higher weight to closer sites, which may not be optimal as forecast horizons increase. A more refined kernel or adaptive model structure may thus assist in identifying relevant cloud-related data features for transfer learning in a forecast setting.

Acknowledgements. This work was supported by Solar Analytics Pty Ltd. and performed on behalf of the Cooperative Research Centre for Low-Carbon Living (University of New South Wales and Solar Analytics Pty Ltd.).

References

1. Alvarez, M.A., Rosasco, L., Lawrence, N.D., et al.: Kernels for vector-valued functions: a review. *Found. Trends® Mach. Learn.* **4**(3), 195–266 (2012)
2. Aryaputera, A.W., Yang, D., Zhao, L., Walsh, W.M.: Very short-term irradiance forecasting at unobserved locations using spatio-temporal kriging. *Sol. Energy* **122**, 1266–1278 (2015). <http://www.sciencedirect.com/science/article/pii/S0038092X15005745>
3. Bessa, R., Trindade, A., Silva, C.S., Miranda, V.: Probabilistic solar power forecasting in smart grids using distributed information. *Int. J. Electr. Power Energy Syst.* **72**, 16–23 (2015). <http://www.sciencedirect.com/science/article/pii/S0142061515000897>, the Special Issue for 18th Power Systems Computation Conference
4. Billionis, I., Constantinescu, E.M., Anitescu, M.: Data-driven model for solar irradiation based on satellite observations. *Sol. Energy* **110**, 22–38 (2014). <http://www.sciencedirect.com/science/article/pii/S0038092X14004393>
5. Boland, J.: Spatial-temporal forecasting of solar radiation. *Renew. Energy* **75**, 607–616 (2015). <http://www.sciencedirect.com/science/article/pii/S0960148114006624>
6. Bonilla, E.V., Krauth, K., Dezfouli, A.: Generic inference in latent Gaussian process models (2016). arXiv preprint: [arXiv:1609.00577](https://arxiv.org/abs/1609.00577)
7. Copper, J., Sproul, A., Jarnason, S.: Photovoltaic (pv) performance modelling in the absence of onsite measured plane of array irradiance (poa) and module temperature. *Renew. Energy* **86**, 760–769 (2016)
8. Cressie, N., Wikle, C.K.: *Statistics for Spatio-Temporal Data*. John Wiley & Sons, Hoboken (2011)
9. David, M., Ramahatana, F., Trombe, P., Lauret, P.: Probabilistic forecasting of the solar irradiance with recursive ARMA and GARCH models. *Sol. Energy* **133**, 55–72 (2016). <http://www.sciencedirect.com/science/article/pii/S0038092X16300172>
10. Diagne, M., David, M., Lauret, P., Boland, J., Schmutz, N.: Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renew. Sustain. Energy Rev.* **27**, 65–76 (2013). <http://www.sciencedirect.com/science/article/pii/S1364032113004334>

11. Domke, J., Engerer, N., Menon, A., Webers, C.: Distributed solar prediction with wind velocity (2016)
12. Gutierrez-Corea, F.V., Manso-Callejo, M.A., Moreno-Regidor, M.P., Manrique-Sancho, M.T.: Forecasting short-term solar irradiance based on artificial neural networks and data from neighboring meteorological stations. *Sol. Energy* **134**, 119–131 (2016). <http://www.sciencedirect.com/science/article/pii/S0038092X16300536>
13. Hensman, J., Fusi, N., Lawrence, N.D.: Gaussian processes for big data. In: *Uncertainty in Artificial Intelligence* (2013)
14. Inman, R.H., Pedro, H.T., Coimbra, C.F.: Solar forecasting methods for renewable energy integration. *Prog. Energy Combust. Sci.* **39**(6), 535–576 (2013)
15. Lauret, P., Voyant, C., Soubdhan, T., David, M., Poggi, P.: A benchmarking of machine learning techniques for solar radiation forecasting in an insular context. *Sol. Energy* **112**, 446–457 (2015)
16. Li, J., Ward, J.K., Tong, J., Collins, L., Platt, G.: Machine learning for solar irradiance forecasting of photovoltaic system. *Renew. Energy* **90**, 542–553 (2016). <http://www.sciencedirect.com/science/article/pii/S0960148115305747>
17. Lonij, V.P., Brooks, A.E., Cronin, A.D., Leuthold, M., Koch, K.: Intra-hour forecasts of solar power production using measurements from a network of irradiance sensors. *Sol. Energy* **97**, 58–66 (2013)
18. Pelland, S., Remund, J., Kleissl, J., Oozeki, T., De Brabandere, K.: Photovoltaic and solar forecasting: state of the art. *iea pvps task 14, subtask 3.1. report iea-pvps t14-01: 2013. Technical report* (2013). ISBN: 978-3-906042-13-8
19. Quiñonero-Candela, J., Rasmussen, C.E.: A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **6**, 1939–1959 (2005)
20. Rana, M., Koprinska, I., Agelidis, V.G.: Univariate and multivariate methods for very short-term solar photovoltaic power forecasting. *Energy Convers. Manag.* **121**, 380–390 (2016)
21. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. The MIT Press, Cambridge (2006)
22. Sampson, P.D., Guttorp, P.: Nonparametric estimation of nonstationary spatial covariance structure. *J. Am. Stat. Assoc.* **87**(417), 108–119 (1992)
23. Shinozaki, K., Yamakawa, N., Sasaki, T., Inoue, T.: Areal solar irradiance estimated by sparsely distributed observations of solar radiation. *IEEE Trans. Power Syst.* **31**(1), 35–42 (2016)
24. Titsias, M.: Variational learning of inducing variables in sparse Gaussian processes. In: *Artificial Intelligence and Statistics* (2009)
25. Voyant, C., Notton, G., Kalogirou, S., Nivet, M.L., Paoli, C., Motte, F., Fouilloy, A.: Machine learning methods for solar radiation forecasting: a review. *Renew. Energy* **105**, 569–582 (2017). <http://www.sciencedirect.com/science/article/pii/S0960148116311648>
26. Yang, D., Gu, C., Dong, Z., Jirutitijaroen, P., Chen, N., Walsh, W.M.: Solar irradiance forecasting using spatial-temporal covariance structures and time-forward kriging. *Renew. Energy* **60**, 235–245 (2013). <http://www.sciencedirect.com/science/article/pii/S0960148113002759>
27. Yang, D., Ye, Z., Lim, L.H.I., Dong, Z.: Very short term irradiance forecasting using the lasso. *Sol. Energy* **114**, 314–326 (2015). <http://www.sciencedirect.com/science/article/pii/S0038092X15000304>

Wind Speed Forecasting Using Statistical and Machine Learning Methods: A Case Study in the UAE

Khawla Al Dhaheeri, Wei Lee Woon, and Zeyar Aung^(✉)

Department of Computer Science, Khalifa University of Science and Technology,
Masdar Institute, Abu Dhabi, UAE

{kaldhaheeri, wwoon, zaung}@masdar.ac.ae

Abstract. Wind energy is a source of sustainable energy which is developing very quickly all over the world. Forecasting wind speed is a global concern and a critical issue for wind power conversion systems as it has a great influence in the scheduling of power systems as well as on the dynamic control of wind turbines. In this research, we deploy and study four forecasting models in order to forecast wind speeds in the city of Abu Dhabi, United Arab Emirates (UAE). Two of these models are conventional statistical methods, namely, (i) Auto Regression Integrated Moving Average (ARIMA) and (ii) Seasonal Auto Regression Integrated Moving Average (SARIMA) models, and the other two are drawn from the field of machine learning, namely, (i) Artificial Neural Networks (ANN) and (ii) Singular Spectrum Analysis (SSA) models. We compare the performances of these four models in order to determine the model which is most effective for forecasting wind speed data. The results show that the forecasting model SSA provides, on average, the most accurate forecasted values compared to the other three models. However, those three models, ARIMA, SARIMA and ANN, offer better results for the first few hours (around 24 h), which indicates that ARIMA, SARIMA, and ANN models are suitable for short-term forecasting, while SSA is suitable for long-term forecasting. The findings of our research could contribute in defining the fitting forecasting model in terms of short-term forecasting or long-term forecasting.

Keywords: Wind speed · Forecasting · Statistical methods · Machine learning

1 Introduction

Wind energy is a sustainable energy source which is developing very quickly all over the world. The penetration of wind energy is increasingly rapidly; therefore, it is important for power systems to plan for the integration of wind power with the other power sources. Forecasting wind speed is a critical issue for wind power conversion systems as it affects the scheduling of connected power systems as well as the dynamic control of wind turbines. Hence, the ability to accurately predict

wind speed is an important capability [1]. However, making accurate wind speed predictions is a difficult and complex challenge because of the chaotic nature of wind speed fluctuations.

There is a large number of research papers which study wind speed prediction and forecasting, and many different methods or models have been applied. These methods are divided into four categories which are: physical methods, statistical methods, spatial correlation methods, and machine learning methods. Each method has a number of advantages and disadvantages.

As implied by the name, physical models use a large number of physical parameters to build a multi-variate forecasting model. These parameters include geographical orientation, geomorphic conditions, temperature, and pressure. Thus, physical models are good for long-term forecasting and is widely applied for weather predictions.

Statistical models employ time series models based on mathematical equations to generate wind speed predictions based on historical data. The most commonly used models are statistical in nature because of the simplicity of their computations and the inherent ability to handle noise. Examples of statistical models are Auto Regression Moving Average (ARMA), Auto Regression Integrated Moving Average (ARIMA), and Seasonal Auto Regression Integrated Moving Average (SARIMA).

The third category, which is spatial correlation methods, uses multi-dimensional datasets from different measurement stations in order to forecast future wind speed.

Machine learning methods have been used to forecast and predict wind speed time series data. Some of these methods are Fuzzy Logic Methods, Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Singular Spectrum Analysis (SSA). SSA models are relatively new to the field of forecasting and decomposition of time series data. There are some existing studies where SSA has been used for the time series analysis, but this was not on wind speed time series data, as will be attempted in this research work.

In this research, our objective is to forecast hourly based wind speed time series dataset for Abu Dhabi city, UAE. The data were collected at Abu Dhabi International Airport. We study four different forecasting models from two different categories. For the statistical models, (i) ARIMA and (ii) SARIMA are explored, and for the machine learning models, (i) ANN and (ii) SSA are studied.

The performance and the accuracy of each model in forecasting the wind speed data for a week time ahead in hourly basis is then compared. The aim of this study, that compares three traditional forecasting methods for wind speed time series data, which are ARIMA, SARIMA and ANN, with a relatively new method (SSA) that have been used for some time series analysis and decomposition, is to proof that the SSA model provides the best results in forecasting accuracy of the wind speed dataset over the other three forecasting models. To our best knowledge, our study is the first to use SSA for the purpose of wind speed forecasting.

This paper is a summary version of the master's thesis of the first author [2].

2 Methods Explored

In total, we explored four methods: two statistical and two machine learning methods. The two statistical methods are: (i) Auto Regression Integrated Moving Average (ARIMA) and (ii) Seasonal Auto Regression Integrated Moving Average (SARIMA). The two machine learning methods are: (i) Artificial Neural Networks (ANN) and (ii) Singular Spectrum Analysis (SSA).

2.1 Statistical Methods

The main objective of statistical time series analysis is to develop mathematical models which provide reasonable descriptions of a sample data. “In order to provide a statistical setting for describing the character of data that seemingly fluctuate in a random fashion over time, we assume that a time series can be defined as a collection of random variables indexed according to the order in which they are obtained in time” [3]. Forecasting is one of the major applications of statistical time series analysis in which the future possible trend(s) is/are forecasted based on the analysis the various components of the historical time series data. Forecasting using statistical methods are very useful for short and medium term forecasts, since the used historical data usually show inertia and do not have huge changes in a short period [4]. Followings are the brief descriptions of the ARIMA and SARIMA methods that we explore in this research.

ARIMA: The Autoregressive Integrated Moving Average (ARIMA) model is one of the most useful class of homogeneous non-stationary time series models. It includes some useful differencing and variance stabilizing transformation processes in order to connect the stationary and the non-stationary time series models [5]. A proper degree of differencing can be taken in order to transform a homogeneous non-stationary time series to a stationary time series. The autoregressive moving average models (ARMA) are useful for describing stationary time series. Nevertheless, the autoregressive integrated moving average models (ARIMA) are useful for describing various homogeneous non-stationary time series that can be used to build a large class of time series model by the use of differencing.

In the general $ARIMA(p, d, q)$ model, p represents auto regression parameter, d represents the differencing parameter, and q represents moving average parameter.

$$\phi_p(B)(1 - B)^d Y_t = \theta_0 + \theta_q(B)a_t \quad (1)$$

$$\phi_p(B)(1 - B)^d Y_t = \theta_q(B)a_t \quad (2)$$

The composite variable $\phi_p(B)$ represents the stationary Auto Regression operator as $\phi_p(B) = (1 - \phi_1 B - \dots - \phi_p B^p)$, and the composite variable $\theta_q(B)$ represents the invertible Moving Average operator as $\theta_q(B) = (1 - \theta_1 B - \dots - \theta_q B^q)$. These two variables are not sharing any common factors. The variable $(1 - B)^d Y_t$ refers to the differencing operator where d is the difference value. The parameter

θ_0 plays very different roles for $d \geq 0$. The series a_t is a Gaussian $N(0, \alpha_a^2)$ white noise process. The deterministic trend parameter θ_0 is removed for simplicity but no loss of generality. Equation 2 is one of the most frequently used models in forecasting applications.

ARIMA and its variants hybridized with other methods have been used in wind power forecasting in [6–8].

SARIMA: The ARIMA model described in the previous section was an ordinary non-seasonal ARIMA model. In this section, we will describe the seasonal ARIMA model which is denoted as SARIMA. SARIMA models are capable of modeling a wide range of seasonal time series data. The formation of the Seasonal ARIMA model is by adding some further seasonal terms to the original ARIMA models, and it is written as following:

$$ARIMA(p, d, q)(P, D, Q)_m$$

The value m refers to the number of periods per season. The uppercase notations represent the seasonal parts of the model, as P refers to seasonal auto regression parameter, D refers to the seasonal differencing parameter, and Q refers to the seasonal moving average parameter, while the lowercase notations represent the non-seasonal parts of the model as described in the previous section. The terms included in the seasonal part of the model are very similar to the non-seasonal terms of the model, however they contain backshifts of the seasonal period. The seasonal ARIMA model formula can be written as:

$$(1 - \phi_p B)(1 - \phi_p B^m)(1 - B)(1 - B^m)y_t = (1 + \theta_q B)(1 + \theta_q B^m)e_t \quad (3)$$

Here $(1 - \phi_p B)$ represents the non-seasonal auto regression parameter, $(1 - \phi_p B^m)$ represents the seasonal auto regression parameter, $(1 - B)$ represents the non-seasonal difference and $(1 - B^m)$ represents the seasonal difference. The variable m refers to the number of periods per season. For the right side of the equation, $(1 + \theta_q B)$ represents the non-seasonal moving average parameter, and $(1 + \theta_q B^m)$ represents the seasonal moving average parameter. In Eq. 2.1, the additional seasonal components are simply multiplied with the non-seasonal components.

The SARIMA method has been used for wind speed forecasting in conjunction with other methods in [9]. It has also been used for time series forecasting in various other application domains [10–13].

2.2 Machine Learning Methods

Machine learning methods have drawn a great attention in the last few years as they have created themselves as a great competitors to the classical statistical models in prediction and forecasting community. As a significant number of studies have been done in forecasting time series data using machine learning models, there are multiple number of studies showing that some of the machine

learning models have improved forecasting performance and accuracy than the traditional statistical models [14]. Followings are the brief descriptions of the ANN and SSA methods that we explore in this research.

ANN: Artificial Neural Networks (ANN) are forecasting methods which are based on simple mathematical calculations of the brain as they allow for complex nonlinear relationships between the predictors and their response variables. A neural network is described as a network consisting of neurons which are organized in a number of layers. The first layer (or the input layer) includes the predictors, and this layer forms the bottom layer. The last layer (or the output layer) forms the top layer. In between, there may be some intermediate layers which are called hidden layer. The simplest example of a neural network contain no hidden layers and it is equivalent to a linear regression. Figure 1 shows an example of a neural network. The model selected in this experiment was an ANN with single hidden layer and maximum number of weight as 2,000.

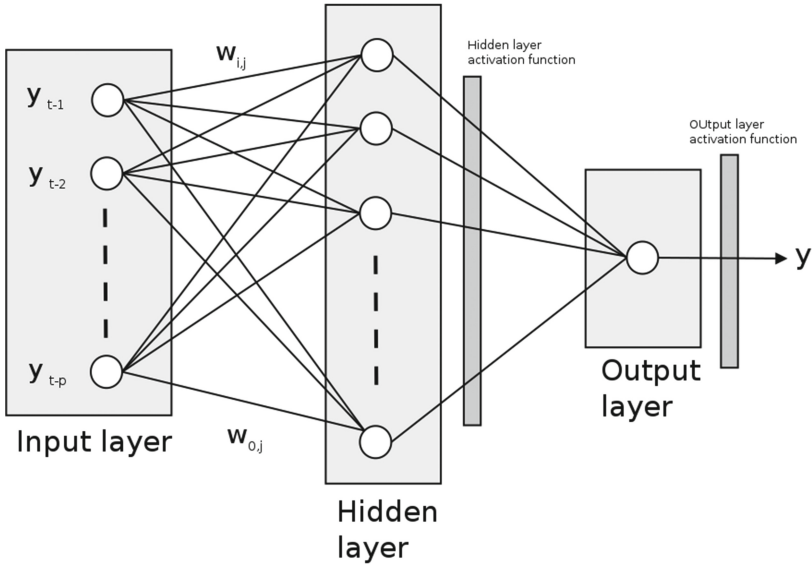


Fig. 1. Example of artificial neural networks.

As shown in Fig. 1, a neural network model is defined by a network of three layers of a simple processing unit. These units are connected to each others by cyclic links as shown. The relationship between the inputs (Y_{t-1}, \dots, Y_{t-p}) and the output (Y_t) has the following mathematical representation:

$$Y_t = w_0 + \sum_{j=1}^q w_j \times g(w_{0j} + \sum_{i=1}^p w_{ij} \times Y_{t-i}) + \varepsilon_t \quad (4)$$

Here w_{ij} ($i = 0, 1, 2, \dots, p$, $j = 1, 2, \dots, q$) and w_j ($j = 1, 2, \dots, q$) are the model parameters which are often called by the connection weights; p is representing the number of input nodes; and q is representing the number of hidden nodes. ε_t is a very small cumulative number that varies depending on the number of input nodes. The function $g()$ performs a nonlinear functional mapping of the presented data [15].

Different flavors of ANN have been used for wind speed forecasting in different geographical locations such as Mexico, Nigeria, and Turkey [16–18] as well as in other application domains [19–22].

SSA: Singular spectrum analysis (SSA) is relatively a recent approach used for modeling and forecasting time series. It is a non-parametric spectral estimation method that uses no assumptions for the underlying process [23]. The SSA model combines elements of multivariate geometry, classical time series analysis, signal processing as well as dynamical systems. Moreover, it can decompose the original time series data to its major independent components, such as trend, oscillation behavior (periodic and quasi-periodic components) as well as noise, efficiently, which are used afterwards for time series forecasting [23, 24].

SSA models have an important advantage in which it allows for producing forecasts for individual components of the time series data or the rebuilt series upon the reconstruction of the time series data under study. This advantage is useful if the aim is to make a prediction about, for instance, the trending/deterministic component of the time series data without taking into account the variability resulting from other sources [25]. The model selection in this experiment was based on default parameter selection of SSA model.

SSA has been used for forecasting in the domains of forecasting electricity load, system failure, and hydrology [23, 24, 26]. However, to our best knowledge, it has not been used for wind speed forecasting before. Our study is the first to use SSA for the purpose of wind speed forecasting.

3 Experimental Setup

3.1 Dataset

The data set used in this study is an hourly time series wind speed data of Abu Dhabi city, UAE (Fig. 2) collected at the Abu Dhabi International Airport. The data set is relatively big as it is recorded from 1982 to 2010 in an hourly base, which makes it a total of 25,4184 data points (Fig. 3).

We are forecasting the wind speed data from 1 h ahead to 1 week (168 h) ahead basis. In order to start the implementation of the forecasting models, the data set should be first divided into two separate sets: the training set and the testing set. The division was based on the number of forecasting hours required for getting our results which is one week ahead. In this case the training set contains 25,4016 data points, and the testing set consists of $24 \times 7 = 168$ data points.



Fig. 2. United Arab Emirates (UAE) and its capital Abu Dhabi. (Source: <https://www.cia.gov/library/publications/the-world-factbook/geos/ae.html>)

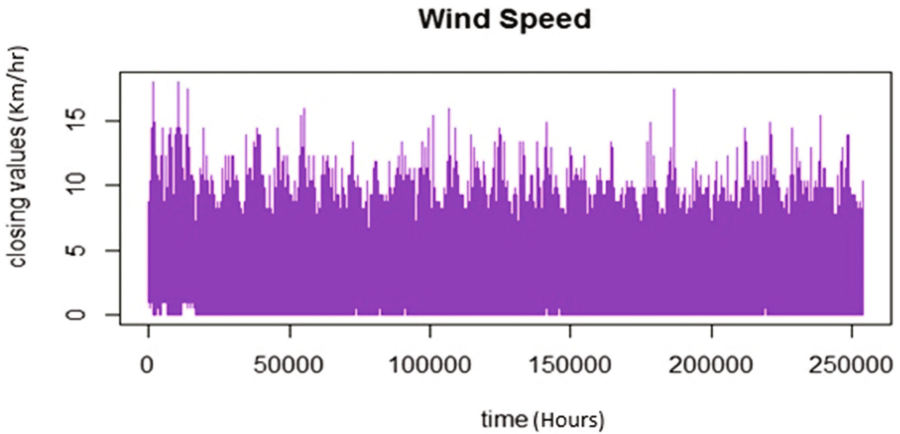


Fig. 3. Line plot of Abu Dhabi wind speed data with 25,4184 data points (recorded hourly from 1982 to 2010).

3.2 Evaluation Criteria

The performances of the forecasting algorithms are evaluated by five evaluation criteria:

1. Mean Error (ME),
2. Mean Absolute Error (MAE),
3. Mean Percentage Error (MPE),
4. Mean Absolute Percentage Error (MAPE), and
5. Root-Mean-Square Error (RMSE).

$$\text{ME} = \frac{1}{n} \sum_{i=1}^n x_i - \hat{x}_i \quad (5)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i| \quad (6)$$

$$\text{MPE} = \frac{100}{n} \sum_{i=1}^n \frac{x_i - \hat{x}_i}{x_i} \quad (7)$$

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^n \frac{|x_i - \hat{x}_i|}{x_i} \quad (8)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (9)$$

In Eqs. 5 to 9, n stands for the number of forecasting data points, x_i ($i = 1, \dots, n$) is the actual value of the i^{th} data point, and \hat{x}_i ($i = 1, \dots, n$) is the forecasted value of the i^{th} data point. Since all the five criteria are the measurement of the forecasting errors of different types, a lower value indicates a better performance of the forecasting method.

In our particular experimental setup, it is always the case that $n = 1$ because with each method we are carrying out 168 individual point forecasts for 168 testing data points (i.e., 1 hour-ahead to 168 hour-ahead forecasts). It should be noted that if $n > 1$, the criteria ME and MPE will not make much sense. For example, if $n = 2$ and $x_1 - \hat{x}_1 = +1$ and $x_2 - \hat{x}_2 = -1$, then $\text{ME} = 0$ even though there are errors on the two individual point forecasts.

4 Experimental Results

We will discuss the experimental results of the four methods of our choice: ARIMA, SARIMA, ANN, and SSA on the Abu Dhabi wind speed dataset in this section.

For ARIMA, initially a number of possible models have been examined in terms of parameters selection as shown in Fig. 4. This is to determine the best model according to the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) values [27], where the best model is the one which has the lowest AIC and BIC values. After examining all possible models, it is observed that the best ARIMA model, based on the AIC value, is ARIMA(4,1,4).

Similarly for SARIMA, initially a number of possible models have been examined in order to determine the best model as shown in Fig. 4 according to the AIC values. After examining all possible models, it is found that the best SARIMA model, based on the AIC value, is SARIMA(4,1,4)(1,0,4)₂₄ (Fig. 5).

The performances of the 4 methods in terms of the five evaluation criteria (as described in Sect. 3.2) are presented in Figs. 6, 7, 8, 9 and 10. In each figure, the

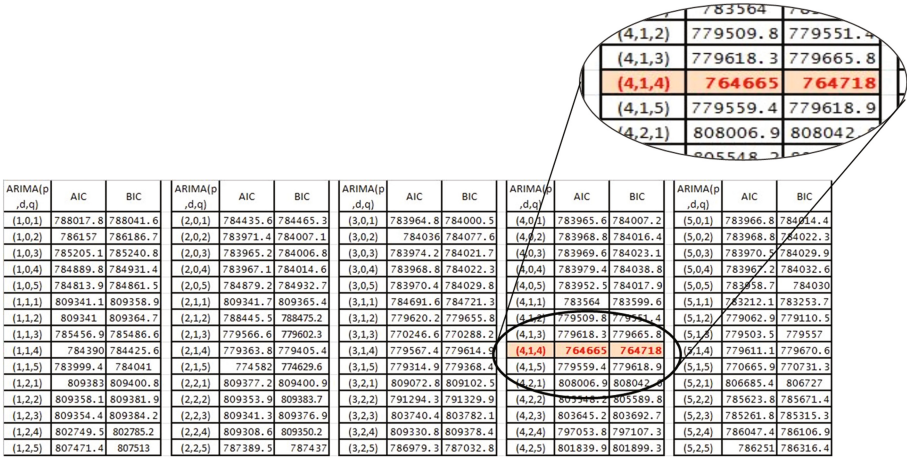


Fig. 4. ARIMA model selection based on AIC and BIC.

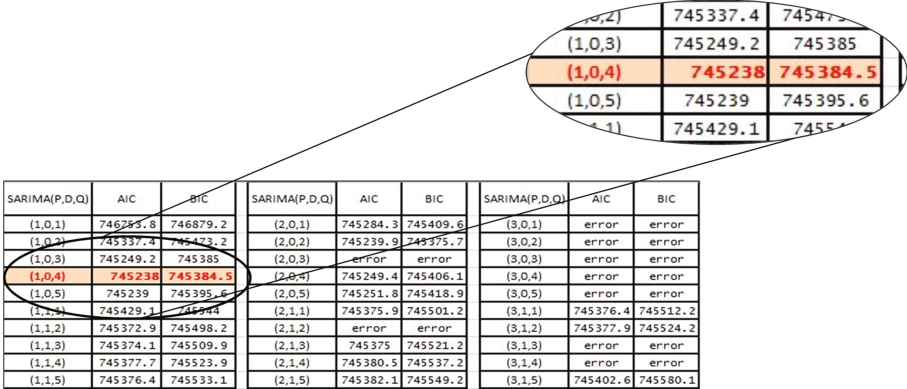


Fig. 5. SARIMA model selection based on AIC and BIC.

X-axis represents the forecasting time horizon, which ranges from 1 h to 1 week (168 h) ahead.

Figures 6 and 8 suggest that all the methods trend to over-forecast (i.e., $\hat{x} > x$) in most cases.

Moreover it can be observed from Figs. 6, 7, 8, 9 and 10 that the performances of ARIMA, SARIMA, and ANN are generally better than those of SSA for short-term forecasts (i.e., 1 hour-ahead to 24 hour-ahead). However, the results of SSA are better than the three competitions for the long-term forecasts (24–168 hour-ahead forecasts).

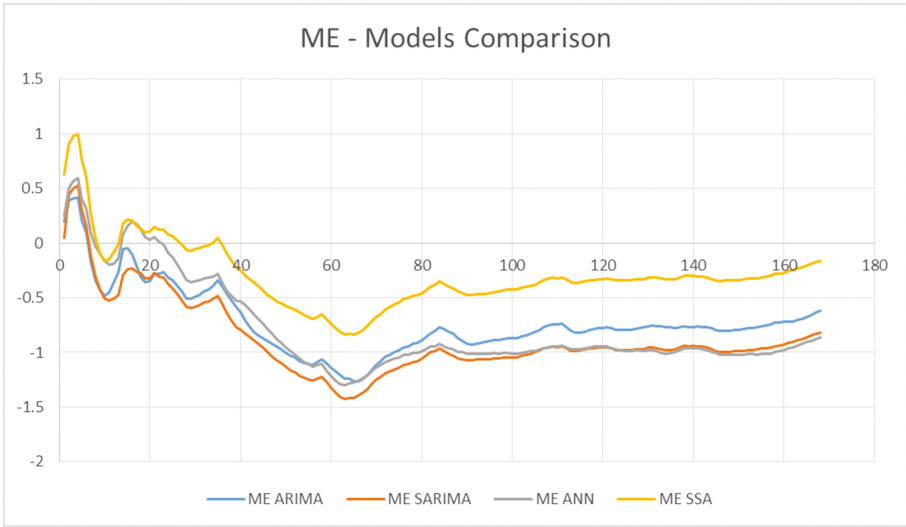


Fig. 6. Mean Error (ME).

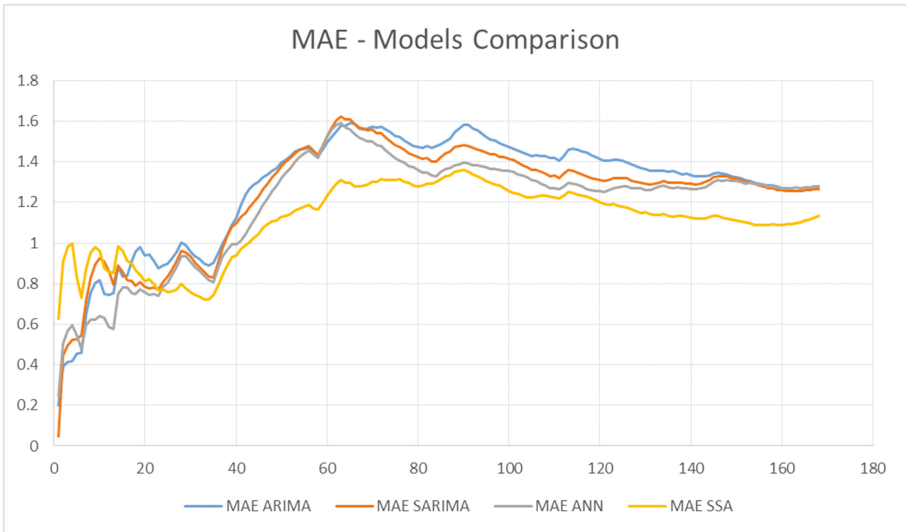


Fig. 7. Mean Absolute Error (MAE).

The relative forecasting accuracies of all the four methods relative to the actual wind speed are depicted in Fig. 11. It can be noted that none of the methods (including SSA) are able to accurately forecast the spikes in the wind speed, thus highlighting the inherent difficulty of the wind speed forecasting problem.

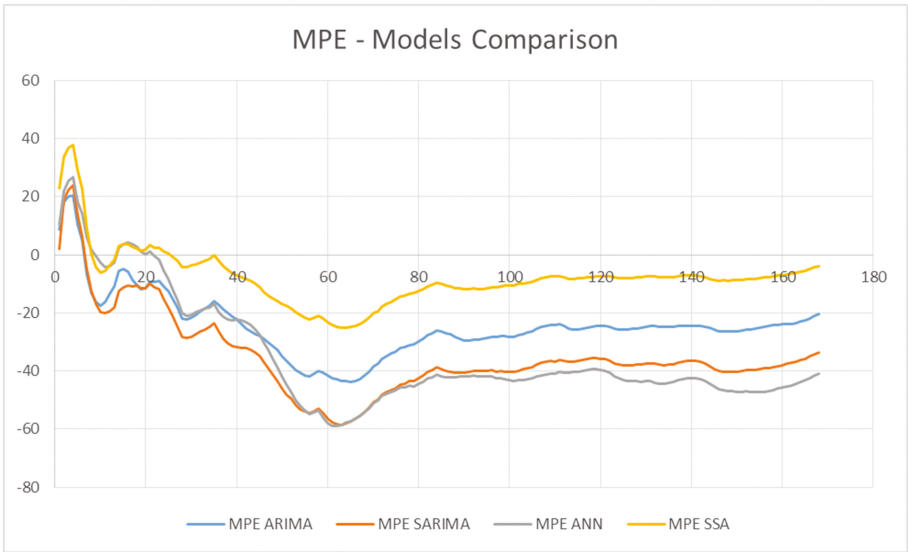


Fig. 8. Mean Percentage Error (MPE).

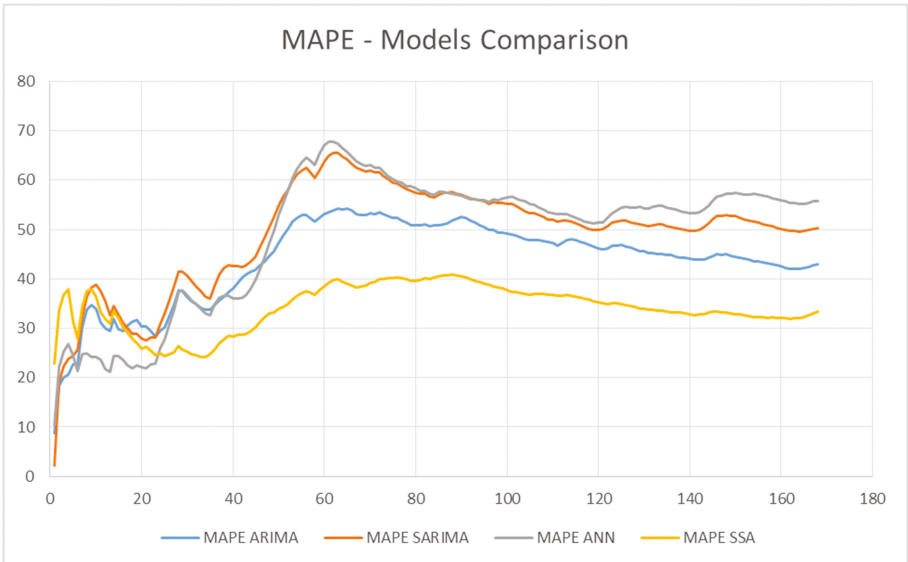


Fig. 9. Mean Absolute Percentage Error (MAPE).

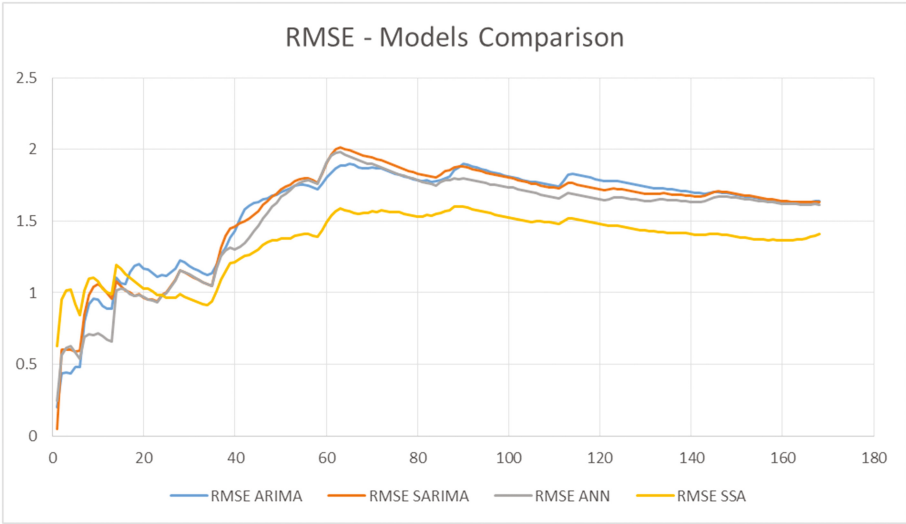


Fig. 10. Root-Mean-Square Error (RMSE).

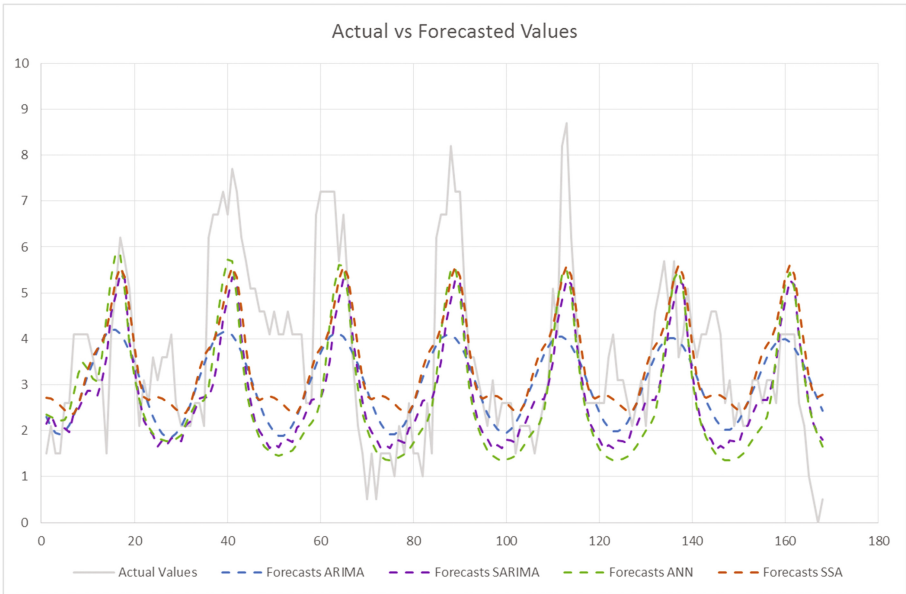


Fig. 11. Actual vs. forecasted win speed values by the four methods.

5 Conclusion

In this paper, we have studied four forecasting models in forecasting wind speed data of Abu Dhabi city. The models are divided into two categories, two from statistical methods (ARIMA and SARIMA) and the other two from machine learning methods (ANN and SSA). Then we compared these four models in order to determine the most efficient model in forecasting the wind speed data. The results of our study showed that the forecasting model SSA had the most accurate forecasted values comparing to the other three models, however, it also showed that those three models, ARIMA, SARIMA and ANN, had better results for the first few hours (around 24 h), which indicates that ARIMA, SARIMA, and ANN models are suitable for short-term forecasting, while SSA model is suitable for long-term forecasting. The findings of our research could contribute in defining the fitting forecasting model in terms of short-term forecasting or long-term forecasting.

Acknowledgement. The authors would like to acknowledge and appreciate the support of Professor Taha Ouarda in providing the Abu Dhabi wind data set, and also thank Dr. Aamna Mohammed Al Shehhi for her support and assistance with the statistical models.

References

1. Perera, K.S., Aung, Z., Woon, W.L.: Machine learning techniques for supporting renewable energy generation and integration: a survey. In: Woon, W.L., Aung, Z., Madnick, S. (eds.) DARE 2014. LNCS (LNAI), vol. 8817, pp. 81–96. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13290-7_7
2. Al Dhaheiri, K.M.A.M.: Forecasting wind speed in Abu Dhabi using statistical modelling and machine learning approaches. Master's thesis, Masdar Institute of Science and Technology, Abu Dhabi, UAE (2016)
3. Shumway, R.H., Stoffer, D.S.: Time Series Analysis and Its Applications: with R Examples, 3rd edn. Springer, New York (2011)
4. Montgomery, D.C., Jennings, C.L., Kulahci, M.: Introduction to Time Series Analysis and Forecasting, 1st edn. Wiley, Somerset (2008)
5. Wei, W.W.S.: Time Series Analysis - Univariate and Multivariate Methods, 2nd edn. Pearson Addison Wesley, Boston (2006)
6. Palomares-Salas, J.C., et al.: Comparison of models for wind speed forecasting. In: Proceedings of the 9th International Conference on Computational Science, pp. 1–5 (2009)
7. Liu, H., Tian, H.Q., Li, Y.F.: Comparison of two new ARIMA-ANN and ARIMA-Kalman hybrid methods for wind speed prediction. *Appl. Energy* **98**, 415–424 (2012)
8. Yunus, K., Thiringer, T., Chen, P.: ARIMA-based frequency-decomposed modeling of wind speed time series. *IEEE Trans. Power Syst.* **31**, 2546–2556 (2016)
9. Zhang, J., Wei, Y., Tan, Z.F., Wang, K., Tian, W.: A hybrid method for short-term wind speed forecasting. *Sustainability* **9** (2017). Article number 596
10. Chen, K.Y., Wang, C.H.: A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan. *Expert Syst. Appl.* **32**, 254–264 (2007)

11. Ruiz-Aguilar, J.J., Turias, I.J., Jiménez-Come, M.J.: Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting. *Transp. Res. Part E Log. Transp. Rev.* **67**, 1–13 (2014)
12. Gocheva-Ilieva, S.G., Ivanov, A.V., Voynikova, D.S., Boyadzhiev, D.T.: Time series analysis and forecasting for air pollution in small urban area: an SARIMA and factor analysis approach. *Stochast. Environ. Res. Risk Assess.* **28**, 1045–1060 (2014)
13. Jeong, K., Koo, C., Hong, T.: An estimation model for determining the annual energy cost budget in educational facilities using SARIMA (seasonal autoregressive integrated moving average) and ANN (artificial neural network). *Energy* **71**, 71–79 (2014)
14. Bontempi, G., Ben Taieb, S., Borgne, Y.-A.: Machine learning strategies for time series forecasting. In: Auafeure, M.-A., Zimányi, E. (eds.) *eBISS 2012*. LNBIP, vol. 138, pp. 62–77. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36318-4_3
15. Khashei, M., Bijari, M.: An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Syst. Appl.* **37**, 479–489 (2010)
16. Cadenas, E., Rivera, W.: Short term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural networks. *Renew. Energy* **34**, 274–278 (2009)
17. Fadare, D.A.: The application of artificial neural networks to mapping of wind speed profile for energy application in Nigeria. *Appl. Energy* **87**, 934–942 (2010)
18. Filik, U.B., Filik, T.: Wind speed prediction using artificial neural networks based on multiple local measurements in Eskisehir. *Energy Procedia* **107**, 264–269 (2017)
19. Allende, H., Moraga, C., Salas, R.: Artificial neural networks in time series forecasting: a comparative analysis. *Kybernetika* **38**, 685–707 (2002)
20. Samsudin, R., Shabri, A., Saad, P.: A comparison of time series forecasting using support vector machine and artificial neural network model. *J. Appl. Sci.* **10**, 950–958 (2010)
21. Mohammed, A.A., Aung, Z.: Ensemble learning approach for probabilistic forecasting of solar power generation. *Energies* **9** (2016). Article number 1017
22. Neupane, B., Perera, K.S., Aung, Z., Woon, W.L.: Artificial neural network-based electricity price forecasting for smart grid deployment. In: *Proceedings of the 2012 IEEE International Conference on Computer Systems and Industrial Informatics*, pp. 1–6 (2012)
23. Rocco, S.C.M.: Singular spectrum analysis and forecasting of failure time series. *Reliability Eng. Syst. Safety* **114**, 126–136 (2013)
24. Afshar, K., Bigdeli, N.: Data analysis and short term load forecasting in Iran electricity market using singular spectral analysis (SSA). *Energy* **36**, 2620–2627 (2011)
25. Hassani, H., Thomakos, D.: A review on singular spectrum analysis for economic and financial time series. *Stat. Interface* **3**, 377–397 (2010)
26. Marques, C.A.F., et al.: Singular spectrum analysis and forecasting of hydrological time series. *Phys. Chem. Earth Parts A/B/C* **31**, 1172–1179 (2006)
27. Burnham, K.P., Anderson, D.R.: Multimodel inference: understanding AIC and BIC in model selection. *Sociol. Meth. Res.* **33**, 261–304 (2004)

NWP Ensembles for Wind Energy Uncertainty Estimates

Alejandro Catalina^(✉) and José R. Dorransoro

Dpto. Ing. Informática and Instituto de Ingeniería del Conocimiento,
Universidad Autónoma de Madrid, Madrid, Spain
`alejandro.catalina@estudiante.uam.es`

Abstract. Numerical weather predictions (NWP) ensembles, i.e., probabilistic variants of NWP forecasts, can be a useful tool to improve the quality of renewable energy predictions as well as to provide useful estimates of uncertainty in NWP-based energy forecasts. In this work we will consider the application of the NWP ensembles provided by the European Center for Medium Weather Forecasts (ECMWF) to deal with these issues. We shall consider both local prediction at a single wind farm as well as the wide area prediction of wind energy over Peninsular Spain and show that while deterministic forecasts have an edge over ensemble based ones, these can be used to derive quite good uncertainty intervals.

Keywords: Numerical weather prediction · Probabilistics ensembles · Wind energy · Multilayer perceptrons · Support vector regression · Uncertainty estimates

1 Introduction

The worldwide efforts towards clean energy sources are leading to a very fast expansion of wind and solar energy productions. While partially driven by subsidies at the beginning, wind and solar energy are achieving an economic logic of their own that results in considerable investments in large solar and wind farms, where installed power on the hundreds of MW are becoming routine, particularly off-shore. As a consequence, their accurate prediction is a key tool to manage and integrate these energy sources and thus it gets a constant research attention.

Two goals are particularly important. The first is, of course, to make energy predictions as accurate as possible but, given the flickery nature of weather conditions, a second crucial goal is to endow energy forecasts with reliable uncertainty estimates. For one to a few (6 to 10) days ahead, Numerical Weather Predictions (NWP), such as those of the European Center for Medium Weather Forecasts (ECMWF), are the key source to derive wind energy estimates. Several modeling methodologies can be applied but Machine Learning-based methods are steadily gaining ground with tools such as Multilayer Perceptrons (MLPs), either shallow or deep [5], Support Vector Regression (SVR) [19], Random Forests and Gradient Boosting Regression [2], Gaussian Processes [6] or, more generally,

ensemble methods [10,16]. In fact, it might be said that the combination of NWP inputs and ML regression models are progressively defining the state of the art in wind energy prediction and this will also be our scenario here, building both MLP and SVR models on top of the ECMWF's NWP products.

The standard NWP input are the so-called deterministic forecasts, based on the best estimate of the model equations and their initial conditions and representing what could be taken as the most accurate weather prediction. As such, they are run using the finest spatial and temporal resolution available at the ECMWF (currently about 0.125° and 1 h); of course, the numerical effort of a standard, 10 day deterministic forecast is very high. However, weather forecasts are also subject to a considerable degree of uncertainty and NWP ensembles are used to partially address this. In the ECMWF case they are currently given by a set of 51 different forecasts built running different NWP models at a 0.25° spatial and 3 h temporal resolutions. Each ensemble model corresponds to slightly random different initial conditions and NWP model parameters which seek to capture different possible weather evolutions, particularly at longer prediction periods and, thus, give a set of possible weather scenarios. One of these models, the so called control forecast, is the counterpart at coarser space (about 0.25°) and time (about 3 h) resolution of the deterministic forecast, and the other 50 correspond to perturbations of the control one.

Thus, NWP ensembles offer at first sight a richer scenario in which to perform renewable, wind in our case, energy predictions, although at a coarser time-space resolution. This suggests some venues for their exploitation. A first obvious question is whether using ensemble information results in better wind energy predictions than those achievable using deterministic NWP alone. A rule of thumb seems to be that for shorter forecasting horizons an ensemble based forecast of weather variables should have a precision similar to that of the deterministic one, while for longer horizons (and hence more uncertainty in predictions), ensemble forecasts should be more reliable. In principle, this might suggest a similar behavior when applied to wind energy prediction. On the other hand, a possible advantage of ensemble forecasts is that they should naturally give uncertainty intervals for the energy predictions with a time varying width, in contrast with the basically fixed width estimates derived from the percentiles of past prediction residuals.

We will address both issues in this work, where we will consider wind energy predictions both for a single wind farm, Sotavento, in northwestern Spain, and also over the much wider area of Peninsular Spain. We will build energy predictions using two powerful Machine Learning (ML) models, Multilayer Perceptrons (MLPs) and Support Vector Regression (SVR) but, rather than proposing competing ML procedures, this has more to do with a first data-based analysis of what can be achieved using ensembles to enhance NWP based energy forecasts. We will study two energy prediction scenarios, namely predictions for the same day where NWP forecasts are produced, and those for two days afterwards. Our main conclusions are the following:

- Energy predictions derived from NWP ensembles are competitive with those derived from the deterministic forecast but the latter still have an edge.
- Ensemble predictions can be used to provide good energy uncertainty estimates, although the direct use of the ensemble prediction spread will give poor results. Instead, it is much better to center them on the deterministic forecast and apply a suitable calibration of the spreads of the ensemble predictions.

The rest of the paper is organized as follows. We briefly review in Sect. 2 ensemble NWP forecasts, particularly those provided by the European Center for Medium Weather Forecasts (ECMWF), and MLP and SVR models in Sect. 3. Numerical results are given in Sect. 4 and the paper ends in Sect. 5 with a short discussion, the main conclusions and proposals for further work.

2 NWP Ensembles

We first point out the different meanings the word “ensemble” may have in the Machine Learning (ML) and meteorology communities. ML ensembles are aggregations of single ML models which are built independently and randomly. Randomness seeks to endow them with a probabilistic structure while independence seeks to lower the variance of the aggregated estimator and, hence, lead to better predictions. These ML ensembles are widely used in many fields, and also for wind [10] or solar [14] energy, where they can either improve single model forecasts or provide uncertainty estimates for them. See the recent paper [16] for a very complete overview of the current state of the art of ML ensemble based energy predictions.

On the hand, and while ultimately having similar goals, NWP ensembles seek to capture the uncertainty inherent to the weather behavior. In [12] an overview of NWP ensemble forecasting focused on the Ensemble Prediction System (EPS) of the ECMWF is given, with a discussion of possible uncertainty sources and how these may be incorporated through perturbations of the initial conditions of a NWP model. Roughly speaking, the general idea is to consider as initial states of the NWP ensembles those perturbations which should result in larger increases of initial errors. In particular, this initialization does not follow statistical sampling procedures, which would be too costly given the high complexity and large dimension of weather models; thus, an ensemble interpretation strictly based on probability grounds could be problematic.

Currently ECMWF’s EPS produces forecasts derived from 51 integrations with an approximately 32 Km spatial resolution and a 3 h time resolution for up to 10 days ahead. 50 of these forecasts are derived from initial conditions which are slightly different but close to the ECMWF’s best estimate of the initial state of the atmosphere, which are used to obtain the ensemble’s **control** forecast. The resulting spread around the control gives a measure of potential weather evolution uncertainty. General guidelines for the use of NWP ensembles either on their own or jointly with deterministic forecasts are given in [1].

As previously mentioned, NWP ensembles appear very attractive for renewable energy prediction where good uncertainty estimates are almost as important as precise point forecasts, but some caveats should be taken into account. In principle, the spread, i.e., the differences between ensemble members, should increase with the time horizons but, however, the EPS forecasts tend to be under dispersive [18], i.e., their spreads tend to be too small, at least for the shorter term horizons. As a consequence, it may be the case [7] that for these shorter horizons, the spread of ensemble-derived wind energy forecasts may be too tight to compute some quantiles. Moreover, and as also pointed out in [7], the quantiles so obtained are not likely to be correct, at least for short time horizons in the following sense: if at a given farm location one ranks between 0 and 100 the actual power values inside the ensemble forecast spread, one would expect this rank to be uniformly distributed on the $[0, 100]$ interval. However, as discussed in [7], Sect. 4.1, the corresponding QQ plots show this not being the case. In fact, any ensemble model will have its own systematic bias and, thus, will not capture statistical uncertainty. Calibration techniques, i.e., methods to correct bias in variance estimates, could be used to improve on this and to make under dispersion less pronounced at longer horizons.

We will address both the quality of ensemble energy forecasts and their usefulness to estimate uncertainty intervals in the following sections, but besides the concrete numerical results we give, one must also conclude that using NWP ensembles to improve on renewable energy predictions or to give power uncertainty estimates requires a careful handling. This should not be surprising. First of all, the ultimate goal of ensemble prediction systems is to be useful tools to help in weather related decisions, such as extreme weather situations or disaster prevention. As a consequence, their real value does not lie in providing probability forecasts correct by themselves in a statistical sense and even less so in any renewable energy application. This does not mean, however, that they should be discarded outright but, rather, that their usefulness in this field has to come from a not too narrow consideration of their nature but, instead, on how their information can be put to use to help energy producers or system operators.

3 MLP and SVR Models

In our experiments we will work with fairly straightforward neural network (NN) and support vector regression (SVR) models, which we briefly review next. For a given size N sample $\{(x^1, y^1), \dots, (x^N, y^N)\}$ in NN regression we try to minimize the L_2 regularized loss

$$e_{NN}(w) = \frac{1}{2N} \sum_p (y^p - f(x^p, w))^2 + \frac{\alpha}{2} \|w\|_2,$$

where by $f(x, w)$ we denote the output on x of an NN whose architecture determines a weight set w . Many possible architecture options are available; here we will work with a fixed number of fully connected layers, each of which with the

same number of hidden units. The general, classic theory of such networks is very well known (see for instance [3], Chap. 5) but the recent developments on deep NNs have resulted in a number of important enhancements, some of which we will apply here. For instance, we will use ReLUs [9] as the activation functions, Glorot–Bengio weight initialization [8] and minibatch–based Adam [11] as the optimization method. All these options are available in neural network frameworks such as Keras [4] or Scikit–learn [15], whose `MLPRegressor` class we will use here.

To minimize the computational effort in hyper–parameter estimation, we have considered fixed MLP architectures with 2 hidden layers of 100 units each for the Sotavento problem (similar results are obtained with 4 layers), and with 4 hidden layers of 1000 units each for the REE problem. We have also fixed the activation function (ReLUs) and used the default hyperparameters for the Adam solver. This leaves the L_2 penalty parameter `alpha` as the only one to hyper–parameterize, which we do by fixed fold cross validation with 2013 as the train fold and 2014 as the validation fold, using again Scikit–learn’s tools. We will consider a logarithmic equi–spaced grid with values $\alpha = 2^k$ with $-20 \leq k \leq 8$.

On the other hand, and while it is more often written as a constrained optimization problem, the minimization problem to be solved for linear SVR (i.e., the primal problem) has the following cost function

$$e_{SVR} = \sum_p [y^p - w \cdot x^p - w_0]_\epsilon + \frac{1}{2C} \|w\|^2 \quad (1)$$

using again L_2 regularization and the ϵ -insensitive loss $[y - \hat{y}]_\epsilon = \max\{|y - \hat{y}| - \epsilon, 0\}$, which defines an ϵ -wide, penalty-free “error tube” around the model. As mentioned, (1) is usually rewritten as a constrained minimization problem which is then transformed using Lagrangian theory into a much simpler dual problem, the one actually being solved [17]. As it is well known, the dual problem only involves patterns through their dot products which, in turn, makes it possible to apply the kernel trick [17], replacing the linear dot products $x \cdot x'$ with the values $k(x, x')$ of a positive definite kernel k . Going back to the primal problem, the final model can be written as

$$f(x) = w_0^* + \sum_{\alpha_p^* > 0} \alpha_p^* k(x^p, x), \quad (2)$$

where the α_p^* are the dual optimal solutions and w_0^* is derived through the KKT conditions; see [17] for more details.

As it is usually done, we will work with the Gaussian kernel $e^{-\gamma \|x - x'\|^2}$. This results in an SVR model having three hyper–parameters: the regularization coefficient C (which is the penalty constant in the standard constrained minimization SVR formulation), the width ϵ of the insensitivity tube and the γ parameter for the kernel width. All three are crucial to achieve good model performance and we will also use here Scikit–learn’s grid search cross validation tools working on a equi–spaced logarithmic scale. More precise, for C we will

consider $C = 4^k$ values in the range $-5 \leq k \leq 5$, for ϵ the values $4^{-k}\sigma$, with $1 \leq k \leq 5$ and σ being the targets' standard deviation, and for γ the values $\frac{4^k}{d}$, with $-2 \leq k \leq 2$ and where d denotes pattern's dimension.

We will use both MLPs and SVRs to build energy predictions based on the same day deterministic and control NWP forecasts. More precisely, we will work with the forecasts produced at UTC hour 00 each day to hyperparameterize and build our models using as features the NWP forecasts for the UTC hours 00 to 23 of each such day and as targets the corresponding wind energy values. Hourly deterministic NWP forecasts are directly available from the ECMWF and we interpolate the 3-hour ensemble forecasts to a one hour resolution using cubic splines. The rationale for this "same day" modeling is precisely that NWP forecasts should be more accurate for such 0-day horizons and, consequently, so would be their associated ML models. From now on we will refer to these same day models as the deterministic and control MLP and SVR models. The same day deterministic model will also be applied to deterministic two day ahead NWP forecasts to obtain the corresponding energy forecasts. For ensembles we will similarly apply the control model to obtain both the same day ensemble energy prediction and the ensemble and control two day ahead forecasts.

4 Numerical Experiments

In our experiments we will work with NWP data for 2013, 2014 and 2015, for which we have downloaded the deterministic and control forecasts from the ECMWF's MARS repository and which we will use for training, validation and test purposes respectively. The remaining 50 ensemble forecasts will be used only for test purposes and, therefore, we have download only these forecasts for 2015. As mentioned we will use wind energy values for the Sotavento wind farm, which makes them available on its web portal, and for the wind energy of Peninsular Spain, kindly provided by Red Eléctrica de España, REE, Spain's TSO.

4.1 Deterministic vs Ensemble Energy Forecasts

Before studying uncertainty intervals, we will perform a straight comparison of the MLP and SVR deterministic energy forecasts with those of the corresponding control forecasts and also with the mean and median of the ensemble based energy forecasts. In all cases we have used the following NWP variables:

- U and V wind speed components and their modules at 10 and 100 m.
- Surface pressure.
- Temperature at 2 m.
- Wind to power conversion at 10 and 100 m, using a generic power curve which saturates at 20 m/s and has a cutoff at 25 m/s.

As customary we normalize NWP variables to 0 mean and 1 standard deviation; energy values are in a 0-100 scale. While, as mentioned, deterministic forecasts are derived using a finer grid, in order to minimize data volumes we will

Table 1. MAE and RMSE results for both MLP and SVR models for same and two day ahead forecasts.

		Sotavento		REE	
		MAE	RMSE	MAE	RMSE
MLP	Det	5.86	8.50	2.76	3.66
	Control	6.84	9.12	2.68	3.67
	Mean	6.01	8.78	2.68	3.65
	Median	5.99	8.78	2.68	3.64
	Det (2d)	7.25	10.50	4.60	5.77
	Mean (2d)	8.37	11.35	6.17	7.55
	Median (2d)	8.21	11.34	6.14	7.54
SVR	Det	5.80	8.52	2.54	3.37
	Control	6.34	9.13	2.56	3.40
	Mean	5.85	8.72	2.56	3.38
	Median	5.84	8.74	2.56	3.39
	Det (2d)	7.19	10.57	4.63	5.80
	Mean (2d)	9.31	12.20	6.70	8.14
	Median (2d)	9.23	12.22	6.66	8.10

work with deterministic and ensemble NWP features from a 0.25° resolution for Sotavento and from a 0.5° one for REE. For Sotavento we will work with an 8×15 rectangular grid approximately centered at the wind farm for both deterministic and ensembles data, and with a 18×29 grid that covers Peninsular Spain for the REE data. Pattern dimension would thus be in principle $1,200 = 8 \times 15 \times 10$ for Sotavento and $5,220 = 18 \times 29 \times 10$ for REE. We point out that model complexity is quite different for SVRs and MLPs. For SVRs the maximum number of α multipliers in (1) is the sample size, 17,520 in our case, corresponding to the number of hours in two years. On the other hand, and just for Sotavento, the number of weights (without counting bias terms) in our NN models is 130,100 while for REE we have 8,221,000 weights. Since we work with hourly data (after interpolation for the ensemble NWP forecasts), the number of yearly patterns is 8,760 for both problems. After hyper-parameterization we will train the MLP and SVR models over the combined data of 2013 and 2014 and apply them on 2015. SVRs solve a convex minimization problem and their solutions are thus essentially unique. However NN training starts at random initial weights and mini-batch training adds further randomness. Hence, there is not a unique final model and we will take advantage of this by training independently 10 NNs and averaging their predictions. This should lead to a degree of variance reduction and, thus, better forecasts.

In Table 1 we give MAE and RMSE values for all models considered for both the same day and the two day ahead scenarios. As it can be seen, SVR performs better than MLP for all configurations, except for the 2 day ahead

Table 2. Uncertainty percentage targets (second column) and percentages of actual energy values inside the corresponding uncertainty intervals for same day and 2 days in advance.

	Confidence	Sotavento		REE	
		0d	2d	0d	2d
MLP	20	20.79	21.42	19.86	20.34
	50	49.70	50.01	53.42	47.07
	80	79.89	80.59	81.40	78.54
	90	90.55	90.22	92.00	89.98
	95	94.89	94.71	96.93	94.54
SVR	20	20.34	20.85	19.29	19.24
	50	51.27	48.71	50.73	47.84
	80	80.18	79.15	81.37	82.13
	90	89.31	89.71	90.73	91.93
	95	94.48	94.58	97.24	95.35

forecasts, where interestingly MLP slightly outperforms SVR for both Sotavento and REE. We also note that the median of the ensembles is usually the second best model after the deterministic one, with the mean often close and observe that the 2day ahead ensemble errors for Sotavento are much closer to those for the same day than the REE errors. It is also interesting to see that all the REE same day models perform similarly (with the exception of the deterministic MLP) while for Sotavento the ensemble median clearly outperforms the control model. Summing things up, these results are to be expected in the sense that, for shorter horizons, deterministic energy forecasts should be slightly better than the best forecast derived from the ensemble predictions. Moreover, SVR models seem to have a slight edge over the MLP ones, despite the much larger number of weights of the latter. This may be due to the very high correlation of the weather variables and it deserves further study.

4.2 Uncertainty Intervals

Here we derive uncertainty intervals for the energy forecasts for which we will use a relatively simple calibration of the ensemble energy spreads which we describe next. Notice that it is sensible to center the uncertainty intervals on the energy forecast under operation (which after the previous Subsection is likely to be a deterministic model) and, also, to take advantage of its known past errors. Thus, for calibration we propose here to combine past deterministic errors and ensemble spreads to compute, for a given confidence percentage s , the uncertainty top and bottom limits using the following formula

$$\widehat{p}_{det} \pm \delta \frac{m_{det}}{m_{ens}} spr_{ens}(s), \quad (3)$$

where \widehat{p}_{det} are the deterministic predictions, δ is a tuning parameter to be fixed, m_{det} is the median of past deterministic errors, m_{ens} is half the median of the ensembles' spread and $spr_{ens}(s)$ is half the difference between the $s/2$ and $100 - s/2$ percentiles of the ensemble energy forecasts. Notice that in (3) the values of δ , m_{det} and m_{ens} are fixed but spr_{ens} changes at each hourly point, resulting in variable width uncertainty estimates.

Since we only have ensemble data for 2015, we will use that year for validation and testing as follows. We first perform a 12-fold consecutive split of 2015, so that each fold should approximately correspond to one month; with a slight abuse of the language we will call these folds "months" from now on. To compute the optimal δ to be used on a given test month, we apply 11-fold cross validation, further splitting the other 11 months in 11 pairs of a 10-month subset for training and the remaining month for validation. We then compute the ratio $\frac{m_{det}}{m_{ens}}$ over the 10 month subsets and consider equi-spaced δ values between 0 and 5 with a step of 0.05. The 0 and 5 extremes ensure that the validation minimum will be attained by a δ^* between them. The corresponding uncertainty boundaries given by (3) are evaluated by applying the following loss function [13]:

$$per_{err}(\delta, s) = |\{\% \text{ of test residuals} \in I_{\delta, s}\} - (100 - s)|, \quad (4)$$

on each of the remaining validation months, where $I_{\delta, s}$ denotes the uncertainty interval defined by (3). Here s is the desired confidence, or percentage value of inside points and $per_{err}(\delta, s)$ gives the percentage of forecasts outside the δ -dependent estimated interval, which ideally should be 0. We then choose the optimal δ^* as the one minimizing the monthly validation errors and compute its test value on the remaining 12-th month using again the loss (4) after re-evaluating $\frac{m_{det}}{m_{ens}}$ over the other 11 months. The final test error we report is just the average of the 12 individual monthly test errors. We point out that optimal δ^* values for Sotavento are similar for the same and two day ahead uncertainty intervals, while for REE the two day δ^* values are slightly smaller, possibly due to larger ensemble spreads. As it should be expected, they increase with the confidence values, and move in the range 1.5, the δ^* value for Sotavento's two days ahead SVR and 20% confidence, to the largest value of 3.3 for Sotavento's same day SVR and 95% confidence. The REE δ^* values are in a 1.6–2.8 range.

The resulting test values for both the same and two day scenarios are shown in Table 2 for Sotavento and REE, which gives for each tested confidence intervals the actual percentage of energy production values inside the estimated uncertainty intervals (i.e., $100 - per_{err}(\delta^*, s)$ for an s percentage target); the ideal value should be precisely s . As we can see, for essentially all the percentages considered the estimated values are quite close to their confidence targets. Also, results for same and two day ahead forecasts are in general quite similar.

As a final remark we would like to point out that both SVR and MLP give fairly good and similar uncertainty estimates. This is likely due because the spread and error statistics for the previous training months also hold for the current test ones and, also, the optimal calibration factor δ^* is essentially the

same for all months (although the spreads spr_{ens} will be different for each month) and increases with the confidence interval.

In Figs. 1 and 2 we show examples of the resulting intervals for same and two day ahead forecasts. We can see that two day estimates are generally wider than the same day estimates, which is to be expected given that same day deterministic forecast have a better accuracy than the two day ahead ones. (The Sotavento intervals are also usually wider, although this is harder to see in the figures given the different scales.)

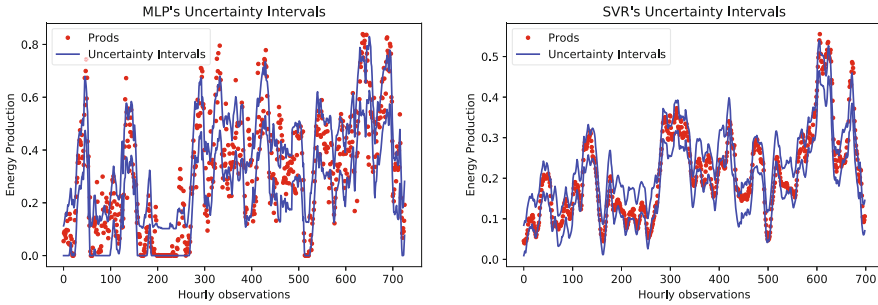


Fig. 1. Same day MLP uncertainty intervals for Sotavento (left) and SVR ones for REE (right) for an 80% confidence on December 2015 where Prods refers to observed energy values.

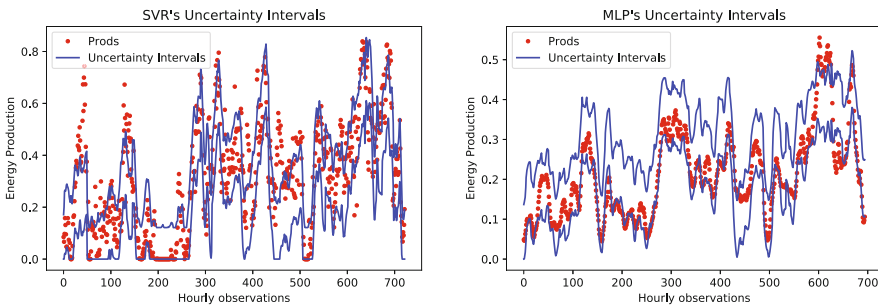


Fig. 2. Two day ahead SVR uncertainty intervals for Sotavento (left) and MLP ones for REE (right) for an 80% confidence on December 2015 where Prods refers to observed energy values.

5 Discussion, Conclusions and Further Work

It is clear from the preceding that the use of NWP ensembles can enrich NWP-based wind energy forecasts, particularly when considering wide areas. Moreover, ensembles allow us to estimate varying width uncertainty intervals with fairly good results. On the other hand, they require a much larger computational

effort, not only because data sizes certainly increase due of the 50 ensemble members but also because prediction times would be also multiplied by 50; this is particularly serious for SVR-based predictions.

Nevertheless, even with the previous caveats, exploring possible advantages of NWP ensembles in wind energy forecasts should be worthwhile. To begin with, we have only considered same and two day ahead scenarios, while ensemble predictions are available for much longer horizons. Moreover, forecasting accuracy could conceivably be improved taking into account the actual spread of ensemble forecasts. In fact, while the correlation between the ensemble's spread and its error is not large, one may still expect some relationship between the spread's magnitude and the stability of the atmosphere. When it is small, the atmosphere's predictability should be higher and its actual state should be near the deterministic forecast. On the other hand, if the ensemble forecasts clearly diverge, they should provide a more sensible prediction under higher uncertainty atmospheric settings. It is thus conceivable that this could be exploited to improve point forecasts, particularly for longer horizons. Finally, while we believe our study of SVR models to be complete enough, there is certainly more work to do over MLPs, where novel convolutional architectures or dropout-based regularization could yield models better than the ones presented here. We are currently working on these ideas.

Acknowledgments. With partial support from Spain's grants TIN2013-42351-P, TIN2016-76406-P, TIN2015-70308-REDT and S2013/ICE-2845 CASI-CAM-CM; work also supported by project FACIL-Ayudas Fundación BBVA a Equipos de Investigación Científica 2016 and the UAM-ADIC Chair for Data Science and Machine Learning. The first author is kindly supported by the UAM-ADIC Chair for Data Science and Machine Learning. We gratefully acknowledge the use of the facilities of Centro de Computación Científica (CCC) at UAM and thank Red Eléctrica de España for kindly supplying wind energy data. We also thank the ECMWF and Spain's AEMET for kindly granting access to the MARS repository.

References

1. Guidelines on ensemble prediction systems and forecasting. Tech. Rep. WMO-No: 1091, World Meteorological Organization (2012)
2. Alonso, Á., Torres, A., Dorransoro, J.R.: Random forests and gradient boosting for wind energy prediction. In: Onieva, E., Santos, I., Osaba, E., Quintián, H., Corchado, E. (eds.) HAIS 2015. LNCS (LNAI), vol. 9121, pp. 26–37. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19644-2_3
3. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics. Springer, New York (2006)
4. Chollet, F.: Keras: Deep learning library for theano and tensorflow (2015), <http://keras.io>
5. Díaz-Vico, D., Torres-Barrán, A., Omari, A., Dorransoro, J.R.: Deep neural networks for wind and solar energy prediction. Neural Process. Lett., April 2017
6. Fang, S., Chiang, H.D.: A high-accuracy wind power forecasting model. IEEE Trans. Power Syst. **32**(2), 1589–1590 (2017)

7. Giebel, G., Badger, J., Landberg, L., Henrik, Nielsen, A., Nielsen, T., Madsen, H., Sattler, K., Feddersen, H., Vedel, H., Tøfting, J., Kruse, L., Voulund, L.: Wind power prediction using ensembles. Tech. Rep. R-1527, Risø Risø National Laboratory Roskilde, Denmark (2005)
8. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, vol. 9, pp. 249–256, May 2010
9. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, April 2011
10. Heinermann, J.: Wind Power Prediction with Machine Learning Ensembles. Ph.D. thesis, Carl Von Ossietzky University, Oldenburg, Germany (2016)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. CoRR abs/1412.6980 (2014), <http://arxiv.org/abs/1412.6980>
12. Leutbecher, M., Palmer, T.N.: Ensemble forecasting. *J. Comput. Phys.* **227**(7), 3515–3539 (2008)
13. Lin, C.J., Weng, R.C.: Simple probabilistic predictions for support vector regression. Technical Report, Department of Computer Science, National Taiwan University (2003)
14. Mohammed, A.A., Yaqub, W., Aung, Z.: Probabilistic forecasting of solar power: an ensemble learning approach. In: Neves-Silva, R., Jain, L.C., Howlett, R.J. (eds.) *Intelligent Decision Technologies. SIST*, vol. 39, pp. 449–458. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19857-6_38
15. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
16. Ren, Y., Suganthan, P., Srikanth, N.: Ensemble methods for wind and solar power forecasting—a state-of-the-art review. *Renew. Sustain. Energy Rev.* **50**, 82–91 (2015)
17. Schölkopf, B., Smola, A.J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press (2001)
18. Sperati, S., Alessandrini, S., Delle Monache, L.: An application of the ECMWF ensemble prediction system for short-term solar power forecasting. *Sol. Energy* **133**, 437–450 (2016)
19. Woon, W.L., Kramer, O.: Enhanced SVR ensembles for wind power prediction. In: *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 2743–2748 (2016)

Author Index

- Al Dhaheri, Khawla 107
Ali, Adhra 47
Alonso, Ricardo 1
Aung, Zeyar 107
- Badr, Nagwa 33
Batzelis, Efstratios 13
Bestil, Abdullah 72
Bonilla, Edwin 94
Braun, Martin 83
Brown, Gavin 13
Buhan, Serkan 72
- Carton, Cérés 59
Catalina, Alejandro 121
- Dahl, Astrid 94
Demirci, Turan 72
Dorronsoró, José R. 121
- Er, Ceyda 72
- Fawzy, Dina 33
- Gil-Lopez, Sergio 1
Guillemé, Maël 59
Guvengir, Umut 72
- Henze, Janosch 83
- Karagoz, Pinar 72
Kazmi, Hussain 47
Kneiske, Tanja 83
Küçük, Dilek 72
Landa-Torres, Itziar 1
- Loebl, Jaroslav 26
- Manjarres, Diana 1
Masson, Véronique 59
Moussa, Sherin 33
- Nikolaou, Nikolaos 13
- Ozkan, Mehmet Baris 72
- Posch, Helmut 26
- Quiniou, René 59
- Rozé, Laurence 59
Rozinajová, Viera 26
- Secen, Ali Unver 72
Sick, Bernhard 83
- Termier, Alexandre 59
- Woon, Wei Lee 107