Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Short-term wind power forecasting: standardization, evaluation and optimization of prediction uncertainty

Juan Manuel González Sopeña

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Department of Civil, Structural and Environmental Engineering

Trinity College Dublin

21-September-2022

Declaration of authorship

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Signature:

Date: March 30th, 2022

Acknowledgments

First of all, I would like to thank my supervisors Bidisha Ghosh and Vikram Pakrashi for giving me the chance of conducting my PhD thesis with them, and supporting me unconditionally over the last three years. I would not be the researcher I became today without their guidance and support.

I would also like to thank Tirthankar Dasgupta and Han Xiao from Rutgers University for sharing their knowledge with me, and making the first lockdown more enjoyable with the countless chats about statistics we had over that time.

This work could not have been possible without the support of the Sustainable Energy Authority of Ireland. Thanks for funding the research project WindPearl. I would like to acknowledge the effort and passion of the other members of this project, Basuraj Bhowmik and Paul Mucchielli. It has been a pleasure to carry out this work side by side.

Special thanks to George Vathakkattil Joseph and Aasifa Rounak for introducing me to neuromorphic computing. The completion of this work would not have been possible without their support.

I do not want to forget Abigaïl Dah and Constant Maury, research students who joined as summer interns to WindPearl. Their enthusiasm and effort helped to push myself to become a better researcher.

Por último (que no menos importante), quería agradecer a mis padres, Juan Manuel y Ana María, a mi hermana Inma, y en especial a mi compañera de fatigas Violeta, por su apoyo y confianza durante todo este viaje. No hubiera sido posible sin vosotros.

Abstract

This thesis reports on research aimed at enhancing our understanding on wind power forecasting by investigating and proposing standards to evaluate such forecasts, quantifying the main sources of forecast error, and optimizing the currently available data and cutting-edge technologies to reduce the impact and energy consumption of the developed algorithms. These research objectives have been achieved by means of state-of-the-art statistical models, evaluated using high resolution data collected at Irish wind farms. Improved wind power forecasts play an essential role in reducing the operation & maintenance (O&M) cost of wind farms, and subsequently increasing the competitiveness of wind energy with respect to more polluting non-renewable sources of energy. The importance of this tool has lead to very active and fruitful research efforts over the years in terms of modelling. However, the impact of this research has been partially diminished due to the lack of standardization in terms of model evaluation and uncertainty quantification. Thus, measures and guidelines are to be developed to unlock the potential of wind power forecasting as a tool to reach a low-carbon future by reducing the cost of wind energy.

Firstly, lack of benchmarks for assessing the performance of wind power forecasting models has lead to a scenario where many methodologies have been applied to the modelling of wind power forecasts, but evaluated under diverse criteria with datasets of different nature and quality. Thus, our first goal is to explore the use of performance evaluation metrics to quantify model performance, and broaden their original purpose to evaluate aspects often disregarded such as the robustness of any wind power forecasting model over varied wind power generation scenarios. These concepts are later extended to propose guidelines on how to evaluate statistical wind power forecasting models.

Another key aspect often overlooked is our limited understanding on wind power forecast errors. Even if we do know the main existing sources of errors, we do not have the tools to evaluate and quantify them during the model development stage. To solve that, we have developed a simulation-based statistical framework which allow us to effectively determine such quantities.

One last contribution of this thesis stems from the fact that not only we pursue to improve the forecasting skill of our algorithms, but to use the data and the current technology in such a way that we reduce our carbon footprint. Thus, this contribution is developed following two lines of thought: first, leveraging the use of high resolution turbine-level data with the use of clustering algorithms, aiming to find a middle ground between forecasting accuracy and computational cost, and second, developing algorithms tailored for cutting-edge platforms such as the case of neuromorphic devices, a brand new technology inspired by the energy-efficient nature of the brain.

Contents

List of Figures

List of Tables

List of abbreviations

ACE	Average Coverage Error
ADMM	Alternating Direction Method of Multipliers
AI	Artificial Intelligence
AIC	Akaike Information Criterion
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
ApEn	Approximate Entropy
AR	Autoregressive
ARMA	Autoregressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
ARFIMA	Autoregressive Fractionally Integrated Moving Average
BIC	Bayesian Information Criterion
BLP	Best Linear Predictor
BM	Balancing Market
BT	Boosting Tree
CEEMD	Complete Ensemble Mode Decomposition
CEEMDAN	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
CFD	Computational Fluid Dynamics

CNN	Convolutional Neural Network
CRPS	Continuous Ranked Probability Score
CWC	Coverage Width-based Criterion
DAM	Day-Ahead Market
DBN	Deep Belief Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DE	Differential Equation
DL	Deep Learning
DT	Decision Tree
EEMD	Ensemble Empirical Mode Decomposition
ELM	Extreme Learning Machine
EMD	Empirical Mode Decomposition
ENTSO-E	European Network of Transmission System Operators for Electricity
ESN	Echo State Network
FCR	Fuzzy C-Regression
FFNN	FeedForward Neural Network
GAN	Generative Adversarial Network
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
GEFCom	Global Energy Forecasting Competition
GBM	Gradient Boosting Machine
GRU	Gated Recurrent Unit
HVDC	High Voltage Direct Current
IA	Index of Agreement
IEA	International Energy Agency

IMF	Intrinsic Mode Function
IDM	IntraDay Market
IS	Interval Sharpness
I-SEM	Integrated Single Electricity Market
KDE	Kernel Density Estimation
KL	Kullback-Leibler
kNN	k-Nearest Neighbors
LASSO	Least Absolute Shrinkage and Selection Operator
LIF	Leaky Integrate and Fire
LS-SVM	Least Squares Support-Vector Machine
LSTM	Long-Short Term Memory
LUBE	Lower Upper Bound Estimation
MA	Moving Average
MAAPE	Mean Arctangent Absolute Percentage Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MDL	Minimum Description Length
MIMO	Multiple-Input Multiple-Output
ML	Machine Learning
MLE	Maximum Likelihood Estimation
MMSE	Minimum Mean Squared Error
MSE	Mean Squared Error
NEF	Neural Engineering Framework

NMAE	Normalized Mean Absolute Error
NREL	National Renewable Energy Laboratory
NRMSE	Normalized Root Mean Square Error
NWP	Numerical Weather Prediction
O&M	Operation & Maintenance
PDF	Probability Density Function
PEI	Percentage Estimation Inflation
PI	Prediction Interval
PICP	Prediction Interval Coverage Probability
PINAW	Prediction Interval Normalized Average Width
PINC	Prediction Interval Nominal Confidence
PMI	Percentage Misspecification Inflation
PSD	Power Spectral Density
PSO	Particle Swarm Optimization.
QR	Quantile Regression
QRF	Quantile Regression Forest
ReLU	Rectified Linear Unit
RF	Random Forest
RKHS	Reproducing Kernel Hilbert Space
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SARIMA	Seasonal ARIMA
SARMA	Seasonal ARMA
SC	Skill Score

SCADA	Supervisory Control And Data Acquisition
SDE	Standard Deviation Error
SLAYER	Spike LAYer Error Reassignment
SLNA	Systematic Literature Network Analysis
SNN	Spiking Neural Network
SNN-TB	SNN conversion ToolBox
SSA	Singular Spectrum Analysis
SVM	Support-Vector Machine
SVR	Support-Vector Regression
TCN	Temporal Convolutional Network
TSA	Time Series Analysis
TPI	Total Percentage Inflation
VAR	Vector Autoregression
VMD	Variational Mode Decomposition
WNN	Wavelet Neural Network
WSF	Wind Speed Forecasting
WPD	Wavelet Packet Decomposition
WPF	Wind Power Forecasting
WRF	Weather Research and Forecasting
WT	Wavelet Transform

Chapter 1

Introduction

1.1 Background

Wind power forecasts (WPFs) are estimates of the power generation production of a wind turbine/wind farm in the foreseeable near future. These estimates are of importance in the energy industry with the increased penetration of wind power in the grid, as accurate WPFs have proven to be critical by contributing to a better decision-making process for trading in electricity markets (Pinson et al., 2007a; Skajaa et al., 2015), and by facilitating the operation of the grid (Bessa et al., 2012a; Wang et al., 2016b).

Compared with more conventional sources of electricity such as coal and gas plants, the electricity generated by wind energy systems is not easily dispatchable due to the fluctuating nature of wind speeds. Since more and more countries are gradually transitioning into renewable energy sources, such forecasts help to manage the required system flexibility to balance electricity consumption and generation, keeping the grid free from supply shortages or any other type of disruption. The crucial role of WPF to keep the stability of the grid will only rise during this decade, as the electricity system flexibility must increase by at least two-thirds even for the most modest energy transition scenarios projected (International Energy Agency, 2021).

Apart from the technical aspects in relation to the grid, WPFs are an indispensable tool for participants in electricity markets. These markets emerge to meet

the electricity consumption (namely load) profiles which are scheduled in advance in order to maintain the balance between electricity production and consumption. Deregulated electricity markets follow a similar structure almost all around the world (Conejo et al., 2010), where different trading markets are established depending on the trading horizon, such as the Day-Ahead Market (DAM) and the Intraday Market (IDM), in which the electricity is traded before the market closes, and the Balancing Market (BM), which occurs after trading. Thus, having anticipated information of energy production in the form of forecasts contribute to a more informed decision-making process in the different electricity markets.

WPF is especially significant in countries with a vast wind resource such as Ireland (Troen and Lundtang Petersen, 1989; Frank and Landberg, 1997), where the Government launched a Climate Action Plan in June 2019 and set a target of 70% of electricity coming from renewable energy sources by 2030 (EirGrid, 2019), with a strong emphasis on wind energy. Such level of penetration of renewable energy in the grid will require the estimation of WPFs in a timely and accurate manner to equip the grid with enough system flexibility to maintain its stability. Furthermore, the Irish scenario is unique in terms of geographical location, as the electrical grid is only connected to Great Britain by the HVDC (high voltage, direct current) Moyle Interconnector through Northern Ireland and the East-West Interconnector through the Republic of Ireland (differing from the high interconnection in Continental Europe), and the increasing installed capacity of wind power over the last few years (EirGrid, 2022), as shown in Figure 1.1. Furthermore, a brand new electricity market arrangement set in both Ireland and Northern Ireland, known as I-SEM (Integrated Single Electricity Market), was established in 2018 to harmonise the existing markets in the whole island and further integrate electricity markets across the European Union (Gaffney et al., 2019).

Therefore, improving the current understanding on WPF is beneficial at a worldwide level to satisfy the system flexibility required by the projected energy transition scenarios over this decade, in which renewable energy sources will play a more important role in terms of electricity generation, as well as at a regional level (e.g. Ireland) to develop tailored data-driven algorithms to forecast wind power in Irish

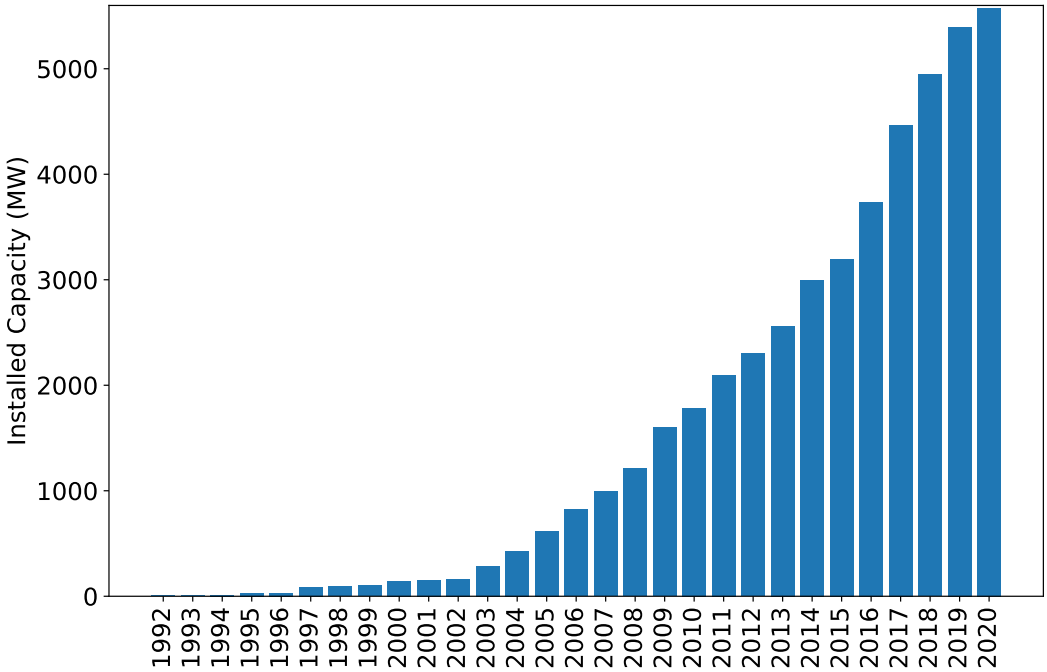

Figure 1.1: Installed wind capacity in Ireland (All-island). Reference: EirGrid (2022).

wind farms in order to satisfactorily reach the sustainability goals set by the Irish government and benefit participants who take part in the I-SEM.

1.2 Research objectives

A vast amount of research in relation to WPF modelling has been established over the years to improve the accuracy of such forecasts. However, lack of standardization in terms of benchmarking and evaluating those models may hinder the true potential of WPFs. Taking this into account, the aim of this thesis is to analyze WPF modelling in terms of applicability, limitations, and prediction possibilities, and subsequently improve the capabilities of WPF models to precisely and accurately predict wind power with computational parsimony. The main research objectives of this thesis are:

- Development of a benchmarking framework to assess the performance of statistical wind power forecasting models. Guidelines are to be developed to standardize criteria such as data requirements, time resolution, and prediction horizon, while simultaneously models are evaluated using varied representative operational conditions of wind farms. This framework is applied to benchmark

statistical models using real data from Irish wind farms.

- Systematic review of performance evaluation metrics used to evaluate statistical wind power forecasting models, and extend their application to analyze the often disregarded robustness of such forecasts.

- Development of a framework to decompose the forecast error from time series data into several components to quantify which are the main sources of error. This framework is to be applied to wind energy forecasting in particular, and any engineering system in general.

- Leverage high-resolution data available from supervisory control and data acquisition (SCADA) systems and collected individually for every turbine to develop more accurate forecasting models, thus increasing the forecasting performance of the overall wind farm.

- Development of wind power forecasting models aimed for neuromorphic computing such as spiking neural network (SNN) models. Neuromorphic computing is a new computational paradigm inspired by the energy-efficient processing of information of the brain to build low-energy and low-latency algorithms, in contrast to the state-of-the-art machine/deep learning algorithms. Thus, these algorithms have the potential to reduce the computational resources needed for building WPF models.

1.3 Thesis organization

This thesis is organized as follows (Figure 1.2). Firstly, section 1.4 presents the literature review. The state-of-the-art wind power forecasting models proposed in the literature are presented along with relevant aspects such as the algorithms applied to model WPFs.

One aspect that has barely been explored quantitatively are the main sources of forecast error, even if this understanding is valuable for the model selection stage. To fill this gap in knowledge, chapter 2 presents a statistical framework to quantify the main sources of forecast error.

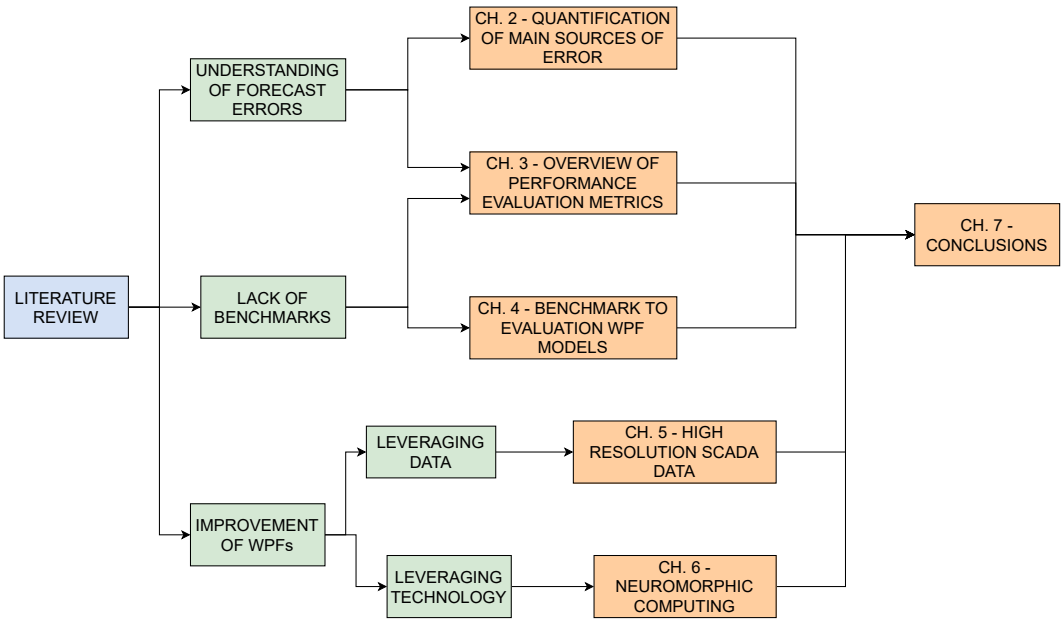

Figure 1.2: Organization of the thesis.

In order to provide a more unified vision of performance evaluation metrics and their meaning in the context of WPF, chapter 3 overviews the main performance evaluation metrics for both point and probabilistic estimates of WPFs.

Chapter 4 addresses the lack of existing benchmarks with standardized criteria to evaluate WPF models. A benchmarking framework is developed with the aim of unifying all these diverse criteria, taking into consideration the main guidelines pointed out by the International Energy Agency (IEA) Wind Task 36.

Chapter 5 provides tools to exploit the new types of data collected in wind farms. In particular, high resolution wind power measurements collected at a turbine level have been used to build specific turbine-level models. In order to leverage these data while keeping a reasonable computational cost, the use of clustering algorithms is explored to group turbines in a wind farm to build cluster-level wind power forecasting models.

New computational devices such as neuromorphic hardware can reduce the high computational cost of machine/deep learning models. Chapter 6 presents a methodology to build SNN models for short-term WPF, whose behavior mimics more closely the nature of the brain, being better suited to be implemented on neuromorphic hardware.

Finally, chapter 7 provides the conclusions of this work, summarizing and provid-

ing a critical assessment of the main contributions of this thesis. Potential directions for future research are addressed as well.

1.4 Literature review

An ample amount of research on WPF has been carried out to this day. Table 1.1 provides a list of review papers which describes in detail the key areas of research for WPF. The progressive development and publication of wind speed forecasting (WSF) and WPF approaches has required a recurrent update of the state-of-the-art wind energy forecasting, reflected in the continuous (yet necessary) reviews published over the years such as Foley et al. (2012), Jung and Broadwater (2014), Giebel and Kariniotakis (2017), or Bazionis and Georgilakis (2021). Some studies also extend their insights to solar energy (Tawn and Browell, 2022), and renewable energy forecasting in general (Sweeney et al., 2020). Furthermore, some systematic literature reviews are presented by Vargas et al. (2019), using an approach called Systematic Literature Network Analysis (SNLA), and by Maldonado-Correa et al. (2021), using the same three-step approach (planning, conducting, reporting) followed by Torres-Carrión et al. (2018).

One of the key areas of research is the uncertainty around WPFs, since this additional information is useful for the decision making-process in energy markets (Pinson et al., 2007a; Conejo et al., 2010) and maintaining the stability of the grid (Soroudi and Amraee, 2013; Bessa et al., 2014). Influential factors with respect to forecasting uncertainty are addressed by Yan et al. (2015b), such as the uncertainty on numerical weather prediction (NWP) data, the power curve, and the forecasting algorithms themselves. Zhang et al. (2014b) address the impact of WPFs on electricity prices and presents optimal bidding strategies for energy markets. In both reviews the authors analyze the existing categorizations of WPF uncertainty: a) probabilistic forecasting (Gneiting and Katzfuss, 2014), where the uncertainty is represented in the form of probability density function (PDF), quantiles, or intervals; b) risk index (Pinson and Kariniotakis, 2004), providing a real value or code to quantify the level of forecasting error, and c) scenario forecasting (Pinson et al., 2009). Additionally, an overview of computational intelligence methods for uncertainty quantification of

Table 1.1: Main review papers on wind power forecasting.

Publication	Nº refs	Review focus
Landberg et al. (2003)	26	Identification of set-up and models for short-term WPF.
Costa et al. (2008)	88	History of short-term WPF.
Lei et al. (2009)	37	Research on WSF and WPF.
Soman et al. (2010)	49	Classification of models with respect to time horizon.
Giebel et al. (2011)	381	State-of-the-art WPF. Deliverable for the ANEMOS project.
Colak et al. (2012)	60	Data mining techniques applied to WPF.
Foley et al. (2012)	68	Physical and statistical models applied to WPF.
Jung and Broadwater (2014)	149	Research on WSF and WPF.
Tascikaraoglu and Uzunoglu (2014)	112	Combined forecasting approaches for short-term WSF/WPF.
Zhang et al. (2014b)	116	Probabilistic methodologies for WPF.
Gallego-Castillo et al. (2015)	69	Wind power ramp forecasting.
Ren et al. (2015)	55	Ensemble methods for wind and solar energy forecasting.
Yan et al. (2015b)	76	Evaluation of uncertainty analysis of WPF.
Okumus and Dinler (2016)	87	Classification of statistical WPF models.
Wang et al. (2016a)	88	Analysis of multi-step ahead strategies applied to wind forecasting.
Giebel and Kariniotakis (2017)	215	Chapter reviewing the state-of-the-art WPF.
Marugán et al. (2018)	189	Application of ANNs to wind energy, including WSF/WPF.
Zendehboudi et al. (2018)	92	SVM-based approaches for wind and solar energy forecasting.
Bokde et al. (2019)	132	EMD-based models for wind forecasting applications.
Liu and Chen (2019)	128	Data processing strategies for wind energy forecasting.
Liu et al. (2019a)	153	Intelligent predictors and auxiliary methods for WSF/WPF.
Qian et al. (2019)	133	Decomposition-based hybrid models.
Quan et al. (2019)	127	Computational intelligence methods for uncertainty quantification.
Vargas et al. (2019)	223	Use of a scientific framework called SLNA.
Yousuf et al. (2019)	114	Deterministic models from a accuracy point of view.
Hanifi et al. (2020)	54	Research on WPF modelling.
Liu et al. (2020)	106	Multi-objective optimization strategies.
Nazir et al. (2020)	123	Wind forecasting methods based on ANNs.
Roungkvist and Enevoldsen (2020)	56	Time scale classifications proposed for wind forecasting.
Santhosh et al. (2020)	102	Research on WSF/WPF modelling.
Sweeney et al. (2020)	102	State-of-the-art renewable energy forecasting.
Wang et al. (2020e)	419	AI-based algorithms for wind farms.
Alkhayat and Mehmood (2021)	135	DL-based methods for wind and solar energy forecasting.
Bazionis and Georgilakis (2021)	98	Deterministic and probabilistic WPF models.
González Sopeña et al. (2021a)	92	Metrics for short-term statistical WPF.
Lipu et al. (2021)	140	AI-based hybrid methodologies.
Lu et al. (2021)	247	Meta-heuristic algorithms.
Maldonado-Correa et al. (2021)	49	Literature review linking research questions with research articles.
Tian (2021)	118	Deterministic forecasting of wind power.
Wang et al. (2021b)	240	DNN-based methodologies for WSF/WPF.
Yang et al. (2021a)	162	Classification and methodologies for WSF/WPF.
Tawn and Browell (2022)	131	Very short-term wind and solar energy forecasting.

wind power is provided by Quan et al. (2019).

WPF modelling is another key area of research. Methodologies such as artificial neural networks (ANNs) (Marugán et al., 2018), deep neural networks (DNNs) (Wang et al., 2021b), and support-vector machines (SVMs) (Zendehboudi et al., 2018) have been reviewed in the literature. More generally, artificial intelligence (AI) based models are reviewed by Wang et al. (2020e) and by Lipu et al. (2021), and forecasting models based on deep learning (DL) strategies are described by Alkhayat and Mehmood (2021). Combination of models and ensemble strategies are described by Tascikaraoglu and Uzunoglu (2014) and by Ren et al. (2015) respectively. An overview of performance evaluation metrics for all kind of statistical methodologies is found in González Sopeña et al. (2021a). Alternatively, WPF models can be classified according to the prediction horizon they are built on (Soman et al., 2010; Roungkvist and Enevoldsen, 2020). To provide the readers a context of the work carried out in this thesis, a description of the taxonomy of state-of-the-art WPF models is presented in section 1.4.1.

Another topic is the use of data processing strategies for WPF. Specifically, Liu and Chen (2019) present a review of data processing strategies for wind energy forecasting. Seven categories are identified: data decomposition, feature selection, feature extraction, denoising, residual error modelling, outlier detection, and filter-based correction. Feature selection using meta-heuristic algorithms is reviewed by Lu et al. (2021), and multi-objective optimization strategies by Liu et al. (2020). Regarding data decomposition, empirical mode decomposition (EMD) based algorithms are outlined by Bokde et al. (2019), whereas a more extended description of decomposition-based hybrid models is found in Qian et al. (2019).

Other authors have investigated other areas of interest such as Gallego-Castillo et al. (2015), where an overview on wind power ramp forecasting is presented, and Wang et al. (2016a), where different multi-step ahead forecasting strategies are described.

1.4.1 Wind power forecasting modelling

An overview on WPF modelling is presented in the remainder of this subsection, outlining the main forecasting structures applied in approximately 200 publications. This overview is limited to statistical modelling of WPF, however it is necessary to stress the importance of physical modelling of WPFs, based on meteorological information and specific site conditions at a current or future wind farm, combined with the laws of physics to produce such predictions, including computational fluid dynamics (CFD) (Ye et al., 2017), and high resolution Weather Research and Forecasting (WRF) model simulations (Prósper et al., 2019). Additionally, note that the modelling of WSFs is closely related to WPF in the sense that WSFs can be converted into forecasts of wind power by using the corresponding power curve, or utilizing WSFs as an exogenous input for a WPF model. The major categorizations of WPF models and estimates are shown in Figure 1.3. Tables 1.2-1.4 show what time series analysis (TSA), AI (i.e., machine learning (ML) and DL models), and other models have been applied respectively in the reviewed literature. For the sake

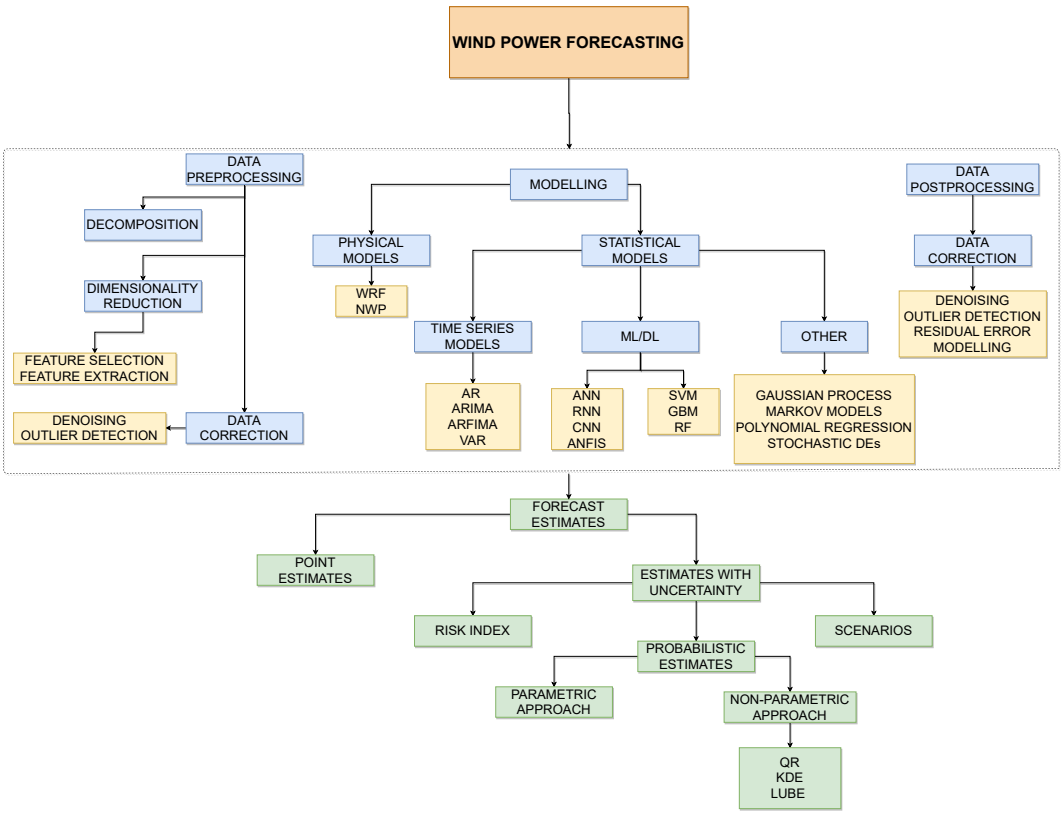

Figure 1.3: Taxonomy of WPF models.

of completeness, a chronological review of the literature is shown in Tables A.1-A.6 indicating the model(s) applied, the type of estimate provided (point, probabilistic or both), country/region where the dataset is from, and resolution of such data.

1.4.1.1 Time series analysis models

TSA is formed by a set of methodologies used to analyze time series datasets using generalized linear dynamic regression models based on statistical probability theory. In the realm of TSA, autoregressive (AR) processes are modelled using a combination of previous variables, whereas moving average (MA) processes are modelled combining previous forecast errors. They can be merged together resulting in ARMA processes, and generalized to non-stationary processes by differencing the original time series, generating the known as autoregressive integrated moving average (ARIMA) models. Compared to other forecasting models, TSA provides a well-established statistical framework that allow to draw conclusions with high confidence (Makridakis et al., 2018). A categorization of the main TSA models reviewed in this section is shown in Table 1.2.

Markov-switching AR models are applied by Pinson and Madsen (2012) and Xie et al. (2018). Two online distributed learning algorithms tailored for high-dimensional AR models are proposed by Sommer et al. (2021). A dynamical adaptive AR model for real-time WPF is presented by Zhang et al. (2021a). Vector autoregression (VAR) is a generalization of AR models that allows to predict wind power for several wind farms considering their spatio-temporal dependencies. The compu-

Table 1.2: References where time series analysis models are applied.

Model	References
VAR	Dowell and Pinson (2015); Cavalcante et al. (2017); Zhao et al. (2018); Messner and Pinson (2019)
AR	Ramirez-Rosado et al. (2009); Pinson and Madsen (2012); Karakuş et al. (2017); Xie et al. (2018); Sommer et al. (2021); Zhang et al. (2021a)
ARMA	Ziel et al. (2016); Jiang et al. (2017); Korprasertsak and Leephakpreeda (2019); Zhang et al. (2019a)
ARIMA	Liu et al. (2010)
SARIMA	Zhang et al. (2021d)
ARFIMA	Yuan et al. (2017)
GARCH	Ziel et al. (2016); Chen et al. (2019a)

tational complexity can be reduced using sparse VAR structures (Dowell and Pinson, 2015; Zhao et al., 2018), or combining such method with least absolute shrinkage and selection operator (LASSO) regularization (Cavalcante et al., 2017; Messner and Pinson, 2019). These methodologies are effective for large datasets, and have been tested using sets containing measurements between 22 and 172 wind farms.

ARMA models have also been applied in the literature. Jiang et al. (2017) apply a hybrid method where an ARMA model is used as a first step to build a base model, and combined with a boosting algorithm to improve its performance. Ziel et al. (2016) combine a time-varying ARMA model with a generalized autoregressive conditional heteroskedasticity (GARCH) based model. Zhang et al. (2019a) apply ARMA to predict the intermediate frequency components extracted after decomposing the data using variational mode decomposition (VMD). Korprasertsak and Leephakpreeda (2019) use predictions from an ARMA model combined with other models (ANN and Grey box models) to estimate WPFs. ARIMA is employed by Liu et al. (2010), autoregressive fractionally integrated moving average (ARFIMA) is used by Yuan et al. (2017) to model the linear components of wind power to characterize long-memory for wind power time series, and seasonal ARIMA (SARIMA) is applied by Zhang et al. (2021d), aiming to capture seasonality in wind data. Chen et al. (2019a) use GARCH-based models to explore the volatility existing in wind power data.

1.4.1.2 Artificial intelligence models

Most of the work on AI (namely *ML/DL models*) is focused on ANNs, as shown in Table 1.3, where the key AI models used in the literature are outlined. A basic feed-forward neural network (FFNN) is defined as a classical network with a hidden layer between the input and output layers, using the backpropagation algorithm to train the network (Rumelhart et al., 1986). Some earlier examples in the literature are for instance Jursa and Rohrig (2008), where the authors use evolutionary algorithms to select automatically the input variables and the model parameters of ANN models, or Amjady et al. (2011b), where the performance of an ANN model is enhanced with an improved version of the particle swarm optimization (PSO) algorithm. Follow-

Table 1.3: References where artificial intelligence models are applied.

Model	References
ANN	Sideratos and Hatziargyriou (2007); Jursa and Rohrig (2008); Bessa et al. (2009); Ramirez-Rosado et al. (2009); Hong et al. (2010); Kusiak and Zhang (2010); Amjady et al. (2011a,b); Blonbou (2011); Catalão et al. (2011); Bhaskar and Singh (2012); Sideratos and Hatziargyriou (2012); Khosravi and Nahavandi (2013); Khosravi et al. (2013); Lee and Baldick (2013); Peng et al. (2013); Quan et al. (2013); Shi et al. (2013); Haque et al. (2014); Buhan and Çadırcı (2015); Han et al. (2015); Kavousi-Fard et al. (2015); Li et al. (2015); Xu et al. (2015); Aghajani et al. (2016); Azimi et al. (2016); Dong et al. (2016); Renani et al. (2016); Chang et al. (2017); Liu et al. (2017); Zameer et al. (2017); He and Li (2018); Leng et al. (2018); Naik et al. (2018b); Wu et al. (2018b); Çevik et al. (2019); Hao et al. (2019); Korprasertsak and Leephakpreeda (2019); Sun et al. (2019); Wang et al. (2019d); Yan and Ouyang (2019); Zhang et al. (2019a); Abedinia et al. (2020b); Aly (2020); Chen and Liu (2020); Li et al. (2020b); Nazaré et al. (2020); Nielson et al. (2020); Ouyang et al. (2020); Shahid et al. (2020a); Sun et al. (2020a); Wang et al. (2020b); Wu et al. (2020); Zhang et al. (2020a); Dong et al. (2021b); Khazaei et al. (2022)
ELM	Wan et al. (2013a,b); Zhang et al. (2014a); Abdoos (2016); Li et al. (2016); Wan et al. (2016); Zhang et al. (2016); Zhao et al. (2016); Afshari-Igder et al. (2018); Khorramdel et al. (2018); Mahmoud et al. (2018); Hao and Tian (2019); Zhao et al. (2019); Acikgoz et al. (2020); Chen and Liu (2020); Liu and Duan (2020); Rayi et al. (2021)
DNN	Qureshi et al. (2017); Qureshi and Khan (2019); Lin and Liu (2020); Lin et al. (2020); Putz et al. (2021)
DBN	Wang et al. (2018); Zhang et al. (2019c); Duan et al. (2022)
RNN	Shi et al. (2017); Chen and Liu (2020); Wang et al. (2021a)
ESN	Zhang et al. (2016); Wang et al. (2019a,b); Hu et al. (2020a)
LSTM	Han et al. (2019a,b); Yin et al. (2019); Yu et al. (2019a); Yuan et al. (2019); Zhang et al. (2019b); Chen and Liu (2020); Devi et al. (2020); Li et al. (2020a); Shahid et al. (2020b); Sun et al. (2020b); Yu et al. (2020); Zhang et al. (2020c); Duan et al. (2021); Gu et al. (2021); Shahid et al. (2021); Wu et al. (2021); Xiang et al. (2021); Zhang et al. (2021b,c,d); Ahmad and Zhang (2022); Duan et al. (2022); Xiong et al. (2022)
GRU	Ding et al. (2019); Wang et al. (2019c); Niu et al. (2020); Hossain et al. (2021); Liu et al. (2021c); Meng et al. (2022)
CNN	Wang et al. (2017a); Hong and Rioflorido (2019); Ju et al. (2019); Yin et al. (2019); Yu et al. (2019b); Abedinia et al. (2020a); Yu et al. (2020); Hossain et al. (2021); Liu et al. (2021c); Wu et al. (2021); Yildiz et al. (2021); Zhang et al. (2021c); Xiong et al. (2022)
TCN	Meka et al. (2021); Xiang et al. (2021); Hu et al. (2022)
GAN	Yin et al. (2021); Yuan et al. (2021); Zhou et al. (2021)
Neuro-fuzzy	Catalão et al. (2010); Pousinho et al. (2011); Osório et al. (2015); Renani et al. (2016); Saleh et al. (2016); Dong et al. (2017); Sharifian et al. (2018); Çevik et al. (2019)
WNN	Bhaskar and Singh (2012); Chitsaz et al. (2015); Sun et al. (2018); Du et al. (2019); Aly (2020); Ghoushchi et al. (2021)
SVM	Kusiak and Zhang (2010); Buhan and Çadırcı (2015); Yang et al. (2015); Heinermann and Kramer (2016); Renani et al. (2016); Çevik et al. (2019); Demolli et al. (2019); Sun et al. (2019); Yan and Ouyang (2019); Li et al. (2020d,e); Lu et al. (2020b); Ouyang et al. (2020); Sun et al. (2020a); Wang et al. (2020b); He et al. (2021); Von Krannichfeldt et al. (2021)
LS-SVM	Shi et al. (2013); Yuan et al. (2015); Zhang et al. (2016); Liu et al. (2017); Safari et al. (2017); Yuan et al. (2017); Hong et al. (2019); Zhang et al. (2019a,c); Gendeel et al. (2021)
RF	Kusiak and Zhang (2010); Renani et al. (2016); Lahouar and Slama (2017); Liu et al. (2017); Demolli et al. (2019); Sun et al. (2019); Yan and Ouyang (2019); Ouyang et al. (2020); Sun et al. (2020a); De Caro et al. (2021); Von Krannichfeldt et al. (2021)
GBM	Landry et al. (2016); Nagy et al. (2016); Ju et al. (2019); Sun et al. (2019); Yan and Ouyang (2019); Sun et al. (2020a); Von Krannichfeldt et al. (2021)
DT	Heinermann and Kramer (2016); Sasser et al. (2022)
Other ML/DL	Zjavka and Mišák (2018); Mishra and Dash (2019); Wang et al. (2020a,c); Lv et al. (2021)

ing ANN models have kept aiming to improve forecast accuracy while reducing or maintaining the computational burden of the network by implementing improved optimization algorithms, using feature selection criteria to avoid the use of redundant data and avoid the propagation of errors generated by additional exogenous inputs, or using hybrid methodologies where the wind power time series is decomposed. For instance, improved optimization algorithms are applied by Chang et al. (2017), where the convergence and accuracy of the training algorithm is improved using an error feedback scheme, by Chitsaz et al. (2015), in which the clonal search algorithm contributes to capture non-linearities in the data, by Osório et al. (2015), using an optimization algorithm where the concepts of evolutionary computing and PSO are combined, and by Nazaré et al. (2020), where the Levenberg-Marquardt and PSO algorithms ared used to update the parameters of an ANN model. Feature selections methods are implemented by Zameer et al. (2017), using genetic programming to prevent the propagation of errors of the predictors, by Azimi et al. (2016), in which data are clustered into groups of similar patterns and the best one is chosen and trained by an ANN, and by Li et al. (2015), using a feature selection technique based on mutual information to select the most informative input variables for an ANN model. Furthermore, different types of data can be fed to build ANN-based WPF models, such as exclusively historical wind power measurements originally collected with SCADA systems (Li et al., 2020b), meteorological variables (Nielson et al., 2020), or a combination of both (Khazaei et al., 2022). In addition to all these modelling aspects, new computational paradigms such as neuromorphic computing benefit from biologically inspired networks such as SNNs (whose implementation for WPF is one of the research objectives of this thesis, as mentioned in Section 1.2). Wang et al. (2020c) propose a WPF model using SNNs as one of its elements. ANNs can also be built to estimate different representations of probabilistic forecasts, such as the Lower Upper Bound Estimation (LUBE) method (Khosravi and Nahavandi, 2013; Khosravi et al., 2013), quantile regression (QR), and kernel density estimation (KDE). Probabilistic estimation of WPFs is further described in section 1.4.1.5.

An alternative to the backpropagation algorithm to train ANN models faster are extreme learning machines (ELMs), an ANN-based technique based on a single-

hidden FFNN where the weights between the input and the hidden layers are randomly assigned and never updated, thus accelerating the training rate of the network (Huang et al., 2006). References such as Abdoos (2016), Hao and Tian (2019), and Rayi et al. (2021) employ ELMs as the main forecasting model. Many examples of probabilistic frameworks are found in the literature to address the uncertainty of WPFs. For instance, Wan et al. (2013a) propose the so-called hybrid intelligent algorithm, setting up a multi-objective optimization problem based on ELM and PSO. A quantum-behaved PSO algorithm is used together with ELM to construct prediction intervals (PIs) in Zhang et al. (2014a), and with a differential evolution optimization method in Mahmoud et al. (2018). Other ELM-based probabilistic methodologies are based on QR (Wan et al., 2016; Zhao et al., 2019), KDE (Khorramdel et al., 2018; Liu and Duan, 2020), and bootstrapping (Afshari-Igder et al., 2018).

Another category of ANN models are DNNs, which are formed by stacking multiple layers between the input and output layers (Bengio, 2009). In the literature, DNN models for WPF are combined with the concept of transfer learning to reduce the training time across different wind farms (Qureshi et al., 2017; Qureshi and Khan, 2019), or used to predict wind power on offshore wind turbines using data in the order of seconds (Lin and Liu, 2020; Lin et al., 2020). Putz et al. (2021) use a deep neural architecture known as N-BEATS (Oreshkin et al., 2019). Deep belief networks (DBNs), originally proposed by Hinton et al. (2006), are another type of deep neural architecture which have been applied for WPF (Wang et al., 2018; Zhang et al., 2019c; Duan et al., 2022).

Recurrent neural networks (RNNs) have the ability of modelling temporal dependencies existing in wind data. For example, Shi et al. (2017) use the LUBE method replacing a basic FFNN model with a recurrent network one. Other RNN models such as echo state networks (ESN) reduce the computational complexity of basic recurrent networks, as well as the lack of guarantee of convergence in basic RNN models (Jaeger, 2007). Some examples in the literature are Wang et al. (2019a), Wang et al. (2019b), and Hu et al. (2020a). More advanced recurrent networks can capture longer-term relationships. In particular, long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have been widely used to produce

point WPFs in the recent literature (Han et al., 2019a,b; Yu et al., 2019a; Devi et al., 2020; Shahid et al., 2020b, 2021; Duan et al., 2021; Ahmad and Zhang, 2022). Probabilistic estimates using LSTM are obtained in the literature considering the Beta distribution (Yuan et al., 2019), using the LUBE framework (Li et al., 2020a), and utilizing KDE (Gu et al., 2021). Gated recurrent units (GRUs) (Cho et al., 2014) are another type of advanced recurrent network which can be regarded as an updated version of a LSTM network with fewer parameters. GRUs have also been applied in recent times to forecast wind power (Ding et al., 2019; Niu et al., 2020; Meng et al., 2022). A method to build PIs using GRU networks after decomposing the wind power data with the VMD algorithm is found in Wang et al. (2019c).

Convolutional neural networks (CNNs) are used to extract relevant hidden patterns in data. A two-dimensional CNN is used by Abedinia et al. (2020a), and a residual-based CNN forecasting model is proposed by Yildiz et al. (2021). Deep CNNs models are also proposed in the literature (Wang et al., 2017a; Hong and Rioflorido, 2019; Yu et al., 2019b). A particular type of CNN able to capture temporal dynamics known as temporal convolutional network (TCN) is employed as the main forecasting model by Meka et al. (2021) and Hu et al. (2022).

Both recurrent and convolutional networks can be jointly used to benefit from their features simultaneously. Thus, such combination results in CNN-LSTM models (Yin et al., 2019; Yu et al., 2020; Wu et al., 2021; Zhang et al., 2021c; Xiong et al., 2022), CNN-GRU models (Hossain et al., 2021; Liu et al., 2021c), and TCN-LSTM models (Xiang et al., 2021). A CNN is combined with another ML algorithm called gradient boosting machine (GBM) to first extract data features with the convolutional networks and then pass the filtered information to the GBM in Ju et al. (2019).

Generative adversarial network (GAN) is a type of ANN designed by Goodfellow et al. (2014). It consists of two neural networks (known as *generator* and *discriminator* networks) which compete with each other to obtain optimal results. GANs are used in the WPF literature by Yin et al. (2021), Yuan et al. (2021), and Zhou et al. (2021). Autoencoders (Schmidhuber, 2015) are another type of neural architecture consisting also of two separate modules (*encoder* and *decoder*) where the input vec-

tor is encoded by the first module, and later translated into an output sequence by the second module (Zhang et al., 2020c; Wang et al., 2021a). Neuro-fuzzy systems integrate fuzzy logic rules within ANN models, such as the adaptive network-based fuzzy inference system (ANFIS) (Catalão et al., 2010; Pousinho et al., 2011; Osório et al., 2015). Other neuro-fuzzy systems in the literature are proposed by Dong et al. (2017) and Sharifian et al. (2018). Other types of ANN found in the WPF literature are wavelet neural networks (WNNs) (Bhaskar and Singh, 2012; Sun et al., 2018; Du et al., 2019; Ghoushchi et al., 2021), combination of ANN-based models (Shahid et al., 2020a; Wu et al., 2020), differential polynomial neural networks (Zjavka and Mišák, 2018), Legendre neural networks (Mishra and Dash, 2019), or Laguerre neural networks (Wang et al., 2020a).

Another type of ML models are SVMs, originally proposed by Cortes and Vapnik (1995). In short, the input vector is non-linearly mapped to a high dimensional space where a linear decision surface with special properties is built to guarantee the generalization of the network. Yang et al. (2015) use SVM to capture the dynamics of wind power ramps to improve the performance of a Markov model. In other cases, SVM is employed as the baseline forecasting model, enhancing its performance with an improved atomic search algorithm (Li et al., 2020d), an improved dragonfly algorithm (Li et al., 2020e), or the grey wolf optimizer (Lu et al., 2020b). He et al. (2021) propose a probabilistic method based on a QR-based SVM model, and subsequently enhanced with the Epanechnikov KDE to determine the complete wind power PDF. The least-squares adaption of SVM (LS-SVM) is also applied in the literature to forecast wind power (Yuan et al., 2015; Safari et al., 2017; Hong et al., 2019). LS-SVM is merged with the LUBE framework to estimate PIs of wind power in Gendeel et al. (2021). Additionally, LS-SVM is applied to predict the nonlinear component of wind power (Yuan et al., 2017), the trend component obtained with singular spectrum analysis (SSA) (Zhang et al., 2019c), and low frequency components existing in wind power data (Zhang et al., 2019a).

Random forests (RF) (Ho, 1995) and GBMs (Friedman, 2001) are both ML algorithms constituted by decision trees (DTs), but combined in a different manner. In the first case, the results of each DT are averaged at the end of the process to obtain

the final forecast, whereas the DTs are combined sequentially to get a more skilled learner when using a GBM. Lahouar and Slama (2017) propose a RF approach to estimate hour ahead WPFs, whereas GBMs are applied by Landry et al. (2016) and Nagy et al. (2016). A DT regression model using atmospheric data is implemented by Sasser et al. (2022).

Several ML approaches (such as ANN, SVM and RF) are compared under different frameworks in Kusiak and Zhang (2010), Renani et al. (2016), and Yan and Ouyang (2019). Demolli et al. (2019) use five ML algorithms to obtain WPFs from daily wind speed data. Other WPF models are based on the idea of combining forecasting models to increase forecasting accuracy (Hyndman and Athanasopoulos, 2018), or establishing mechanisms to select the best from a pool of models (Petropoulos et al., 2022). Sun et al. (2019, 2020a) employ a reinforcement learning algorithm known as Q-learning to select the best model among a set of predetermined models (ANN, SVM, GBM, and RF). Ouyang et al. (2020) propose an ensemble method combining the results of an ANN, SVM and a RF models based on a Markov-switching regime and model performance. Chen and Liu (2020) introduce a multi-learner ensemble of sixteen models constituted by a variety of ANN models such as ELM or LSTM networks. An ensemble method named DAFT-E is proposed by De Caro et al. (2021), comprised by a set of computationally inexpensive models (RF, Lazy learning, and persistence). An online ensemble method is introduced by Von Krannichfeldt et al. (2021), using a QR approach to combine the results of SVM, RF, gradient boosting regression trees, and extremely randomized trees models. In Buhan and Çadırcı (2015), ANN and SVM models are combined by means of weighted averaged combinations of these models which yield the minimum forecast error. Shi et al. (2013) propose a dynamic hybrid model combining LS-SVM and ANN. Similarly, Wang et al. (2020b) introduce a combined approach based on Bayesian model averaging and ensemble learning using ANN- and SVM-based models as baseline methods. Other combination paradigms are proposed by Zhang et al. (2016), using LS-SVM, ESN and ELM models, by Heinermann and Kramer (2016), using an heterogeneous ensemble constituted by DTs and SVM-based models, by Liu et al. (2017) combining of ANN-based models and LS-SVM using ANFIS, and by

Çevik et al. (2019), combining the results of ANFIS, ANN, and SVM models.

1.4.1.3 Other methodologies

A summary of other methodologies applied for WPF is shown in Table 1.4. One possibility is to model WPFs with Gaussian Processes, which are defined by Rasmussen (2003) as *"a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions"*, and are suitable for non-linear regression. Chen et al. (2013) use first a Gaussian Process to correct the forecast wind speed from a NWP model, and a censored Gaussian Process to map the relation between wind speed and wind power. Lee and Baldick (2013) develop an ensemble method consisting of Gaussian Processes and ANN models. Kou et al. (2013) present an adaptation of Gaussian Processes called warped Gaussian Process to model non-Gaussian uncertainty of wind power time series. A temporally local Gaussian Process is proposed by Yan et al. (2015a) to model the time-varying features existing in wind power data. A learning procedure called *teaching learning based optimization* is introduced by Yan et al. (2016) to train and accelerate the learning rate of a Gaussian Process model. Ahmadpour and Farkoush (2020) compare different Gaussian Process based models to forecast wind power at a wind farm and regional level. A sparse variational Gaussian Process is proposed by Wen et al. (2022), aiming to address issues inherent to the original Gaussian Process such as inference complexity or the determination of hyperparameters. Other authors propose WPF models based on linear regression (Ozkan and Karagoz, 2015) or kernel regression models (Wang et al., 2017b; Naik

Table 1.4: References where other methodologies are applied.

Model	References
Gaussian Process	Chen et al. (2013); Kou et al. (2013); Lee and Baldick (2013); Yan et al. (2015a); Baxevani and Lenzi (2018); Ahmadpour and Farkoush (2020); Wen et al. (2022)
Kernel regression	Wang et al. (2017b); Naik et al. (2018a,c)
Grey box model	An et al. (2012); Korprasertsak and Leephakpreeda (2019); Lu et al. (2020a); Qian and Wang (2020)
Markov model	Pinson and Madsen (2012); Carpinone et al. (2015); Yang et al. (2015); Yan et al. (2016); Xie et al. (2018)
kNN	Kusiak and Zhang (2010); Renani et al. (2016); Zhang and Wang (2016); Yesilbudak et al. (2017); Demolli et al. (2019)
Stochastic DEs	Iversen et al. (2017); Alshelahi et al. (2021)
Bernstein polynomial	Dong et al. (2021a,b)

et al., 2018a).

Other WPF modelling methodologies used in the literature are for instance grey box models (An et al., 2012; Korprasertsak and Leephakpreeda, 2019; Lu et al., 2020a; Qian and Wang, 2020), Markov models (Carpinone et al., 2015), stochastic differential equations (Iversen et al., 2017; Alshelahi et al., 2021), k-nearest neighbors (kNN) (Zhang and Wang, 2016; Yesilbudak et al., 2017), and Bernstein polynomials (Dong et al., 2021a,b).

1.4.1.4 Decomposition-based hybrid models

Data decomposition is a frequently used preprocessing step for wind energy forecasting applications (González Sopeña et al., 2021c). In particular, 57 of the reviewed papers in this thesis use some sort of preprocessing applying any decomposition algorithm (Table 1.5), but more than 100 publications for wind forecasting applications are found in the literature, indicating promising outcomes in terms of performance (Qian et al., 2019). As wind power is such complex data, affected by meteorological conditions and activities performed in a wind farm, the aim of performing such decomposition is to replace a difficult task (such as WPF is) for a set of subtasks with a lower degree of complexity (Dong et al., 2019). We provide a summary of these algorithms as we will utilize some of them in the following chapters as our base models. A more complete description of such algorithms is found in chapter 4.

Wavelet transform (WT) (Meyer, 1992; Daubechies and Bates, 1993) is a mathematical approach which allow us to extract the *approximation* and *detail* components of data. In particular, the approximation components contain the trend existing in the original data (i.e., the lower-frequency components), whereas the detail components represent the higher-frequency components.

EMD is a recursive algorithm proposed by Huang et al. (1998), where the original time series is decomposed into a set of subseries known as *modes* or *intrinsic mode functions* (IMFs), plus a residue. The complexity of these subseries is lower compared to the original one, thus allowing to build more accurate forecasting models after decomposing the signal. Over the years, adaptations of this algorithm have been proposed to mitigate issues existing in the original EMD definition, such as mode

Table 1.5: Decomposition algorithms applied to WPF.

Reference	WT	EMD	EEMD	CEEMD	CEEMDAN	VMD	SSA	Other
Catalão et al. (2010)	×							
Liu et al. (2010)	×							
Catalão et al. (2011)	×							
An et al. (2012)		×						
Bhaskar and Singh (2012)	×							
Haque et al. (2014)	×							
Zhang et al. (2014a)			×					
Li et al. (2015)	×							
Osório et al. (2015)	×							
Abdoos (2016)						×		
Aghajani et al. (2016)	×							
Azimi et al. (2016)	×							
Zhang et al. (2016)						×		
Dong et al. (2017)	×							
Safari et al. (2017)			×				×	
Wang et al. (2017a)	×							
Afshari-Igder et al. (2018)	×							
Leng et al. (2018)	×							
Naik et al. (2018a)						×		
Naik et al. (2018b)						×		
Naik et al. (2018c)		×						
Çevik et al. (2019)	×							
Du et al. (2019)					×			
Han et al. (2019a)						×		
Han et al. (2019b)						×		
Hao and Tian (2019)						×		
Hong et al. (2019)								×
Qian and Wang (2020)								×
Wang et al. (2019a)	×							
Wang et al. (2019c)						×		
Zhang et al. (2019a)						×		
Zhang et al. (2019c)							×	
Yin et al. (2019)		×				×		
Abedinia et al. (2020a)	×							
Abedinia et al. (2020b)		×						
Chen and Liu (2020)				×				
Devi et al. (2020)			×					
Li et al. (2020d)	×							
Liu and Duan (2020)	×							
Lu et al. (2020a)					×			
Sun et al. (2020b)	×							
Wang et al. (2020a)							×	
Wu et al. (2020)	×	×				×		
Dong et al. (2021a)				×				
Dong et al. (2021b)	×							
Duan et al. (2021)						×		
Gendeel et al. (2021)						×		
Liu et al. (2021c)		×						
Rayi et al. (2021)						×		
Wang et al. (2021a)						×		
Yildiz et al. (2021)						×		
Yin et al. (2021)			×					
Zhang et al. (2021d)	×							
Zhou et al. (2021)						×		
Duan et al. (2022)						×		
Khazaei et al. (2022)	×							
Meng et al. (2022)			×					

mixing (Yang et al., 2009). One of such adaptations is called ensemble empirical mode decomposition (EEMD) (Wu and Huang, 2009), where mode mixing is fixed by setting multiple trials using the EMD algorithm and mixing the original data with Gaussian white noise. Other adjustments to the original version of EMD are the complete ensemble empirical mode decomposition (CEEMD) algorithm (Yeh et al., 2010), where a collection containing a combination of Gaussian white noise and its complementary are used instead, and the complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) algorithm (Torres et al., 2011), where the algorithm is further adjusted to reduce the number of trials required in comparison to the EEMD and CEEMD algorithms.

VMD (Dragomiretskiy and Zosso, 2013) is another decomposition algorithm that extracts modes or IMFs following the same definition as EMD-based algorithms. Compared to EMD, the modes are extracted concurrently together with their center frequencies, by solving a variational problem using an algorithm called alternating direction method of multipliers (ADMM) (Hestenes, 1969; Boyd et al., 2011).

SSA is a non-parametric spectral extraction approach used to decompose data into a trend, periodic/quasi-periodic, and noise components (Elsner and Tsonis, 1996). First, data are decomposed by means of a two-stage process. Secondly, in the reconstruction stage, those components which are similar are grouped, and later mapped to form the reconstructed time series.

Two of the references use different decomposition algorithms rather than the ones mentioned above. Hong et al. (2019) use an algorithm called morphological high-frequency filter, which separates the data into a trend and a high frequency component, and Qian and Wang (2020) apply the Hodrick-Prescott filter to identify the trend and cyclic component existing in the data.

1.4.1.5 Probabilistic estimates of WPFs

Traditionally, WPF models have provided outputs as point (or deterministic) estimates, meaning that a single value is computed for every time step to be forecast. Nonetheless, every prediction is associated with a certain degree of uncertainty which is impossible to reduce entirely. Probabilistic forecasts overcome this issue and allow

us to obtain a probabilistic estimate for future wind power outputs. Several representations for probabilistic estimates are found in the literature: predictive densities which represent the probability distribution of future outputs, quantiles that divide the probability distribution into intervals, and PIs which provide the range where a value will be located under a given distribution. The latter representation tends to be more appealing for end users, so usually PIs are preferred as probabilistic estimates. Table 1.6 indicates what references use some sort of methodology to estimate probabilistic representations of WPFs. Further insights on the performance of probabilistic estimates are found in chapter 3.

A non-parametric approach known as LUBE method (Khosravi and Nahavandi, 2013; Khosravi et al., 2013) consists of an ANN model with two outputs which represent the upper and lower boundaries of the constructed PI. A multi-objective optimization problem is set as the loss function, based on the two main properties of the PI: coverage and width. Quan et al. (2013) convert the original multi-objective optimization problem into a constrained single-objective one. Kavousi-Fard et al. (2015) add a fuzzy-based loss function to the LUBE method to facilitate the adjustment of the neural network parameters. Wu et al. (2018b) use the LUBE method with NWP data combined with the charged system search algorithm to adjust the parameters of the ANN. Pinson and Tastu (2014) argued the validity of this method to build optimal intervals. An answer from the authors of the LUBE framework was

Table 1.6: Main methods for probabilistic estimations of WPFs.

Model	References
QR	Nielsen et al. (2006); Haque et al. (2014); Gallego-Castillo et al. (2016); Wan et al. (2016); Zhang et al. (2016); He and Li (2018); Zhao et al. (2019); Hu et al. (2020b); Yu et al. (2020); He et al. (2021); Hu et al. (2022)
KDE	Bessa et al. (2012b,c); Zhang and Wang (2016); He and Li (2018); Khorramdel et al. (2018); Lin et al. (2018b); Liu and Duan (2020); Gu et al. (2021); He et al. (2021); Lv et al. (2021); Dong et al. (2022)
LUBE	Khosravi and Nahavandi (2013); Khosravi et al. (2013); Quan et al. (2013); Kavousi-Fard et al. (2015); Shi et al. (2017); Wu et al. (2018b); Zou et al. (2019); Li et al. (2020a); Wang et al. (2020c); Gendeel et al. (2021)
GMM	Sun et al. (2019); Zhang et al. (2019b); Dong et al. (2021a)
Other	Pinson and Kariniotakis (2010); Alessandrini et al. (2015); Lin et al. (2018b)

provided in Khosravi and Nahavandi (2014), indicating solutions to improve the reliability of this method, such as modifying the original multi-objective optimization problem into a constrained single-objective problem (already applied by Quan et al. (2013)), and modifying the original score skill to evaluate intervals in the case of zero-width intervals.

Another alternative to build probabilistic estimates of WPFs is QR, a non-parametric method where the uncertainty is estimated by means of a set of forecast quantiles. QR is characterized by its free-distribution approach and its flexibility to include predictors (Bremnes, 2004). Nielsen et al. (2006) use the forecast errors from an existing system as a starting point to build the 25% and 75% quantiles of the forecast errors with linear QR. This methodology can be combined easily with ANN models to extend points estimates to PIs, such as Haque et al. (2014), where QR is used to establish probabilistic estimates from an ANN-based methodology. Other authors, such as Gallego-Castillo et al. (2016) and Hu et al. (2020b) formulate their QR-based approaches within the reproducing kernel Hilbert space (RKHS) framework (Berlinet and Thomas-Agnan, 2011). Additionally, a QR based neural network is combined with another non-parametric approach known as KDE in He and Li (2018), where the predictions at different quantiles are used as an input for the KDE to model the PDF. Other probabilistic methods using ANNs is provided by Sideratos and Hatziargyriou (2012), where eight different quantiles of the future wind power distribution are obtained to model the uncertainty of the WPF.

KDE is another method applied in the literature, using time-adaptive frameworks (Bessa et al., 2012b,c), or combined with other methodologies such as Bayesian learning (Lin et al., 2018b). Gaussian mixture models (GMM) represent another probabilistic model which consists of a certain number of Gaussian distributions, which are later aggregated to estimate the PDF of the variable of study. Sun et al. (2019) use GMM to model the marginal distribution of WPFs, Zhang et al. (2019b) examine the error distribution of WPFs for short-term prediction horizons with a GMM, and Dong et al. (2021a) use this model to fit noise distribution in order to improve the generalization of a baseline forecasting model based on Bernstein polynomials. Other probabilistic methodologies are adapted resampling (Pinson and Kariniotakis, 2010),

an empirical nonparametric approach to produce conditional interval forecasts using recent forecast errors, a fuzzy inference model, and a resampling scheme to combine probability distribitutions; and analog ensemble (Alessandrini et al., 2015), a method that provides a set of WPFs based on historical data and deterministic model predictions.

1.4.2 Datasets for wind power forecasting

All the reviewed references are benchmarked with datasets collected from different parts of the world (Figure 1.4). In addition, some authors have reported to utilize data from several regions of Europe, the Iberian Peninsula, and data from the Global Energy Forecasting Competition 2012 (GEFCom2012) (Hong et al., 2014) and GEFCom2014 (Hong et al., 2016), whereas no reference to the region/country of origin was found in 12 of the reviewed publications. Usually, WPFs are modelled using historical data from the wind farm, NWP data, or both. In some cases, these datasets are not publicly available as the providers are unwilling to share these data, even if privacy is ensured (Gonçalves et al., 2020).

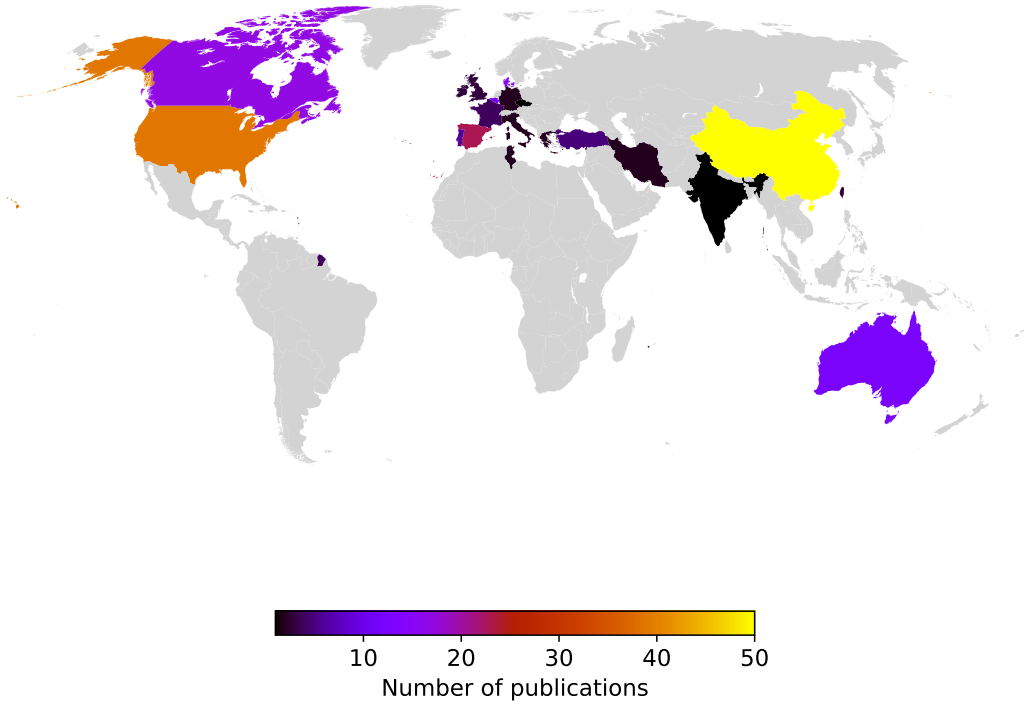

Figure 1.4: Country of origin of the datasets used in the reviewed publications.

The dataset of the GEFCom2012 is originally constituted by hourly resolution historical wind power measurements of seven wind farms and meteorological forecasts of the zonal and meridional components of surface winds at 10 m above ground level. This dataset has been used in references such as Gallego-Castillo et al. (2016) and Karakuş et al. (2017). On the other hand, GEFCom2014 data are obtained from 10 wind farms located in Australia. Hourly wind power measurements are available, as well as meteorological forecasts at 10 and 100 m above ground level. These data are applied to benchmark some models such as the ones proposed by Lin et al. (2018b) and Wen et al. (2022).

Some wind farms have data available online. For instance, Sotavento wind farm provides several types of data, including wind power, speed, and direction at a 10-minute resolution (Sotavento Wind Farm, 2022). Belgian wind power data are also publicly available at 15-min intervals (ELIA, 2022). The National Renewable Energy Laboratory (NREL) also make public their datasets, such as the wind integration national dataset (WIND) toolkit (Draxl et al., 2015). A complete categorization of open-source wind and wind power datasets is provided by Effenberger and Ludwig (2022).

1.5 Scope of the work

A number of research gaps in the literature have been identified after conducting the literature review.

Comparing the different WPF models proposed in the literature is a challenging task as they are tested under different conditions. The inputs given to the model differ, as univariate time series can be considered (i.e., only wind power data) or multivariate time series leveraging the dependency on other variables such as wind speed or wind direction. The input dataset can be distinguished in terms of its scale, as the model could be fitted at either a turbine, farm or regional/national production level, and in terms of sample size, as it varies from a few months to approximately three years of data to benchmark the model, and in terms of time resolution, usually from 10 minutes to 1 hour, as intra-hour resolution data show higher volatility (Sorensen et al., 2007; Liu et al., 2019b). While forecasting competitions represent a

good opportunity to compare different forecasting models (Hong et al., 2014, 2016; Hyndman, 2020), they may not consider all the elements deemed necessary to make such a comparison fair and representative. Therefore, this gap in the literature is addressed in chapter 3, where an overview of performance evaluation metrics for statistical WPF models is presented, and in chapter 4, where a novel benchmarking framework to evaluate WPF models is introduced.

Furthermore, we do not have a complete picture of what originates the errors of WPFs. Even if it is widely known that the main components of uncertainty are the model (*epistemic uncertainty*) and the data (*aleatoric uncertainty*), we do not have tools to quantify their contributions as such. The statistical framework presented in chapter 2 aims to establish a foundation to clarify such questions.

Exceptionally, in the most recent literature a high number of studies have used ML/DL based models to forecast wind power, as shown in Table 1.3. However, the training of more complex neural models is tied to a high computational cost, hampering any possible implementation for real- or near real-time applications. We approach this issue in two different manners. Chapter 5 explores the possibilities of leveraging high resolution SCADA data combined with clustering algorithms to find a middle ground between improved WPFs and the total computational cost. On the other hand, chapter 6 investigates the application of SNNs for short-term WPF, as the potential of these algorithms can be maximized with neuromorphic computing, a brand new computational technology which aims to reduce the computational cost of ML/DL by developing devices mimicking the energy-efficient behavior of the brain as source of inspiration.

Chapter 2

Sources and decomposition of errors

The correct modelling of errors is critical for real-time applications in the energy industry such as the case of wind power forecasting. However, errors from wind energy applications are often not clearly defined, and some variables are disregarded when forecast errors are modelled. Therefore, this issue hinders the applicability and validity of wind power forecasts, in particular during the model development stage as forecast errors are key performance evaluators of wind power forecasting models. Therefore, knowing and quantifying the main sources of forecast error are necessary to further establish comparison benchmarks and markers of error.

2.1 Statistical error modelling

Statistical error modelling is a process that involves the development of relationships between measurements, the required measurements, and the errors related to the measurements, where the error is defined as the difference between the value of a certain quantity and the best estimate of the value of such a quantity (Cox and Harris, 2004). That is, statistical models are assigned to the deviations associated with the measurements. This error can result from: a) random effects only and b) a combination of random and systematic effects.

Systematic effects can be reduced by increasing the number of measurements.

For simplicity, if we consider a linear regression problem:

$$v_i = \sum_{j=1}^{m} a_j h_j(t_i) + e_i, \quad i = 1, \ldots, m \tag{2.1}$$

where v_i is a measurement corresponding to a stimulus t_i, h_j basis functions defining a model relating the measurements and the stimuli, a_j are the model parameters, and e_i the measurement deviations. Besides, let us define $e = (e_1, \ldots, e_m)^T$ as the vector containing all the measurement deviations. If the error only comes from random effects, then

$$e = e_r \tag{2.2}$$

where e_r is a sample of random variables E_r, and the estimate of E_r is a zero vector with an associated covariance (uncertainty) matrix V_r. If the measurement deviations come from both random and systematic effects, then

$$e = e_r + b \tag{2.3}$$

where b is defined as a sample of random variables B, where E_r and B are independent, and the estimate of B is the zero vector with an associated covariance matrix V_b.

The parameters a_j can be estimated one way or another depending on what effects

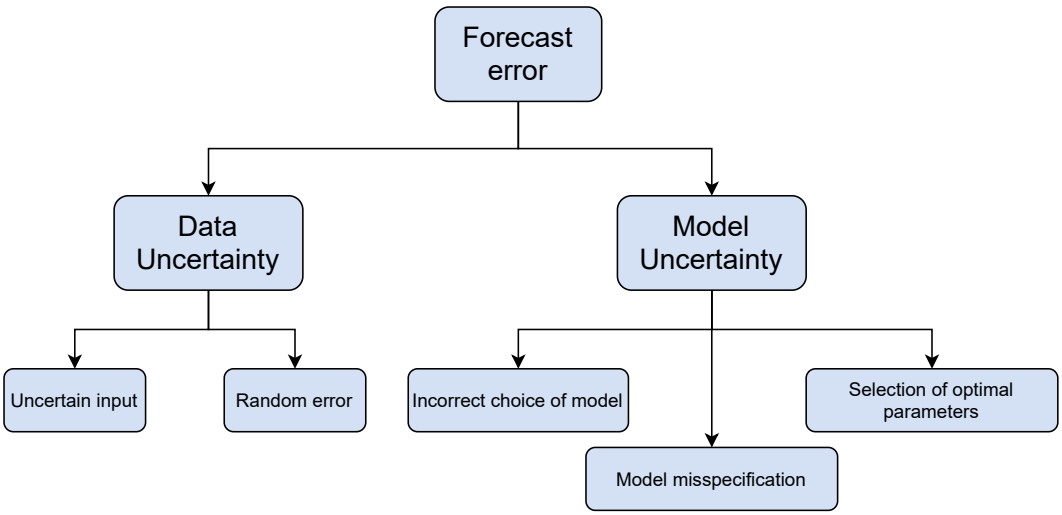

Figure 2.1: Main sources of prediction uncertainty.

apply for the measurement deviations (random and/or systematic) and whether the random effects are quantified or not.

Taking this theoretical background into consideration, two main sources of error can be identified (Figure 2.1): the data (also known as *aleatoric uncertainty*) and the forecasting model itself (namely *epistemic uncertainty*). Such concepts have been an important topic of discussion in the past (Faber, 2005), and such identification is useful to reduce the error without needing to resort to great advances in terms of modelling (Der Kiureghian and Ditlevsen, 2009).

2.1.1 Data uncertainty

In the case of data uncertainty, time series are usually addressed as deterministic systems where every time step is conformed by a single data point (Petropoulos et al., 2018). However, this assumption neglects the multiple realizations of the time series, as these values might have been estimated by sampling different measurements in a given time interval. This is the case of wind speed or wind power measurements obtained with SCADA systems, which can be collected in the order of seconds (Gonzalez et al., 2019). Despite that, these data are resampled depending on the application. For instance, applications such as the operation of reserves or trading in the electricity markets usually operate in intervals of 10 (Song et al., 2014) or 15 minutes (Mazzi et al., 2015).

2.1.2 Model uncertainty

The model chosen to estimate the WSF/WPF might not be the most adequate, or that model might not even exist (in the words of the statistician George Box, "*all models are wrong*" (Box, 1976)). Even in the case we can define one model as the best existing one, we might not be able to estimate the optimal parameters of that model. The error stemming from these limitations is defined as model uncertainty (Wit et al., 2012). In the remainder of this chapter we will focus on the error stemming from the model, assuming that this error is exclusively coming from the model using data generated synthetically (González Sopeña et al., 2022b), so only one realization of the time series actually exists for every time step.

One of the sources of forecast error is model misspecification (Chatfield, 1996). For example, suppose the wind data are actually generated by an ARIMA model of the order (p, d, q) (where p is the order of the AR term, d the degree of differencing, and q the order of the MA term), and is misspecified as an ARIMA(p^*, d^*, q^*) model. Although there are various tools available for model specification or identification that identify the true model almost surely from an asymptotic perspective, it is not uncommon for an analyst to identify an incorrect model from a sample dataset of small size. Pukkila et al. (1990) suggested a method for determining the order of an ARIMA$(p, 0, q)$ or ARMA(p, q) model, that was shown to perform well with samples of size 100 or more for $p + q \leq 3$. However, for samples of size 50, the performance was not as good - e.g., an ARMA(1,2) model with the AR parameter $\phi_1 = 0.60$ and the MA parameters $\theta_1 = -0.50$ and $\theta_2 = -0.90$ was correctly identified only 34% of the time from samples of size 50. On the remaining occasions, the model was incorrectly identified as ARMA(2,0), ARMA(0,2), ARMA(1,0) and ARMA(3,0). An interesting question that immediately arises is, how much does the prediction suffer if one of these incorrect models is used.

It has sometimes been argued that incorrect model specification is sometimes a consequence of a nearly equivalent mathematical representation of the true model. For example, Kendall (1971) had argued that the time and effort spent in identifying the correct order of ARMA models can be saved by fitting moderately long autoregressive models. However, many authors, including Box and Jenkins (1973), have provided a counter-argument that such non-parsimonious models result in noisy forecasts due to uncertainty involved in estimation of a large number of model parameters. Thus, even if two models have almost similar mathematical representations, resulting in nearly equal forecast error, if the misspecified model is less parsimonious, the forecast error is likely to be inflated due to the uncertainty associated with estimation of model parameters. This aspect is often ignored in modern data science (including WSF/WPF), where overfitting is considered benign by many.

This brings us to the question we want to investigate: if one makes a forecast using a time series model M, assumed different from the true data generating model, what are the sources of the forecast error and how is the forecast error distributed

among these sources? Such a decomposition is useful in many real-life scenarios such as forecasting for wind energy applications. In particular, studies have modelled wind forecasts using ARMA (Torres et al., 2005; Erdem and Shi, 2011), ARIMA (Su et al., 2014; Grigonytė and Butkevičiūtė, 2016; Cadenas et al., 2016), and SARIMA models (Liu et al., 2021b; Zhang et al., 2021d). Furthermore, ARIMA models have been applied to wind energy forecasting combined with other type of models to estimate a final forecast (Shi et al., 2012; Wang and Hu, 2015), and to model low and high frequencies found in wind speed time series (Yunus et al., 2015; Liu et al., 2021a). However, the proposed models typically differ with respect to their autoregressive and moving average orders. Assuming that one of these models is true and the inherent error is the same, it is of practical interest to see how much the forecast error will be inflated if other models are chosen, and to which sources these errors will be attributed. These insights can be of relevance for other fields such as traffic (Ghosh et al., 2009), ecology (Mac Nally et al., 2018) or dynamical systems in general (Mangan et al., 2017), where there is an aim to understand how to better select models.

Thus, the proposed framework would (a) allow the decomposition of forecast error (measured by the mean squared error or MSE) from an arbitrary time series model into three components, and (b) outline a simulation procedure that would enable a researcher to decompose the forecast error of a proposed model, and compare it with that of a *benchmark model*. Surprisingly, in spite of the existence of a fairly large literature on time series model misspecification, a decomposition of forecast error like the one described above appears to have been scantily addressed and discussed. Our proposed framework is based on ideas similar to that developed by Davies and Newbold (1980), who studied how forecast errors are inflated if misspecified ARIMA models are used in lieu of true data generating models. The main contributions of this Chapter include: (i) providing a decomposition formula that helps to quantify the percentage contributions of the sources of forecast error, (ii) providing a new definition of "optimal parameters of a misspecified model" used in (i), and (iii) developing a simulation algorithm that can be used to estimate the percentage contributions of each source in the decomposition. We also discuss the

application of the proposed approach pertaining wind speed forecasting.

In the following section we introduce some notation, briefly describe the basics of forecasting with ARMA models and introduce the notion of parameter uncertainty as a component of forecast error. In section 2.3, we explore the effect of model misspecification on forecast error, introduce the notion of "optimal misspecified model", and propose measures of inflation of the forecast error arising from parameter and model uncertainty. In section 2.4, we lay out a comprehensive simulation framework to assess the source-wise inflation of error of forecast made from a misspecified model. In section 2.5, we describe the application of the proposed simulation framework tailored for WSF. Some concluding remarks are presented in section 2.6.

2.2 Contribution of parameter uncertainty in forecast error

The notation introduced here can be found in most common and well-known time series textbooks like Box and Jenkins (1970) and Brockwell and Davis (2002). Suppose we have observed $Y_1, \ldots, Y_n$, i.e., n data points from a time series generated from a "true" model $\mathcal{M}$. Under a specified model M, the best or minimum mean squared error (MMSE) h-steps ahead forecast, i.e., forecast of Y_{n+h} from n data points under model M is given by

$$\hat{Y}^{(M)}_{n+h|n} = E_M(Y_{n+h}|Y_1, \ldots, Y_n), \tag{2.4}$$

where $E_M(\cdot)$ denotes expectation under model M, assuming the true model parameters are known. The forecast error is $e^{(M)}_{n+h|n} = \hat{Y}^{(M)}_{n+h|n} - Y_{n+h}$, and the MSE of this forecast is

$$MSE^{(M)}_{h|n} = E_{\mathcal{M}}\left[e^{(M)}_{n+h|n}\right]^2, \tag{2.5}$$

where the expectation is taken over the true model $\mathcal{M}$.

Suppose M represents a stationary ARMA(p,q) model with zero mean,

$$Y_t - \phi_1 Y_{t-1} - \ldots - \phi_p Y_{t-p} = \epsilon_t + \theta_1 \epsilon_{t-1} + \ldots + \theta_q \epsilon_{t-q},$$

where $\{\epsilon_t\}$ are assumed to be white noise (mutually independent) with zero mean and common variance σ_ϵ^2.

Let B denote the backward shift operator B such that $B^h Y_t = Y_{t-h}$. Then the above model can be written in the polynomial form

$$\phi(B) Y_t = \theta(B) \epsilon_t,$$

where $\phi(B) = (1 - \phi_1 B - \ldots - \phi_p B^p)$ and $\theta(B) = (1 + \theta_1 B + \ldots + \theta_q B^q)$ respectively represent the pth order AR and qth order MA polynomials. We assume that both $\phi(z) = 0$ and $\theta(z) = 0$ have no roots on or inside the unit circle, so that $\{Y_t\}$ is causal and strictly invertible, and has an infinite moving average representation:

$$Y_t = \psi_0 \epsilon_t + \psi_1 \epsilon_{t-1} + \ldots, \tag{2.6}$$

where for $j = 0, 1, 2, \ldots$, ψ_j is the coefficient of B^j in the infinite expansion $\phi^{-1}(B)\theta(B) = \sum_{j=0} \psi_j B^j$.

We have started with the general MMSE forecasting problem (Eq. 2.4), and from now on, we will focus on the Gaussian ARMA model. Under normality the MMSE predictor and the best linear predictor (BLP) coincide. Therefore, even without the normality assumption, the following discussion and results will remain the same if we consider the BLP instead of the MMSE predictor, which has been the convention of the forecasting based on ARMA models.

For a causal and invertible ARMA model, the temporal dependence decays geometrically fast, so the MMSE predictor based on $\{Y_1, \ldots, Y_n\}$ and on the infinite past $\{\ldots, Y_0, \ldots, Y_n\}$ are very close as long as n is reasonably large. As a result, we redefine the notation $\hat{Y}^{(M)}_{n+h|n}$ as

$$\hat{Y}^{(M)}_{n+h|n} = E_M(Y_{n+h} | \ldots, Y_0, \ldots, Y_n). \tag{2.7}$$

Using the representation of Eq. 2.6, one can write Y_{n+h} as:

$$Y_{n+h} = \psi_0\epsilon_{n+h} + \psi_1\epsilon_{n+h-1} + \ldots + \psi_{h-1}\epsilon_{n+1} + \psi_h\epsilon_n + \psi_{h+1}\epsilon_{n-1} + \ldots, \quad (2.8)$$

Then the MMSE predictor of Y_{n+h} is:

$$\hat{Y}^{(M)}_{n+h|n} = E_M(Y_{n+h}|Y_1, \ldots, Y_n) = E_M(Y_{n+h}|\epsilon_1, \ldots, \epsilon_n) = \psi_h\epsilon_n + \psi_{h+1}\epsilon_{n-1} + \ldots.$$

Now, let $\hat{Y}^{(\hat{M})}_{n+h|n}$ denote the MMSE predictor of Y_{n+h} when the parameters of model M are unknown and are estimated from observations $Y_1, \ldots, Y_n$. Then, this MMSE predictor is given by

$$\hat{Y}^{(\hat{M})}_{n+h|n} = \hat{\psi}_h\epsilon_n + \hat{\psi}_{h+1}\epsilon_{n-1} + \ldots,$$

where $\hat{\psi}_j$'s are estimated from the data $Y_1, \ldots, Y_n$. The corresponding forecast error and MSE are respectively given by

$$e^{(\hat{M})}_{n+h|n} = \hat{Y}^{(\hat{M})}_{n+h} - Y_{n+h}, \quad MSE^{(\hat{M})}_{h|n} = E_{\mathcal{M}}\left[e^{(\hat{M})}_{n+h|n}\right]^2, \quad (2.9)$$

2.2.1 Forecast errors from true model

If the assumed model M is the true model $\mathcal{M}$ with known parameters, then the representation shown in Eq. 2.8 is the correct expansion of Y_{n+h}, and consequently the forecast error

$$e^{(\mathcal{M})}_{n+h|n} = \hat{Y}^{(\mathcal{M})}_{n+h|n} - Y_{n+h} = \psi_0\epsilon_{n+h} + \psi_1\epsilon_{n+h-1} + \ldots + \psi_{h-1}\epsilon_{n+1}, \quad (2.10)$$

yields the MSE

$$MSE^{(\mathcal{M})}_{h|n} = \left(\psi_0^2 + \ldots + \psi_{h-1}^2\right)\sigma_\epsilon^2, \quad (2.11)$$

where, as assumed earlier, σ_ϵ^2 is the common variance of the residuals ϵ. This quantity $MSE^{(\mathcal{M})}_{h|n}$ can be referred to as the "inherent model error of forecast". It is a measure of forecast error in the *best possible scenario* where the true model specification

(including the parameters) is known, and thus can be interpreted as the *unavoidable error*.

Now consider the situation where the parameters of the true model are unknown. Then the forecast error is given by $e^{(\widehat{\mathcal{M}})}_{n+h|n} = \hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - Y_{n+h}$. Thus the MSE can be decomposed as

$$\begin{aligned} MSE^{(\widehat{\mathcal{M}})}_{h|n} &= E_{\mathcal{M}}\left[e^{(\widehat{\mathcal{M}})}_{n+h|n}\right]^2 = E\left[\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - Y_{n+h}\right]^2 \\ &= E_{\mathcal{M}}\left[\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - \hat{Y}^{(\mathcal{M})}_{n+h|n} + \hat{Y}^{(\mathcal{M})}_{n+h|n} - Y_{n+h}\right]^2 \\ &= E_{\mathcal{M}}\left[\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - \hat{Y}^{(\mathcal{M})}_{n+h|n} + \epsilon^{(\mathcal{M})}_{n+h}\right]^2 \\ &= E_{\mathcal{M}}\left[\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - \hat{Y}^{(\mathcal{M})}_{n+h|n}\right]^2 + E_{\mathcal{M}}\left[\epsilon^{(\mathcal{M})}_{n+h}\right]^2. \end{aligned}$$

The last step follows from the following facts: (i) The difference $\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - \hat{Y}^{(\mathcal{M})}_{n+h|n}$ (i.e, the difference between the predictor with the true model parameters and that with estimated model parameters) depends on the past residuals $\epsilon_n, \epsilon_{n-1}, \ldots, \ldots$ whereas, (ii) from Eq. 2.10 it is clear that $e^{(\mathcal{M})}_{n+h|n}$ depends only on the future residuals $\epsilon_{n+1}, \ldots, \epsilon_{n+h}$. Consequently, by mutual independence of residuals, $\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - \hat{Y}^{(\mathcal{M})}_{n+h|n}$ is independent of $\epsilon^{(\mathcal{M})}_{n+h|n}$ and the expectation of the product term vanishes. We thus have the following well-known (e.g. Mazzeu et al., 2018) result:

Proposition 1. The MSE of forecast under the true model $\mathcal{M}$ with unknown coefficients, where the model parameters are estimated from observed data, can be decomposed as

$$MSE^{(\widehat{\mathcal{M}})}_{h|n} = \delta^{(\widehat{\mathcal{M}},\mathcal{M})}_{h|n} + MSE^{(\mathcal{M})}_{h|n}, \tag{2.12}$$

where $MSE^{(\mathcal{M})}_{h|n}$ is the inherent model error given by Eq. 2.11, and

$$\delta^{(\widehat{\mathcal{M}},\mathcal{M})}_{h|n} = E_{\mathcal{M}}\left[\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n} - \hat{Y}^{(\mathcal{M})}_{n+h|n}\right]^2 \tag{2.13}$$

represents the contribution of "parameter uncertainty" to the overall forecast error.

Note that the quantity $\delta^{(\widehat{\mathcal{M}},\mathcal{M})}_{h|n}$ defined in Proposition 1 depends on the sample size n and the forecast horizon h, and loosely speaking, is expected to converge to

zero for fixed h as n goes to infinity, under fairly mild conditions related to the convergence of the parameter estimators to the true model parameters (see Fuller (1996)). Proposition 1 also suggests that if it is possible to estimate $MSE_{h|n}^{(\widehat{\mathcal{M}})}$ and $MSE_{h|n}^{(\mathcal{M})}$, then their difference will provide an estimate of $\delta_{h|n}^{(\widehat{\mathcal{M}},\mathcal{M})}$.

2.3 Forecast errors from misspecified models

Now assume that M is a misspecified model with parameter θ_M (typically a vector), which is different from the true model $\mathcal{M}$ with parameter $\theta_{\mathcal{M}}$. To forecast Y_{n+h} from n data points using M, one has to estimate the parameters θ_M from observations $Y_1, \ldots, Y_n$ and follow the procedure described earlier to obtain the predictor $\hat{Y}_{n+h}^{(\hat{M})}$. The associated error will be denoted by $e_{n+h}^{(\hat{M})}$ and the MSE by $MSE_{h|n}^{(\hat{M})}$.

To decompose $MSE_{h|n}^{(\hat{M})}$, we visualize a *population version of the misspecified model* with some "true value" of θ_M, and define the MMSE predictor $\hat{Y}_{n+h}^{(M)}$ assuming that true value is known. While such a "true value" from a misspecified model does not make much sense, we can consider it to be the "best value" or "optimal value" (in some sense) that generates observations from the time series from the misspecified model similar to those generated by the true model. We will provide a more precise definition of such a best value in the context of ARMA models later. Now $MSE_{h|n}^{(\hat{M})}$ can be decomposed as follows:

$$\begin{aligned}
MSE_{h|n}^{(\hat{M})} &= E_{\mathcal{M}}\left[\hat{Y}_{n+h|n}^{(\hat{M})} - Y_{n+h}\right]^2 \\
&= E_{\mathcal{M}}\left[\hat{Y}_{n+h|n}^{(\hat{M})} - \hat{Y}_{n+h|n}^{(M)} + \hat{Y}_{n+h|n}^{(M)} - Y_{n+h}\right]^2 \\
&= E_{\mathcal{M}}\left[\hat{Y}_{n+h|n}^{(\hat{M})} - \hat{Y}_{n+h|n}^{(M)} + e_{n+h|n}^{(M)}\right]^2 \\
&= E_{\mathcal{M}}\left[\hat{Y}_{n+h|n}^{(\hat{M})} - \hat{Y}_{n+h|n}^{(M)}\right]^2 + 2E_{\mathcal{M}}\left[\left(\hat{Y}_{n+h|n}^{(\hat{M})} - \hat{Y}_{n+h|n}^{(M)}\right) e_{n+h|n}^{(M)}\right] + E_{\mathcal{M}}\left[e_{n+h|n}^{(M)}\right]^2 \\
&= \delta_{n|h}^{(\hat{M},M)} + 2E_{\mathcal{M}}\left[\left(\hat{Y}_{n+h|n}^{(\hat{M})} - \hat{Y}_{n+h|n}^{(M)}\right) e_{n+h|n}^{(M)}\right] + MSE_{h|n}^{(M)}, \qquad (2.14)
\end{aligned}$$

which takes a form similar to Eq. 2.12 except for the fact that the product term does not vanish in this case. However, for a large sample size, both $\delta_{n|h}^{(\hat{M},M)}$ and the

product term should be small. The last term can again be decomposed as

$$\begin{aligned} MSE_{h|n}^{(M)} &= E_{\mathcal{M}}\left[\hat{Y}_{n+h}^{(\hat{M})} - Y_{n+h}\right]^2 \\ &= E_{\mathcal{M}}\left[\hat{Y}_{n+h}^{(M)} - \hat{Y}_{n+h}^{(\mathcal{M})} + \hat{Y}_{n+h}^{(\mathcal{M})} - Y_{n+h}\right]^2 \\ &= \delta_{h|n}^{(M,\mathcal{M})} + MSE_{h|n}^{(\mathcal{M})}, \end{aligned} \tag{2.15}$$

where

$$\delta_{h|n}^{(M_1,M_2)} = E_{\mathcal{M}}\left[\hat{Y}_{n+h}^{(M_1)} - \hat{Y}_{n+h}^{(M_2)}\right]^2, \tag{2.16}$$

is a measure of the impact of *model uncertainty* on the forecast made from two models M_1 and M_2.

Remark 1. We note here that $\delta_{h|n}^{(M,\mathcal{M})} \geq 0$ for all h and n, with equality holding if and only if M is the true model $\mathcal{M}$. Consequently,

$$MSE_{h|n}^{(M)} \geq MSE_{h|n}^{(\mathcal{M})}, \tag{2.17}$$

with equality holding if and only if M is the true model $\mathcal{M}$.

Remark 2. Note that $\delta_{h|n}^{(M_1,M_2)}$ can be called the *Expected squared discrepancy* between forecasts of Y_{n+h} made from two models M_1 and M_2, that may or may not be completely specified (in terms of parameter values). Thus, $\delta_{h|n}^{(M,\mathcal{M})}$ in the RHS of Eq. 2.15, $\delta_{h|n}^{(\hat{M},M)}$ in the RHS of Eq. 2.14 and $\delta^{(\widehat{\mathcal{M}},\mathcal{M})}$ in Eq. 2.13 are all special cases of this discrepancy.

Davies and Newbold (1980) derived a closed form expression for $\delta_{h|n}^{(M,\mathcal{M})}$ when the true model $\mathcal{M}$ and the misspecified model M are respectively ARMA$(p_{\mathcal{M}}, q_{\mathcal{M}})$ and ARMA(p_M, q_M). We state the main result of Davies and Newbold (1980) below:

Proposition 2. (Davies and Newbold, 1980). Let the true model $\mathcal{M}$ be an ARMA$(p_{\mathcal{M}}, q_{\mathcal{M}})$ model represented by $\phi_{\mathcal{M}}(B)Y_t = \theta_{\mathcal{M}}(B)\epsilon_t$ where $\phi_{\mathcal{M}}(B)$ and $\theta_{\mathcal{M}}(B)$ are polynomials of order $p_{\mathcal{M}}$ and $q_{\mathcal{M}}$ respectively and let the misspecified model M be an ARMA(p_M, q_M) model represented by $\phi_M(B)Y_t = \theta_M(B)\epsilon_t$ where $\phi_M(B)$ and $\theta_M(B)$ are polynomials of order p_M and q_M respectively. Also,

let $\phi_{\mathcal{M}}(B)\theta_{\mathcal{M}}^{-1}(B)\epsilon_t = \sum_{j=0}^{\infty}\psi_{\mathcal{M},j}\epsilon_{t-j}$ and $\phi_M(B)\theta_M^{-1}(B)\epsilon_t = \sum_{j=0}^{\infty}\psi_{M,j}\epsilon_{t-j}$ be the infinite moving average representations of $\mathcal{M}$ and M respectively. Finally, let $\phi_M(B)\theta_M^{-1}(B)\phi_{\mathcal{M}}^{-1}(B)\theta_{\mathcal{M}}(B) = \sum_{j=0}^{\infty}\widetilde{\psi}_j B^j$. Then the contribution of the model uncertainty $\delta_{h|n}^{(M,\mathcal{M})}$ associated with the h-steps ahead forecast made from models M and $\mathcal{M}$ is given by

$$\delta_{h|n}^{(M,\mathcal{M})} = \sum_{j=0}^{\infty}\{\psi_{\mathcal{M},j+h} - a_j(h)\}^2, \tag{2.18}$$

where

$$a_j(h) = \sum_{k=0}^{j}\psi_{M,h+k}\widetilde{\psi}_{j-k}.$$

Proposition 2 assumes that the misspecified model is completely specified and the coefficients $\psi_{M,j}$'s are known. This is obviously not the case in practice. In the process of forecasting with a misspecified model, one would estimate the parameters of such a model from the data generated by the true model. We need to establish a connection of such an estimate with the specified value of the parameter in model M. This topic is revisited in section 2.3.1.

Finally, substituting Eq. 2.16 in Eq. 2.14 we arrive at the following proposition:

Proposition 3 (Decomposition of forecast MSE)**.** The MSE of prediction under the misspecified model M with unknown coefficients, where the model parameters are estimated from observed data, can be decomposed as

$$MSE_{h|n}^{(\hat{M})} = \delta_{h|n}^{(\hat{M},M)} + 2E_{\mathcal{M}}\left[\left(\hat{Y}_{n+h}^{(\hat{M})} - \hat{Y}_{n+h}^{(M)}\right)\epsilon_{n+h}^{(M)}\right] + \delta_{h|n}^{(M,\mathcal{M})} + MSE_{h|n}^{(\mathcal{M})}, \tag{2.19}$$

where $MSE_{h|n}^{(\mathcal{M})}$ is the inherent model error given by Eq. 2.11, $\delta_{h|n}^{(M,\mathcal{M})}$ is given by Eq. 2.16 and $\delta_{h|n}^{(\hat{M},M)}$ is given by Eq. 2.13.

Now, for any model M, we can write $\hat{Y}_{n+h}^{(\widehat{M})} = g_n(\widehat{\theta}_{M,n})$ and $\hat{Y}_{n+h}^{(M)} = g_n(\theta_M)$, where θ_M denotes the model parameter, $\widehat{\theta}_{M,n}$ its estimator based on observations $Y_1, \ldots, Y_n$, and $g_n(\cdot)$ is a continuous function which is finite for all n. If $\hat{\theta}_n$ is a consistent estimator of θ, as is the case for maximum likelihood estimators of ARMA model parameters, assuming normality of innovations. Then by the continuous mapping theorem, $g_n(\widehat{\theta}_{M,n}) - g_n(\theta_M)$ converges to zero in probability as $n \to \infty$. Consequently,

the first two terms on the RHS of (2.19) $\delta_{h|n}^{(\hat{M},M)}$ and $2E_{\mathcal{M}}\left[\left(\hat{Y}_{n+h}^{(\hat{M})}-\hat{Y}_{n+h}^{(M)}\right)\right]$ both converge to zero. Thus we arrive at the following corollary.

Corollary 1. For large n, the MSE of prediction under the misspecified model M with unknown coefficients, where the model parameters are consistently estimated from observed data, can be approximated as

$$MSE_{h|n}^{(\widehat{M})} \approx MSE_{h|n}^{(M)} = \delta_{h|n}^{(\widehat{M},M)} + MSE_{h|n}^{(\mathcal{M})}.$$

2.3.1 Optimal parameters of a misspecified model

The foregoing discussion suggests the need to visualize and define "true" parameters of the misspecified model in terms of their estimated version. Davies and Newbold (1980) provided such a connection by assuming that the misspecified model ARMA(p_M, q_M) has an AR representation of order p_M, and estimated the parameters using least square estimation. The true parameters were then taken as the probability limits of the least square estimator. However, whereas this definition seems reasonable if the misspecified model is AR(p_M), it does not seem to incorporate additional model uncertainty if the misspecified model if ARMA(p_M, q_M) with $q_M \geq 1$, and consequently does not provide a way to define "true" parameters of the true misspecified model.

We now provide a more formal definition of an "optimal value of the parameter of a misspecified model" in the context of forecasting from ARMA models.

Definition 1 (Optimal value of misspecified model parameter)**.** Let M be any arbitrary ARMA model with parameter θ_M. The optimal value of θ_M is defined as the value θ_M^* that minimizes the MMSE of prediction of Y_{n+1} based on observations $\{\ldots, Y_{-1}, Y_0, Y_1, \ldots, Y_n\}$. Formally,

$$\theta_M^* = \arg\min_{\theta_M} E_{\mathcal{M}}\left[\hat{Y}_{n+1}^{\theta_M} - Y_{n+1}\right]^2,$$

where $\hat{Y}_{n+1}^{\theta_M}$ denotes the predictor of Y_{n+1} based on model M and parameter value θ_M.

Remark 3. If we require in addition that the ARMA model M is Gaussian, then this best misspecified model is equivalent to the one that minimizes the Kullback-Leibler

divergence from θ_M to $\mathcal{M}$:

$$\theta_M^* = \arg\min_{\theta_M} \mathrm{KL}(\mathcal{M}||\theta_M),$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence. This equivalence is explained in Appendix C.

Remark 4. Definition 1 is equivalent to the notion of the "true" misspecified model by Davies and Newbold (1980) if the misspecified model M is purely autoregressive. In fact, if the misspecified model is an autoregressive process, then according to Definition 1, θ_M^* is the Yule-Walker estimator based on the autocovariances of $\{Y_t\}$. Davies and Newbold (1980) suggest using the limit of the least squares estimators, which is exactly the Yule-Walker estimator. Therefore, the two definitions are equivalent when the misspecified model is autoregressive.

Remark 5. Definition 1 also provides us with a specific algorithm to obtain the value of θ_M using available data $Y_1, \ldots, Y_n$. Adopting the notations of Proposition 2, suppose the true model is a causal and invertible ARMA$(p_{\mathcal{M}}, q_{\mathcal{M}})$: $\phi_{\mathcal{M}}(B)Y_t = \theta_{\mathcal{M}}(B)\epsilon_t$, and the misspecified model M is a causal and invertible ARMA(p_M, q_M): $\phi_M(B)Y_t = \theta_M(B)\epsilon_t$. Here without loss of generality we assume there is no intercept in both models, and $\mathrm{Var}(\varepsilon_t) = 1$. We describe how to find the "optimal" parameters under the misspecified model. Instead of using AR and MA coefficients, the misspecified model can be equivalently parametrized by the factorizations $\phi_M(z) = (1 - w_1 z) \cdots (1 - w_{p_M} z)$ and $\theta^*(z) = (1 - v_1 z) \cdots (1 - v_{q_M} z)$. The one-step ahead prediction error (using the infinite past) under the model M is given by

$$\frac{\phi_M(B)}{\theta_M(B)} Y_t = \frac{\phi_M(B)}{\theta_M(B)} \times \frac{\theta_{\mathcal{M}}(B)}{\phi_{\mathcal{M}}(B)} \epsilon_t.$$

Let

$$\frac{\phi_M(z)}{\theta_M(z)} \times \frac{\theta_{\mathcal{M}}(z)}{\phi_{\mathcal{M}}(z)} = \sum_{k=0}^{\infty} \widetilde{\psi}_k z^k.$$

Note that each $\tilde{\psi}$ in the preceding equation depends on $\{w_1, \ldots, w_{p_M}, v_1, \ldots, v_{q_M}\}$ implicitly. According to Definition 1, the optimal values of the parameters

$\{w_1^*, \ldots, w_{p_M}^*, v_1^*, \ldots, v_{q_M}^*\}$ are given by

$$\arg \min_{\{w_i, v_j\}} \sum_{k=0}^{n} \tilde{\psi}_k^2. \tag{2.20}$$

The optimization problem (Eq. 2.20) is related to the maximum likelihood estimation (MLE) of the ARMA model. It will be convenient in practice to find θ_M^* through simulation: (i) simulate a long series (e.g. of length 100,000) from the true model, and (ii) find the MLE under the misspecified model. This MLE serves as an estimate of θ_M^*, whose accuracy can be controlled by the length of the simulated series.

Example 1. Suppose $\mathcal{M}$ is AR(1): $Y_t = \phi Y_{t-1} + \epsilon_t$, and let M: MA(1) be the misspecified model: $Y_t = \epsilon_t + \theta \epsilon_{t-1}$. Assume $|\phi| < 1$ so that the true model is invertible. It holds that

$$\sum_{k=0}^{n} \tilde{\psi}_k z^k := \frac{1}{(1-\phi z)(1+\theta z)} = \frac{1}{\phi + \theta} \left[\frac{\phi}{1-\phi z} + \frac{\theta}{1+\theta z} \right]$$

and

$$\sum_{k=1}^{n} \tilde{\psi}_k^2 = \frac{1}{(\phi+\theta)^2} \sum_{k=0}^{\infty} \left[\phi^{k+1} + (-1)^k \theta^{k+1} \right]^2.$$

Therefore, the optimal θ^* is the θ that minimizes the preceding infinite sum. We give the optimal θ^* corresponding to $\phi \in \{.1, .2, \ldots, .9\}$ in Table 2.1. The estimated $\hat{\theta}^*$ obtained from a simulated series of length 100,000 is also reported in the third row of the table.

Table 2.1: Optimal θ^* for the MA(1), when the true model is AR(1).

ϕ	.1	.2	.3	.4	.5	.6	.7	.8	.9
θ^*	.099	.193	.279	.356	.428	.496	.565	.640	.735
$\hat{\theta}^*$	.098	.192	.280	.360	.429	.497	.562	.642	.735

2.3.2 Inflation of prediction error

Based on the definitions and results of the previous section, we now define measures of inflation of forecast MSE due to model misspecification and parameter uncer-

tainty. First, note that in the problem of obtaining an h-step ahead forecast Y_{n+h} from observations $Y_1, \ldots, Y_n$ generated by a true ARMA model $\mathcal{M}$, the unavoidable or *intrinsic* uncertainty is $MSE_{h|n}^{(\mathcal{M})}$ given by Eq. 2.11. When an arbitrary model M is used to make the forecast, model parameters are estimated from the data and subsequently plugged into the forecast. This process inflates the forecast error by $MSE_{h|n}^{(\hat{M})} - MSE_{h|n}^{(\mathcal{M})}$, where $MSE_{h|n}^{(\hat{M})}$ is given by Proposition 3. We now formally define this measure of inflation and its components, relative to the intrinsic uncertainty $MSE_{h|n}^{\mathcal{M}}$.

Definition 2 (Total Percentage Inflation or TPI)**.** In the problem of obtaining an h-step ahead forecast Y_{n+h} from observations $Y_1, \ldots, Y_n$ generated by a true ARMA model $\mathcal{M}$, assuming model M and estimating its parameters, the *total percentage inflation* (TPI) is given by

$$\begin{aligned} \mathrm{TPI}_{h|n}^{(M)} &= \frac{MSE_{h|n}^{(\hat{M})} - MSE_{h|n}^{(\mathcal{M})}}{MSE_{h|n}^{(\mathcal{M})}} \times 100 \qquad (2.21) \\ &= \begin{cases} \frac{\delta_{h|n}^{(\hat{M},M)}}{MSE^{(\mathcal{M})}} \times 100, & \text{if } M \equiv \mathcal{M}, \\ \frac{\delta_{h|n}^{(\hat{M},M)} + 2E_{\mathcal{M}}\left[\left(\hat{Y}_{n+h}^{(\hat{M})} - \hat{Y}_{n+h}^{(M)}\right)\epsilon_{n+h}^{(M)}\right] + \delta_{h|n}^{(M,\mathcal{M})}}{MSE^{(\mathcal{M})}} \times 100, & \text{otherwise} \end{cases} \end{aligned}$$

Definition 3 (Percentage Estimation Inflation or PEI)**.** In the problem of obtaining an h-step ahead forecast Y_{n+h} from observations $Y_1, \ldots, Y_n$ generated by a true ARMA model $\mathcal{M}$, assuming model M and estimating its parameters, the percentage *estimation inflation* (PEI) is given by

$$\begin{aligned} \mathrm{PEI}_{h|n}^{(M)} &= \frac{MSE_{h|n}^{(\hat{M})} - MSE_{h|n}^{(M)}}{MSE_{h|n}^{(\mathcal{M})}} \times 100 \qquad (2.22) \\ &= \begin{cases} \frac{\delta_{h|n}^{(\hat{M},M)}}{MSE_{h|n}^{(\mathcal{M})}} \times 100, & \text{if } M \equiv \mathcal{M}, \\ \frac{\delta_{h|n}^{(\hat{M},M)} + 2E_{\mathcal{M}}\left[\left(\hat{Y}_{n+h}^{(\hat{M})} - \hat{Y}_{n+h}^{(M)}\right)\epsilon_{n+h}^{(M)}\right]}{MSE_{h|n}^{(\mathcal{M})}} \times 100, & \text{otherwise} \end{cases} \end{aligned}$$

where M represents the completely specified model with some underlying "best" parameter values per Definition 1.

Definition 4 (Percentage Misspecification Inflation or PMI)**.** In the problem of obtaining an h-step ahead forecast Y_{n+h} from observations $Y_1, \ldots, Y_n$ generated by a true ARMA model $\mathcal{M}$, assuming model M and estimating its parameters, the percentage *misspecification inflation* (PMI) is given by

$$\begin{aligned} \mathrm{PMI}_{h|n}^{M} &= \frac{MSE_{h|n}^{(M)} - MSE_{h|n}^{(\mathcal{M})}}{MSE_{h|n}^{(\mathcal{M})}} \times 100 \qquad (2.23) \\ &= \begin{cases} 0, & M \equiv \mathcal{M} \\ \frac{\delta_{h|n}^{(M,\mathcal{M})}}{MSE_{h|n}^{(\mathcal{M})}} \times 100, & \text{otherwise} \end{cases} \end{aligned}$$

where M represents the completely specified model with some underlying "best" parameter value as per Definition 1.

2.3.3 Extension to other ARMA class of models

Seasonal ARMA (SARMA) models allow users to incorporate seasonal effects into the ARMA framework. A typical multiplicative SARMA $(p, q) \times (P, Q)_s$ model (Brockwell and Davis, 2002) is of the form:

$$\phi_p(B)\Phi_P(B^s)Y_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t, \qquad (2.24)$$

where as before, B^h denotes the h-step backshift operator for $h \geq 1$, $\phi_p(\cdot)$, $\Phi_P(\cdots)$, $\theta_q(\cdot)$ and $\Theta_Q(\cdot)$ are polynomials of orders p, P, q and Q respectively, and $\{\epsilon_t\}$ is a white noise term. Such models can include more than two seasonal polynomials and consequently allow for multiple sources of seasonality (e.g., daily, weekly etc.) to be incorporated into the model. Because the model shown in Eq. 2.24 can easily be expressed in the form of Eq. 2.6, all the concepts, definitions and results discussed so far in this section can be extended in a straightforward manner to SARMA models.

A similar decomposition of the mean squared prediction error can be given for the ARIMA process. For simplicity, we only consider the ARIMA models with integration order 1. Suppose the true model of $\{X_t\}$ is an ARIMA$(p_{\mathcal{M}}, 1, q_{\mathcal{M}})$

$$\phi(B)\Delta X_t = \theta(B)\epsilon_t.$$

Assume the (possibly) misspecified model M is ARIMA$(p_M, 1, q_M)$, *i.e.* it is still an ARIMA with integrated order 1, but with (possibly) misspecified p_M and q_M. Let $Y_t = \Delta X_t$, then $\{Y_t\} \sim$ ARMA$(p_{\mathcal{M}}, 0, q_{\mathcal{M}})$, and the decomposition discussed earlier holds for $\{Y_t\}$. We now describe how these results can be adapted for the process $\{X_t\}$. First of all, note that

$$\hat{X}^{(M)}_{h|n} = X_n + \sum_{k=1}^{h} \hat{Y}^{(M)}_{k|n}, \quad \text{and} \quad e^{(M)}_{h|n}(X) = X_{n+h} - \hat{X}^{(M)}_{h|n} = \sum_{k=1}^{h} e^{(M)}_{k|n}(Y)$$

for any model M, whether it is true or misspecified. Here we use (X) and (Y) to specify the prediction errors for $\{X_t\}$ or $\{Y_t\}$ respectively. Proposition 1 leads to

$$MSE^{(\widehat{\mathcal{M}})}_{h|n}(X) = \delta^{(\widehat{\mathcal{M}},\mathcal{M})}_{h|n}(X) + MSE^{(\mathcal{M})}_{h|n}(X),$$

where the inherent $MSE^{(\mathcal{M})}_{h|n}(X)$ is (comparing Eq. 2.11):

$$MSE^{(\mathcal{M})}_{h|n}(X) = \sigma^2_\epsilon \cdot \sum_{k=1}^{h} (\psi_0 + \ldots + \psi_{k-1})^2 .$$

Note that the inherent prediction MSE of $\{X_t\}$ goes to infinity as the forecast horizon h increases. Proposition 3 translates into

$$MSE^{(\hat{M})}_{h|n}(X) = \delta^{(\hat{M},M)}_{h|n}(X) + 2E_{\mathcal{M}}\left[\left(\hat{X}^{(\hat{M})}_{n+h} - \hat{X}^{(M)}_{n+h}\right) e^{(M)}_{n+h}(X)\right] + \\ + \delta^{(M,\mathcal{M})}_{h|n}(X) + MSE^{(\mathcal{M})}_{h|n}(X).$$

The Inflation measures TPI, PEI and PMI can then be defined similarly for the ARIMA process $\{X_t\}$. However, it is important to note that the above procedure can be generalized to any integration order only when it is the same for the ARIMA processes under comparison. Thus, our framework will allow us to compare an ARIMA$(p_1,\ d_1,\ q_1)$ process with an ARIMA$(p_2,\ d_2,\ q_2)$ process if and only if $d_1 = d_2$.

2.4 Simulation algorithm and examples

We now present a simulation process that will help researchers estimate the different components of forecast inflation in a setting where model M is mistakenly used instead of a true model $\mathcal{M}$. It will also help compare different models by assessing their roles in inflating the inherent forecast error.

The input to the simulation code is (i) an ARIMA (p, d, q) model with specified values of known parameter $\theta_{\mathcal{M}}$ (assumed to be the true model $\mathcal{M}$), (ii) the order (p_M, d_M, q_M) of an ARIMA model which is the misspecified model M. A very long time series consisting of $N_{\max}$ observations from the known true model $\mathcal{M}$ is generated, and parameters θ^*_M of the misspecified model M are estimated from these data. Thus an ARIMA(p_M, d_M, q_M) model with parameters θ^*_M is assumed to be the optimal misspecified model M. The number of observations to estimate the parameters depends on the complexity of the true model $\mathcal{M}$, as a larger number of observations will be necessary for more complex processes.

Next, a sample size n and a forecast horizon h is fixed. For each iteration $i = 1, \ldots, I$,

1. A total of $n + h$ observations are generated from the true model $\mathcal{M}$. Let $Y_{t,i}$ denote the t-th observation, $t = 1, \ldots, n + h$, $i = 1, \ldots, I$.

2. The true model $\mathcal{M}$ and misspecified model M are now fitted to the first n observations, yielding estimated parameter $\hat{\theta}_{\mathcal{M},i}$ and $\hat{\theta}^*_{M,i}$ respectively.

3. Four forecasts: $\hat{Y}^{(\hat{M})}_{n+h|n,i}$, $\hat{Y}^{(M)}_{n+h|n,i}$, $\hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n,i}$ and $\hat{Y}^{(\mathcal{M})}_{n+h|n,i}$ of $Y_{n+h,i}$ are now obtained from the respective models using estimated parameter $\hat{\theta}^*$, optimal parameter θ^* of misspecified model M, estimated parameter $\hat{\theta}$ of the true model and known parameter θ of true model $\mathcal{M}$. The forecasts of any true model are estimated with the equations obtained using the backshift notation.

4. The corresponding forecast errors are estimated as:

$$\begin{aligned} e^{(\hat{M})}_{n+h|n,i} &= \hat{Y}^{(\hat{M})}_{n+h|n,i} - Y_{n+h,i}, \quad e^{(M)}_{n+h|n,i} = \hat{Y}^{(M)}_{n+h|n,i} - Y_{n+h,i} \\ e^{(\widehat{\mathcal{M}})}_{n+h|n,i} &= \hat{Y}^{(\widehat{\mathcal{M}})}_{n+h|n,i} - Y_{n+h,i}, \quad e^{(\mathcal{M})}_{n+h|n,i} = \hat{Y}^{(\mathcal{M})}_{n+h|n,i} - Y_{n+h,i} \end{aligned}$$

5. The mean squared errors of forecast $MSE^{(\hat{M})}_{n+h|n}$, $MSE^{(M)}_{n+h|n}$, $MSE^{(\widehat{\mathcal{M}})}_{n+h|n}$ and $MSE^{\mathcal{M}}_{n+h|n}$ are estimated by squaring and averaging the forecast errors $e^{\hat{M}}_{n+h|n,i}$, $e^{M}_{n+h|n,i}$, $e^{(\widehat{\mathcal{M}})}_{n+h|n,i}$ and $e^{\mathcal{M}}_{n+h|n,i}$ over $i = 1, \ldots, I$ respectively.

6. Estimation of inflation ratios:

   (i) $\text{PEI}^{(\mathcal{M})}_{h|n}$ for the true model $\mathcal{M}$ is estimated by substituting the estimated mean squared errors in Eq. 2.22 for $M \equiv \mathcal{M}$. Recall that $\text{PMI}^{(\mathcal{M})}_{h|n}$ for the true model is zero.

   (ii) $\text{PEI}^{(M)}_{h|n}$ for the misspecified model M is estimated by substituting the estimated mean squared errors in Eq. 2.22.

   (iii) $\text{PMI}^{(M)}_{h|n}$ for the misspecified model M is estimated by substituting the estimated mean squared errors in Eq. 2.23.

We now present an example of this simulation algorithm and presentation of the results.

Example: Misspecifying ARMA(1,1) as AR(1) or MA(1):

Assume that the true data generating process is ARMA(1,1) with parameters $\phi = 0.8$ and $\theta = -0.3$, and consider forecasting from such a process using two incorrect models: an AR(1) and a MA(1) process. The simulation is conducted using the process described above with $N_{\max} = 10^6$ data points used to estimate optimal parameters of the misspecified models. Various sample sizes n ranging from 50 to 500 in steps of 50 and forecast horizons $h = 1, 2, 3$ are considered for the simulations. For each (n, h) combination, the mean squared errors and inflation indices PEI and PMI are estimated from $I = 5000$ data sets.

Figure 2.2 shows the break up of the inflation of forecast error associated with two sample sizes of $n = 100$ and $n = 1000$ under the three models. For $n = 100$, the contribution of the parameter uncertainty, i.e., $\text{PEI}_{100|1}$ is 1.74 for the true model ARMA(1,1), and smaller (1.07 for AR(1) and 1.03 for MA(1)) for the two misspecified models. This is an expected outcome, because the number of parameters estimated under the true model is larger than that estimated under each of the misspecified

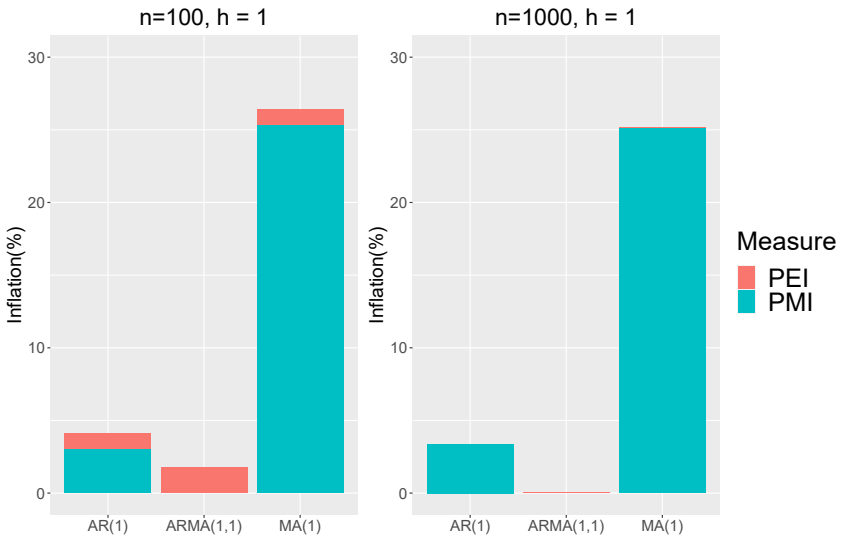

Figure 2.2: Decomposition of percentage inflation in forecast errors for a true ARMA(1,1) process with parameters $\phi = 0.8$, $\theta = -0.3$, and $\sigma^2 = 100$ under misspecified models AR(1) and MA(1). One-step ahead forecasting with a sample size of $n = 100$ (left) and $n = 1000$ (right) is considered.

models. On the contrary, there is no contribution of model uncertainty in the true model, whereas we have $\text{PMI}_{100|1}$ as 3.06 and 25.38 for the AR(1) and the MA(1) models respectively. This is also expected, as MA(1) is a much poorer replacement of ARMA(1,1) than AR(1). For $n = 1000$, as expected, the PEI almost vanishes for all three models, but the PMI remains almost the same for all models.

Figure 2.3 presents a more comprehensive picture of this misspecification, in which indices PEI and PMI are plotted against the sample size n and represented by areas. Three different forecast horizons $h = 1, 2, 3$ are considered. This figure shows that when the correct model is fitted, the inflation of forecast error is only affected by the estimation of parameters, which decreases as the sample size increases. Otherwise, when fitting misspecified models, the inflation is larger and comes mostly from the incorrect choice of model. The PEI becomes negligible as the sample size increases, but the PMI remains more or less unchanged. The PMI is much larger for MA(1), as compared to that of AR(1), and attains its maximum for $h = 2$. This behavior, of course, may change if the parameters of the true model change.

Example: SARMA/SARIMA models

We consider daily data produced by a seasonal ARMA $(2, 2) \times (2, 2)_7$ process with

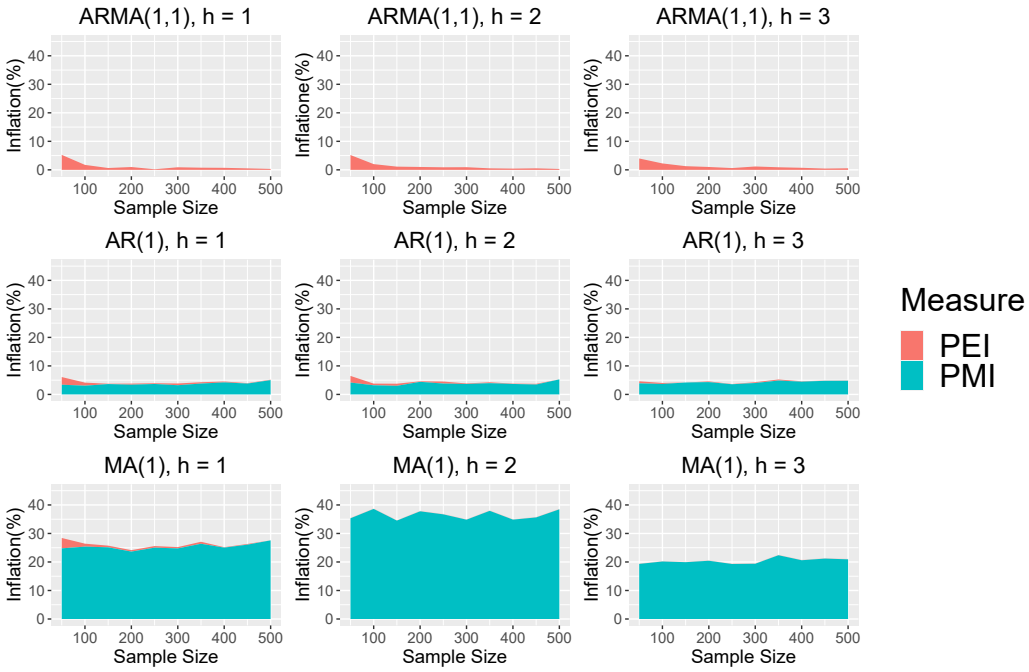

Figure 2.3: Decomposition of percentage inflation in forecast errors for a true ARMA(1,1) process with parameters $\phi = 0.8$, $\theta = -0.3$, and $\sigma^2 = 100$ under misspecified models AR(1) and MA(1).

a weekly seasonal component, and consider forecasting using the following sequence of models, each of which is "weaker" (in terms of departure from the true model) than the preceding one, in the sense that:

- ARMA(2,2)(1,2)$_7$, where one autoregressive term of the season component is dropped.
- ARMA(2,2)(1,1)$_7$, where both the AR and MA orders of the seasonal component are one less than that of the true model.
- ARMA(2,2)(1,0)$_7$, where the moving average term of the weekly seasonal component is omitted.
- ARMA(2,2), completely dropping the seasonal component.
- ARMA(1,2), incorrectly modeling the main ARMA process.
- ARMA(1,1), the weakest model in the sequence.

Two additional models, an ARMA(2,2)(2,3)$_7$ and an ARMA(2,2)(3,3)$_7$, are also

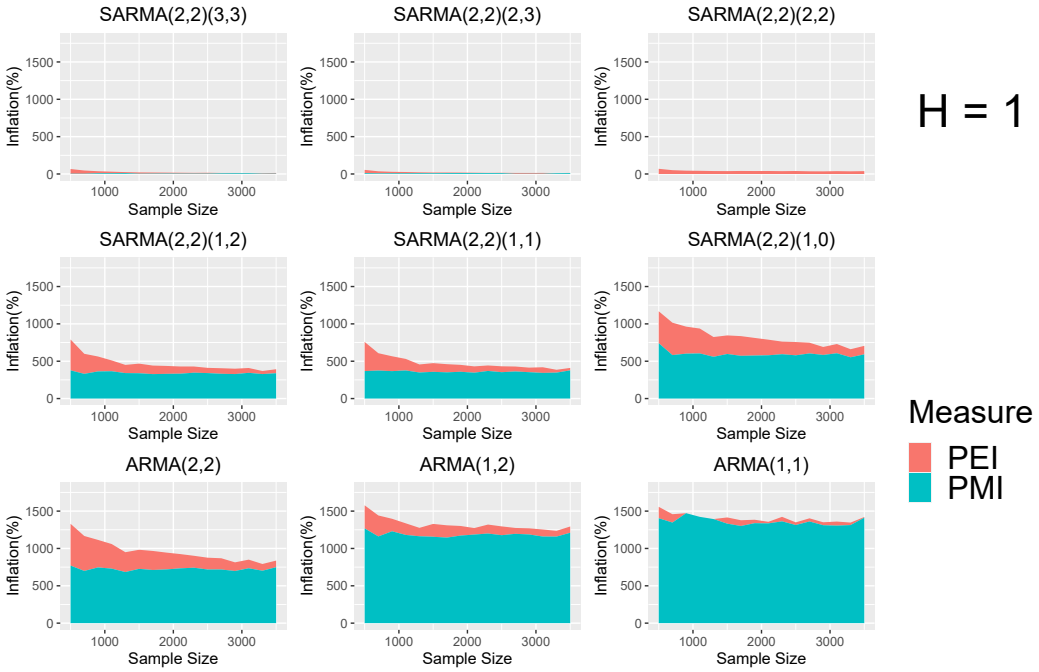

Figure 2.4: Decomposition of percentage inflation in forecast errors for a true SARMA(2,2)$(2,2)_7$ process for 1-step ahead predictions. 5000 simulations have been run for every scenario.

considered, with the aim of analyzing how the forecast is inflated for overfitted models.

Figure 2.4 shows the decomposition of percentage inflation under each model for one-step ahead forecasts. The consequences of forecasting from misspecified models and how the inflation of forecast MSE increases as one drops components of the true model is consistent with the expectations. Additional simulations with 7 and 14-step ahead forecasts (shown in Appendix D) reveal that the consequences of misspecifying models becomes less severe as the forecast horizon increases.

2.5 Application: modelling wind speed

Understanding WSF errors is a good starting point for our objectives, as they could be later converted into wind power generation values using a power curve, or used as an exogenous input for a more advanced WPF model (as previously stated in section 1.4.1). Some ARIMA models considered in the WSF literature are: (i) $(3,1,1)$ (Grigonytė and Butkevičiūtė, 2016), (ii) $(3,1,4)$ (Radziukynas and Klementavicius, 2014), (iii) $(1,1,1)$ (Cadenas et al., 2016), and (iv) $(1,1,0)$ (Cadenas et al., 2016).

Among these four models, Grigonytė and Butkevičiūtė (2016) selects the set of (p,d,q) values of model (i) after providing an extensive model comparison of ARIMA models with combinations of orders up to 10 for both AR and MA components. Thus, assuming model (i) to be the true model, we generate data using different sample sizes (Figure 2.5) and fit the four models to assess the inflation of forecast error for the remaining three. This assessment essentially boils down to the study of the influence of adding a higher order MA parameter (in model (ii)), dropping two AR parameters (in model (iii)), and dropping two AR parameters and the MA term (in model (iv)).

Figures 2.6-2.7 shows the percentage inflation in forecast error when generating data from true ARIMA $(3, 1, 1)$ processes for several forecasting horizons (1- to 6-step, 12-step, and 24-step ahead forecasts). A different sample size is considered to generate the data from the true ARIMA $(3, 1, 1)$ processes: 25 data points (Figure 2.6), and 500 data points (Figure 2.7). Models (iii) and (iv) show the higher inflation, stemming mostly from the incorrect choice of model. The percentage inflation from model (ii) comes mostly from parameter estimation. The later is result of choosing an oversized version of the true model by adding additional MA terms. In addition, the inflation in forecast error also shows the importance of selecting an adequate

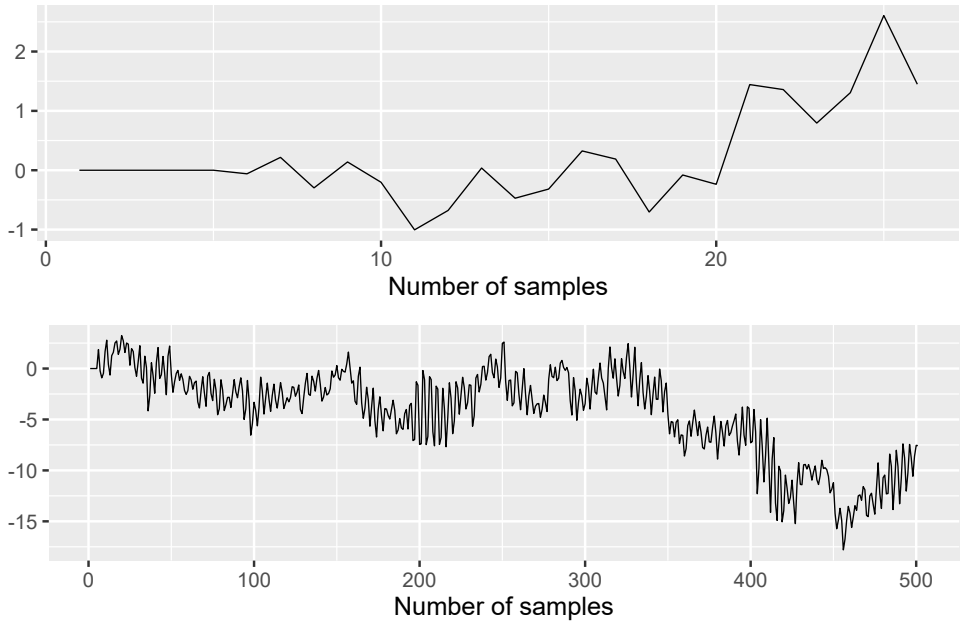

Figure 2.5: Synthetic wind speed time series data generated using a sample size of n = 25 (top) and n = 500 data points (bottom).

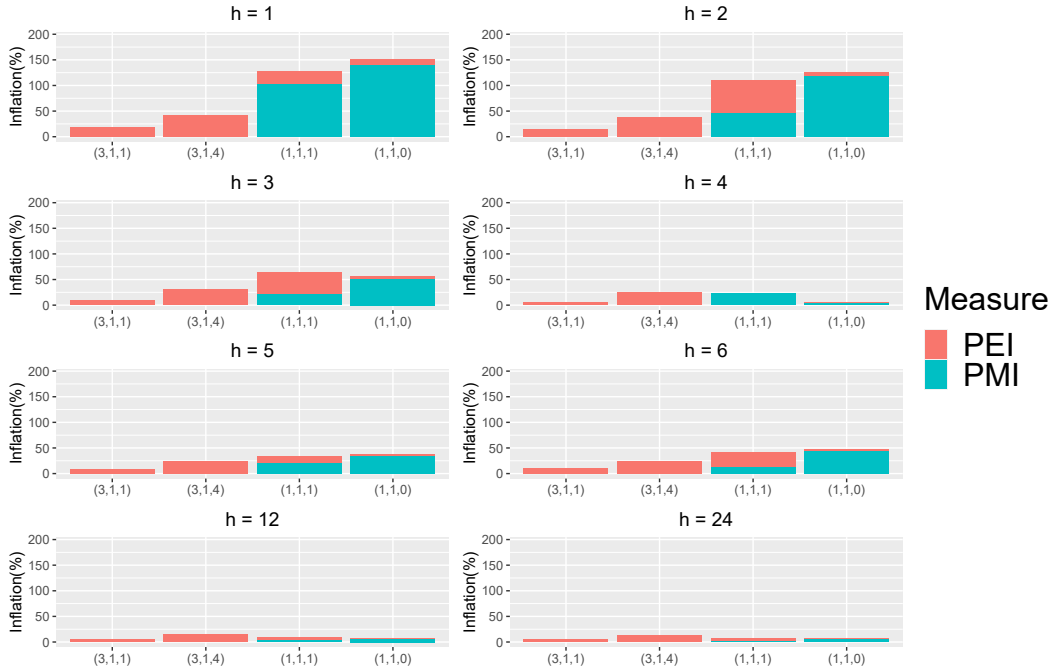

Figure 2.6: Decomposition of percentage inflation in forecast errors for a true ARIMA(3,1,1) process with parameters $\phi_1 = 0.3$, $\phi_2 = -0.8$, $\phi_3 = 0.3$, $\theta = -0.2$, and $\sigma^2 = 100$ for a sample size n = 25. 20000 simulations have been run for every scenario.

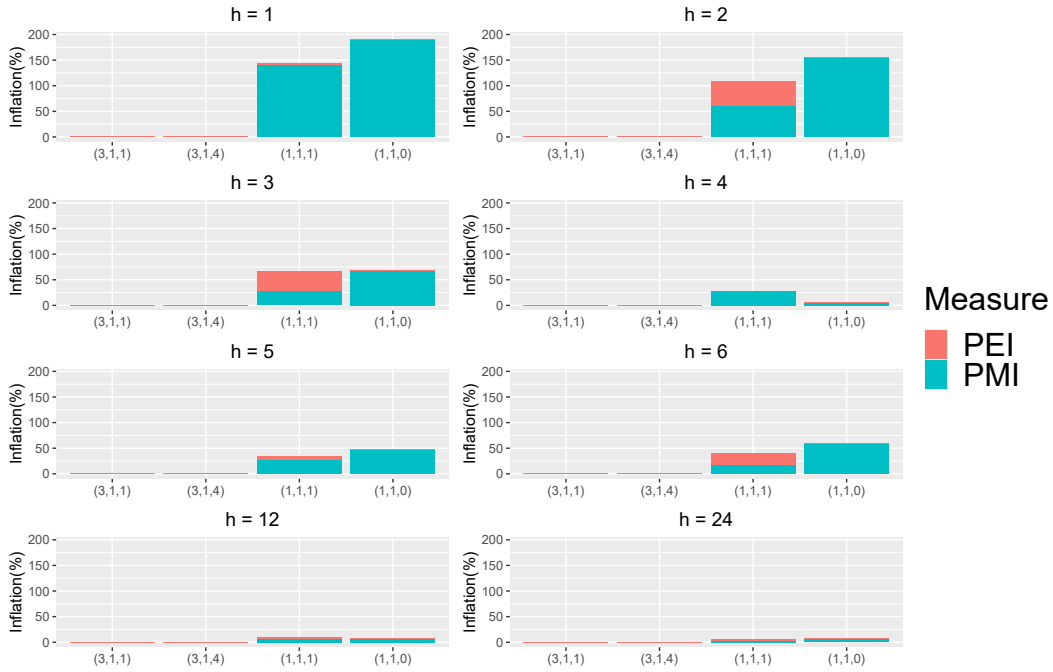

Figure 2.7: Decomposition of percentage inflation in forecast errors for a true ARIMA(3,1,1) process with parameters $\phi_1 = 0.3$, $\phi_2 = -0.8$, $\phi_3 = 0.3$, $\theta = -0.2$, and $\sigma^2 = 100$ for a sample size n = 500. 20000 simulations have been run for every scenario.

sample size. The inflation of the true model decreases when augmenting the sample size, as augmenting the data sample facilitates to identify the true parameters of the model, whereas the inflation is larger for models (iii) and (iv) for a greater sample size.

2.6 Conclusions

In this chapter, we investigate the consequences of using a misspecified model for wind speed time series forecasting on the forecast error. On the basis of a decomposition of the MSE of the forecast obtained from the misspecified model, we define two indices associated with the inflation of the MSE compared to the true model. One quantifies the inflation associated with the incorrect use of a forecasting model whereas the other measures the contribution of parameter uncertainty from the incorrect model. A simulation algorithm is proposed to perform this assessment for any ARIMA or SARIMA model, assuming that both the true model and the misspecified model can be converted to a stationary ARMA process by differencing the same number of times.

The proposed framework helps to assess the consequences of sacrificing information by forecasting from models of lower order compared to true models of higher order. On the other hand, it also helps assess the consequences of using unnecessarily complex and larger models compared to the true model. We believe that the latter assessment is particularly important in a world where almost unlimited computing power is creating a natural tendency to overfit, without paying enough attention to the probable consequences of overfitting. Assuming that wind speed data can be generated using ARIMA$(3, 1, 1)$ processes, we analyze the inflation of forecast errors for several ARIMA models to study the influence of misspecifying the model and its parameters.

Thanks to this framework, we can establish and quantify the influence of incorrectly modelling WSFs on the forecast error. Such WSFs are relevant in the context of estimating WPFs, as they can be easily converted into forecasts of wind power by using the power curve of any particular wind turbine (Carrillo et al., 2013) or become an additional input for another WPF model (Bhaskar and Singh, 2012).

Additionally, having a deeper understanding on error modelling by means of such a decomposition is significant as forecast errors are used as markers to assess the performance of WPF models.

In essence, this chapter describes and investigates the main sources of forecast error. The work carried out is used as the baseline for the following chapter (and in fact the remainder of this thesis), where we investigate methods to quantify and minimize such errors. Particularly, an analysis of how these forecast errors are used to evaluate model performance in the context of WPF is performed in the next chapter.

Chapter 3

Performance evaluation metrics for wind power forecasting

As shown in the previous chapter, the estimation of the different sources of forecast error is a topic which has barely been explored before, despite the importance of forecast errors in the context of evaluating WPFs. Furthermore, even if many aspects of WPF have been thoroughly reviewed in the literature as described in section 1.4, none of the discussed studies goes into the limitations of evaluating WPFs, especially in terms of robustness, meaning that the forecast accuracy of a certain WPF model should not be affected by the wind power generation process, and therefore perform similarly for different scenarios of wind power generation (Chen, 1997). Even though performance evaluation metrics are introduced in the literature to analyze the forecast accuracy of WPF models, the robustness of the models is often disregarded. The main contributions of this chapter are the review of the main performance evaluation metrics used in the literature to assess WPFs models and the empirical evaluation of this feature through a case study using a set of WPF models and two different multi-step ahead forecast strategies. Furthermore, forecasts are estimated using two different resolutions (10 and 60 minutes) to examine the influence of data resolution in the evaluation of statistical models.

The remainder of this chapter is organized as follows. Section 3.1 presents the different performance evaluation methods for assessing WPFs. Section 3.2 presents different techniques used to estimate multi-step ahead forecasts that are considered

for longer prediction horizons. Section 3.3 presents a numerical study with data from Ireland. Section 3.4 includes the concluding remarks of this chapter.

3.1 Performance evaluation metrics of WPFs

The assessment of a forecasting model is a crucial step in its development to address its validity for estimating future values of wind power. The forecast accuracy of point estimates is evaluated measuring the discrepancy between the forecast and actual values through several criteria. The evaluation of probabilistic estimates is a more challenging task, as the forecast cannot be compared directly to the actual values, and several properties of the forecast have to be addressed to verify the forecast accuracy of the model.

3.1.1 Performance of deterministic estimates

Many performance evaluation methods are used in the literature to assess the accuracy of point estimates. The most common ones are shown down below.

- Mean absolute error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{3.1}$$

- Root mean square error ($RMSE$):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{3.2}$$

- Mean absolute percentage error ($MAPE$):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \cdot 100\% \tag{3.3}$$

- Standard deviation error (SDE):

$$SDE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\epsilon_i - \bar{\epsilon})^2} \tag{3.4}$$

- Bias:

$$BIAS = \frac{1}{N}\sum_{i=1}^{N} y_i - \hat{y}_i \tag{3.5}$$

- Index of Agreement (IA):

$$IA = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(|\hat{y}_i - \overline{y}| + |y_i - \overline{y}|)^2} \tag{3.6}$$

where N is the number of samples, y_i is the actual value, $\hat{y}_i$ the predicted value, $\overline{y}$ is the mean value of the real values, $\epsilon_i = y_i - \hat{y}_i$ is the prediction error (also known as *residual*), and $\overline{\epsilon}$ is the average value of the errors. Table 3.1 indicates the performance evaluation metrics used in the literature for wind power point estimates.

Eq. 3.1 shows the mean absolute error (MAE). It is defined as the average value of the predictions errors in absolute values. The root mean square error ($RMSE$), shown in Eq. 3.2, depicts the standard deviation of the residuals. Normalized versions of the MAE ($NMAE$) and the $RMSE$ ($NRMSE$) are commonly used in the literature as well. While both MAE and $RMSE$ are suitable indicators for assessing the performance of a model, the $RMSE$ should be preferred when the model errors follow a Gaussian distribution (Chai and Draxler, 2014).

Another statistical measure is the $MAPE$ (Eq. 3.3). It quantifies the accuracy as a percentage of the error. However, the $MAPE$ produces very large values when the actual values are close to zero and is undefined when the actual value is equal to zero. Alternative versions of the $MAPE$ have been proposed to prevent this shortcoming. For instance, the mean absolute scaled error ($MASE$) is an alternative defined as the MAE of the forecast values scaled by the MAE of the in-sample naïve forecast (Hyndman and Koehler, 2006). It is specially useful for wind power time series, as there are periods where a wind farm does not generate any power. The mean arctangent absolute percentage error ($MAAPE$) is another alternative option to the $MAPE$ (Kim and Kim, 2016). It transforms the $MAPE$ using the arctangent function. Its main advantage is the preservation of the characteristics of the $MAPE$

Table 3.1: Performance evaluation metrics for point estimates.

Reference	MAE	NMAE	RMSE	NRMSE	MAPE	SDE	Bias	IA	Others
Lee and Baldick (2013)	×		×		×	×	×		×
Haque et al. (2014)		×		×	×				
Buhan and Çadırcı (2015)		×							
Chitsaz et al. (2015)		×		×					
Dowell and Pinson (2015)	×		×						
Han et al. (2015)	×		×						
Li et al. (2015)				×	×				
Osório et al. (2015)		×		×	×				
Ozkan and Karagoz (2015)		×		×					
Yang et al. (2015)	×		×						
Abdoos (2016)		×		×					
Azimi et al. (2016)	×		×					×	
Heinermann and Kramer (2016)									×
Renani et al. (2016)	×				×	×			
Yan et al. (2016)		×		×					
Zhang and Wang (2016)			×						
Zhao et al. (2016)		×		×		×	×		×
Ziel et al. (2016)	×								
Chang et al. (2017)					×				×
Iversen et al. (2017)	×		×						
Jiang et al. (2017)	×	×	×						
Karakuş et al. (2017)				×	×		×		
Lahouar and Slama (2017)	×	×	×		×				×
Liu et al. (2017)		×		×	×				
Qureshi et al. (2017)	×		×			×			
Wang et al. (2017b)		×		×					×
Yuan et al. (2017)	×		×		×				
Zameer et al. (2017)	×		×			×			
Baxevani and Lenzi (2018)	×		×				×		
Zjavka and Mišák (2018)	×		×						×
He and Li (2018)			×		×				×
Naik et al. (2018b)	×		×		×				
Sharifian et al. (2018)			×		×				
Chen et al. (2019a)	×		×		×				
Çevik et al. (2019)	×	×	×	×					
Demolli et al. (2019)		×	×						×
Ding et al. (2019)	×		×						
Du et al. (2019)	×		×		×				
Hao and Tian (2019)	×		×		×			×	
Messner and Pinson (2019)	×		×				×		
Prósper et al. (2019)	×	×							×
Qureshi and Khan (2019)	×		×			×			
Wang et al. (2019a)	×		×		×				×
Wang et al. (2019b)	×		×		×			×	×
Abedinia et al. (2020a)		×		×	×				
Abedinia et al. (2020b)		×	×	×	×				×
Chen and Liu (2020)	×		×						×
Li et al. (2020d)		×		×	×				
Li et al. (2020e)		×		×	×				×
Liu and Duan (2020)	×		×					×	
Lu et al. (2020b)		×		×					×
Nazaré et al. (2020)					×				
Nielson et al. (2020)	×								×
Niu et al. (2020)				×	×				×
Wang et al. (2020a)	×		×						
Wang et al. (2020b)			×		×				
Wu et al. (2020)	×				×				×
Zhang et al. (2020a)		×		×					
Duan et al. (2021)	×		×					×	
Liu et al. (2021c)			×		×				
Putz et al. (2021)					×				×
Wu et al. (2021)	×		×	×	×				
Yildiz et al. (2021)	×		×						×
Khazaei et al. (2022)		×	×	×					×

while overcoming the limitations of the original definition.

The prediction error can be decomposed into the random error, which is inherently unpredictable, and the systematic error, which occurs due to inaccuracies in the system. The standard deviation of errors (Eq. 3.4) addresses the random component of the prediction error, whereas the bias (Eq. 3.5) deals with the systematic component (Madsen et al., 2005).

Lastly, another common metric in the literature to assess point estimates is the index of agreement (Eq. 3.6). It was originally proposed by Willmott (1981) and refined versions of this index have been developed since then (Willmott et al., 2012). It measures to which degree the predictions are error-free and takes values between zero when the adjustment between predictions and observations is null, and one when the predictions fully pair with the actual values.

Another alternative to examine model performance is to use probabilistic statistical measures that address not only the performance but also the complexity of the model. Some of them are the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), and Minimum Description Length (MDL).

In order to facilitate the comparison with benchmark models, the improvement of a technique is defined by means of the relation (Madsen et al., 2005):

$$\text{Improvement} = 1 - \frac{M}{M_{ref}} \tag{3.7}$$

where M is the value of the selected measure for a specific model, and M_{ref} is the value of the same measure for the benchmark model. Comparing different models become an issue as there is not a unified criterion for selecting benchmarks to evaluate them. Typically, models are compared to the persistence model, as it is a requirement to outperform it to be considered skillful, and state-of-the-art methods such as neural networks.

3.1.2 Performance of probabilistic estimates

Among the possible representations of probabilistic estimates, PIs are preferred due to their ease of operation. A PI is defined as an interval that gives the expectation of where a future value will fall with a specified probability. Therefore, the PI relies

on the significance level α. The probability that a future wind power output y_i lies within the PI is known as prediction interval nominal confidence ($PINC$):

$$PINC = 100(1 - \alpha)\% \tag{3.8}$$

Taking this into consideration, a PI for a future time step i and a significance level α is defined as:

$$\hat{I}_i^{\alpha} = \hat{U}_i^{\alpha} - \hat{L}_i^{\alpha} \tag{3.9}$$

where $\hat{U}_i^{\alpha}$ and $\hat{L}_i^{\alpha}$ are the upper and lower boundaries of the PI respectively. If we assume that model and data uncertainty are statistically independent, the variance of the forecast error results from the variance of model uncertainty and the variance of data noise (Quan et al., 2019):

$$\sigma_i^2 = \sigma_{\text{model}}^2 + \sigma_{\text{noise}}^2 \tag{3.10}$$

The variance of model uncertainty will be more influential under a small sample size of data, but can be reduced by increasing the sample size or better selecting the forecasting model following the ideas presented in Chapter 2. On the other hand, data noise (a component of data uncertainty) represents the part of the forecast error which cannot be reduced. By definition, the uncertainty of a prediction interval consider both model and data noise variance, as it represents a random variable not yet observed (De Gooijer and Hyndman, 2006). The concept of prediction interval should not be confused with a confidence interval, associated to a parameter (the true mean) instead of a random variable. Therefore, a confidence interval only takes into consideration the model uncertainty component in Eq. 3.10. In practice, these two sources of uncertainty are not to be necessarily identified if the PI is estimated (Hüllermeier and Waegeman, 2021), even if some authors (e.g. Wan et al. (2013b)) have proposed some methodologies to model these components separately. However, modellers should preferably focus on reducing model uncertainty to minimize the variance of the total forecast error, while characterizing both of them correctly to

avoid improper assessments of their influence in the prediction (Der Kiureghian and Ditlevsen, 2009).

The most common metrics defined for probabilistic estimates (in particular, for PIs) are shown down below.

- Prediction interval coverage probability ($PICP$):

$$PICP = \frac{1}{N}\sum_{i=1}^{N} c_i \tag{3.11}$$

where N is the number of samples and c_i is

$$c_i = \begin{cases} 1, & \text{if } y_i \in \hat{I}_i^{\alpha} \\ 0, & \text{otherwise} \end{cases} \tag{3.12}$$

- Prediction interval normalized average width ($PINAW$):

$$PINAW = \frac{1}{NR}\sum_{i=1}^{N} \hat{I}_i^{\alpha} \tag{3.13}$$

where R is the range of the target variable.

- Coverage width-based criterion (CWC):

$$CWC = PINAW\left[1 + \gamma(PICP)e^{-\eta(PICP-\mu)}\right] \tag{3.14}$$

where $\gamma(PICP)$ is a step function dependent on the values of PICP and μ:

$$\gamma(PICP) = \begin{cases} 0, & \text{if } PICP \geq \mu \\ 1, & \text{if } PICP < \mu \end{cases} \tag{3.15}$$

- Average coverage error (ACE):

$$ACE = PICP - PINC \tag{3.16}$$

- Interval sharpness (IS):

$$IS = \frac{1}{N}\sum_{i=1}^{N} b_i \tag{3.17}$$

where b_i is

$$b_i = \begin{cases} -2\alpha\hat{I}_i^{\alpha} - 4(\hat{L}_i^{\alpha} - y_i), & \text{if } y_i < \hat{L}_i^{\alpha} \\ -2\alpha\hat{I}_i^{\alpha}, & \text{if } y_i \in \hat{I}_i^{\alpha} \\ -2\alpha I_i^{\alpha} - 4(y_i - \hat{U}_i^{\alpha}), & \text{if } y_i > \hat{U}_i^{\alpha} \end{cases} \tag{3.18}$$

- Continuous ranked probability score ($CRPS$):

$$CRPS = \frac{1}{N}\sum_{i=1}^{N}\int_{0}^{P_{max}} [CDF_i - H(y - y_i)]^2 dy \tag{3.19}$$

where CDF_i is the cumulative form of the distribution and H is the Heaviside step function:

$$H = \begin{cases} 0, & \text{if } y < y_i \\ 1, & \text{otherwise} \end{cases} \tag{3.20}$$

- Skill Score (SC):

$$SC = \frac{1}{N}\sum_{i=1}^{N} SC_i = \frac{1}{N}\sum_{i=1}^{N}\left[\sum_{j=1}^{M}(\xi^{\alpha_j} - \alpha_j)(y_i - q_i^{\alpha_j})\right] \tag{3.21}$$

where SC_i is a set of quantiles on a single time step i, α_j is the quantile proportion, $q_i^{\alpha_j}$ is the quantile forecast, and ξ^{α_j} is an indicator variable denoted by

$$\xi^{\alpha_j} = \begin{cases} 1, & \text{if } y_i < q_i^{\alpha_j} \\ 0, & \text{otherwise} \end{cases} \tag{3.22}$$

Table 3.2 shows the application of these performance evaluation metrics for probabilistic estimates in the literature.

The two main features of a PI, the most common representation for probabilistic

estimates, are reliability and informativeness: a PI will be reliable when the actual wind power output falls within the interval, whereas it will be informative depending on its width. Ideally, the PI should be as narrow as possible to facilitate the decision-making process. The uncertainty of the prediction can be attributed to several

Table 3.2: Performance evaluation metrics for probabilistic estimates.

Reference	PICP	PINAW	CWC	ACE	IS	CRPS	SC	Others
Pinson and Madsen (2012)						×		
Khosravi and Nahavandi (2013)	×	×						
Khosravi et al. (2013)	×	×	×					
Quan et al. (2013)	×	×	×					
Wan et al. (2013b)	×			×	×			
Haque et al. (2014)							×	×
Zhang et al. (2014a)	×	×						
Alessandrini et al. (2015)						×		×
Dowell and Pinson (2015)						×		×
Kavousi-Fard et al. (2015)	×	×						
Yang et al. (2015)						×		
Gallego-Castillo et al. (2016)						×		
Wan et al. (2016)							×	×
Zhang and Wang (2016)								×
Zhang et al. (2016)	×	×						
Iversen et al. (2017)						×		
Lahouar and Slama (2017)								×
Shi et al. (2017)	×	×	×					×
Wang et al. (2017a)				×	×	×		
Wang et al. (2017b)	×	×	×	×	×			
Afshari-Igder et al. (2018)	×				×			
Baxevani and Lenzi (2018)						×		
He and Li (2018)	×	×						
Khorramdel et al. (2018)	×	×			×			
Lin et al. (2018b)						×		×
Mahmoud et al. (2018)	×	×			×			×
Wu et al. (2018b)	×	×	×					
Xie et al. (2018)	×						×	
Wang et al. (2019b)	×	×		×				×
Yuan et al. (2019)	×				×			×
Zhang et al. (2019b)	×						×	×
Zhao et al. (2019)	×	×		×	×		×	
Zou et al. (2019)	×	×	×					×
Ahmadpour and Farkoush (2020)		×		×	×			
Hu et al. (2020a)	×	×						×
Li et al. (2020a)	×	×	×					×
Liu and Duan (2020)	×	×	×					
Sun et al. (2020b)		×		×	×			
Yu et al. (2020)	×	×						
Zhang et al. (2020a)		×		×	×	×		
Gendeel et al. (2021)	×	×						×
Gu et al. (2021)	×							
He et al. (2021)	×	×	×					
Lv et al. (2021)	×							
Von Krannichfeldt et al. (2021)				×	×	×	×	
Dong et al. (2022)						×	×	
Wen et al. (2022)					×	×		×

sources. As indicated in section 2.1, one source is the model uncertainty (or epistemic uncertainty), which occurs due to a misspecification of the forecasting model or the parameters of the model. The other main source is the data uncertainty (or aleatoric uncertainty), which quantifies the inherent noise of the observations.

The *PICP* (Eq. 3.11) is a metric that measures exclusively the reliability of the PI. It accounts for the average of target values covered by the interval. Its counterpart in terms of width is the *PINAW* (Eq. 3.13). The *PICP* and the *PINAW* can be further merged with the *CWC* (Eq. 3.14). In this equation, η and γ are two controlling hyperparameters that determine how much invalid PIs are penalised. Alternative versions of the *CWC* have been introduced in the literature. For instance, the new *CWC* is proposed by Shi et al. (2017), which includes a new term designed to take better into consideration the information provided by the actual measurements. Another alternative is presented in Zhang et al. (2014a), which considers an additional function to account for those samples that lie beyond the interval.

Another parameter that describes the reliability of a PI is the *ACE* (Eq. 3.16). It is defined as the deviation between the *PICP* and the *PINC*. Smaller deviations indicate more reliable PIs. This metric provides additional information compared to the *PICP* since a larger *PICP* is not necessarily better for a given *PINC* (Pinson and Kariniotakis, 2010).

The interval sharpness (also known as Winkler score) (Eq. 3.17) evaluates the PI in terms of its width (Winkler, 1972). Narrower intervals are rewarded by this metric, whereas those PIs where the observations do not lie inside are penalised.

The *CRPS* (Eq. 3.19) is a global criterion as it assesses both features simultaneously. This metric is equivalent to the *MAE* when a forecast is generated as a point estimate. It differs from other metrics as the *CRPS* assesses cumulative distribution functions. Lower scores of the *CRPS* mean a better performance of the model.

Equation 3.21 denotes the scoring rule proposed by Pinson et al. (2007b) when probabilistic forecasts are estimated by non-parametric models and are represented by a set of quantile forecasts. Its orientation is positive and a value of zero represents a perfect forecast.

Comparison with benchmark models for probabilistic estimates can be performed

by using the same relation presented for point estimates in Eq. 3.7.

3.2 Prediction horizon for WPFs

The prediction horizon is one of the main aspects to consider when a WPF model is developed. Very-short term forecasts consider predictions up to 30 minutes ahead. The main applications of these forecasts are wind farm control and operation of reserves (Pinson, 2012). Short-term forecasts take into account predictions from hours to a few days and are an indispensable tool for power system management and energy trading. The use of additional exogenous inputs such as meteorological data may be considered to train a statistical model for this horizon as the dynamics of wind power generation become significant and could potentially lead to a better performance of the model. Nonetheless, the use of these data could increase the computational complexity of the model (Yan et al., 2013) and these datasets are associated with their own prediction error that will affect the prediction accuracy (Zhang et al., 2020b). Alternatively, physical models are used for short-term forecasting as well in the absence of an actual wind farm, although they are more computationally expensive compared to statistical models. Longer-term forecasts usually make use of physical methods and are used to make decisions on unit commitment or maintenance scheduling (Soman et al., 2010).

For longer prediction horizons from very short-term and short-term statistical models, it is necessary to consider multi-step ahead forecasts. A multi-step ahead forecast estimates the next H steps $[y_{t+1}, \ldots, y_{t+H}]$ of a time series. There are several strategies to approach this matter (Taieb et al., 2012; Wang et al., 2016a): the recursive, the MIMO (Multiple-input Multiple-output) and the direct strategies are the most commonly used approaches to estimate multi-step ahead predictions.

In the recursive strategy (Taieb et al., 2012), also known as iterated or multi-state strategy, the model is trained to compute one-step ahead forecasts. Afterwards, the next steps are predicted iteratively using the previous one-step ahead forecasts as inputs. This strategy is sensitive to larger prediction horizons, as the errors of the

predictions accumulate for every iteration.

$$\hat{y}_{t+h} = \begin{cases} f(y_t, \ldots, y_{t-d+1}) & \text{if } h = 1, \\ f(\hat{y}_{t+h-1}, \ldots, \hat{y}_{t+1}, y_t, \ldots, y_{t-d+h}) & \text{if } h \in \{2, \ldots, d\} \\ f(\hat{y}_{t+h-1}, \ldots, \hat{y}_{t+h-d}) & \text{if } h \in \{d+1, \ldots, H\} \end{cases} \tag{3.23}$$

where d denotes the number of steps used in the input set.

The MIMO strategy (Taieb et al., 2010) produces a vector with the whole sequence of outputs training a single model.

$$[\hat{y}_{t+H}, \ldots, \hat{y}_{t+1}] = f(y_t, \ldots, y_{t-d+1}), \tag{3.24}$$

The direct strategy (Taieb et al., 2012) consists of training H different models independently, one for each horizon. While it prevents the accumulation of errors, it neglects the dependencies between the H forecasts and it carries a larger computational cost to train every model separately.

$$\hat{y}_{t+h} = f_h(y_t, \ldots, y_{t-d+1}), \quad h \in \{1, \ldots, H\} \tag{3.25}$$

Further combinations of these models lead to other strategies. For instance, the DirRec strategy (Sorjamaa and Lendasse, 2006) combines elements from the recursive and direct approaches, and the DIRMO strategy (Taieb et al., 2009) presents a trade-off between the direct and the MIMO strategies.

3.3 Numerical study

In this section, a numerical study is performed to investigate the features of the performance evaluation metrics for WPF models considering data from a single wind farm, modelled using a set of decomposition-based hybrid models (Qian et al., 2019; Liu and Chen, 2019). First, we introduce succinctly the data and forecasting models used, followed by a discussion of the features of the metrics over the testing set.

3.3.1 Data

The dataset used for this numerical study contains measurements from a wind farm located in Ireland. Measurements are collected every 10 minutes between January 2017 and December 2017 (Figure 3.1). As shown in this figure, wind power generation shows a large variability as it is influenced by wind and other meteorological variables, as well as human activities such as maintenance operations. Less than 1% of the values are missing and have been reconstructed considering the previous and posterior values. Simulations are run for one wind turbine with the reconstructed dataset with a temporal 10-min resolution and a resampled dataset with a 1-h resolution. The dataset is divided into training, validation and testing sets to train the model and test the accuracy of the forecasts provided by the models. Table 3.3 shows the number of samples for each set considering the low-resolution (1-h) and high-resolution (10-min) data and the summary statistics for both sets of data. The location of the wind farm is not disclosed due to confidentiality reasons. A more detailed description of this dataset is found in the next chapter (section 4.2.3).

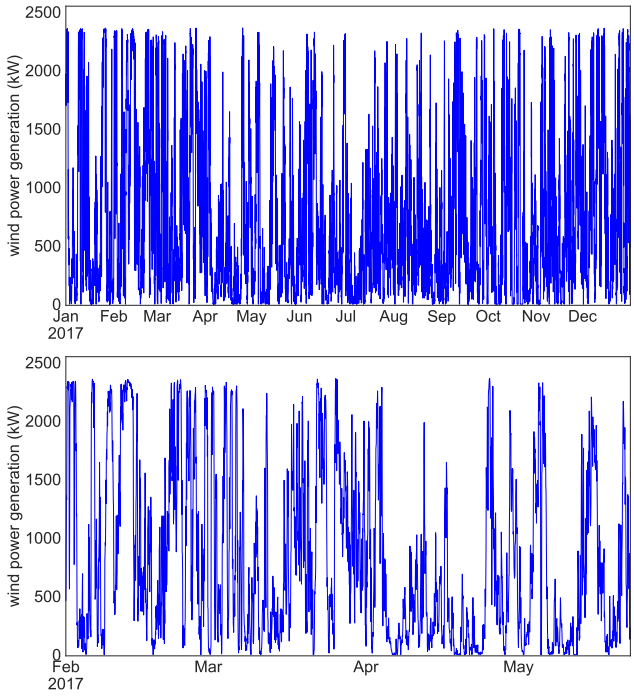

Figure 3.1: Historical wind power generation during the year 2017 (left) and a sample of the wind power generation time series from February to May (right). Data are shown with a temporal resolution of 60 minutes.

Table 3.3: Sample size and summary statistics of the datasets.

	Low-resolution set	High-resolution set
Resolution	60-min	10-min
Train	7296	43776
Validation	1080	6480
Test	384	2304
Mean (kW)	726.49	726.49
Std (kW)	696.15	715.30
Min (kW)	0	0
Q1 (kW)	157.13	140
Q2 (kW)	478	462
Q3 (kW)	1134	1151
Max (kW)	2364.3	2365

Table 3.4: WPF models used for the numerical study.

Decomposition technique	Training model	Multi-step strategy
VMD	FFNN	MIMO
VMD	FFNN	Recursive
EEMD	FFNN	MIMO
EEMD	FFNN	Recursive
-	FFNN	MIMO
-	FFNN	Recursive

3.3.2 Experimental design

As stated in section 1.4.1.4, decomposition-based hybrid models are a family of statistical models that can contribute to a better forecasting accuracy for short-term WPF (Qian et al., 2019). Some of these models are used in this study to analyze the performance evaluation metrics. In particular, two techniques are applied to decompose the wind power time series: EEMD (Wu and Huang, 2009) and VMD (Dragomiretskiy and Zosso, 2013). Additionally, two multi-step forecast strategies (MIMO and Recursive) are implemented. In total, six forecasting models are employed (Table 3.4), including two FFNN models where the wind power time series is not decomposed.

Point estimates are obtained using only one output for every FFNN, whereas the boundaries of PIs are estimated by quantile regression, using an asymmetric loss function (also known as pinball loss function) that depends on the required quantiles τ (Koenker and Bassett Jr, 1978; Cannon, 2011). This technique has been chosen for this numerical study to provide interval estimates as it is a well-established technique in the field of WPF (Bremnes, 2004; Nielsen et al., 2006). The final prediction is

comprised of a single output for every step for point estimates and the lower and upper boundaries of the interval in the case of PIs.

A detailed description of this modelling methodology is provided in the next chapter (decomposition algorithms are explained in section 4.2.1, whereas QR-based PIs are described in section 4.2.2).

3.3.3 Results

The selected models (Table 3.4) are used to obtain forecast estimates, that will allow to produce numerical values for the performance evaluation metrics. In this study, the metrics not only evaluate the general forecast accuracy of the models, as usually considered during the model development stage, but also assess them in terms of time resolution, robustness, and prediction horizon length. In the case of probabilistic estimates, PIs are chosen to present the uncertainty of the forecast as they are the most widespread representation. Hence the CRPS and SC are not considered in the study, as they are used to assess other representations of probabilistic estimates.

3.3.3.1 Deterministic predictions

Forecasts are estimated 6-h and 24-h ahead. These horizons are important for activities such as energy trading, where an initial forecast is usually provided 24-h ahead and subsequently corrected between 6 and 8 hours ahead. Table 3.5 shows the results of the performance evaluation measurements for 6-h and 24-h ahead forecasts.

The values of *MAE*, *RMSE*, *BIAS*, and *SDE* are normalized by the capacity of the wind turbine to facilitate their understanding and the assessment of the model errors. The values of *MAPE* and *IA* are not normalized, as *MAPE* is established by definition as a percentage, and *IA* only takes values between zero and one. The lower scores for *NMAE* and *NRMSE* indicate that overall the VMD-FFNN MIMO model produces better forecasts considering every source of error. Additionally, the better scores for the *NBIAS* and *NSDE* indicate that this model deals better with the systematic and random error separately. The numerical values of the metrics are larger for 24-h ahead forecasts, indicating that the forecast accuracy is lower. Values close to one for the *IA*, such as the VMD-FFNN MIMO models for both

Table 3.5: Results for point estimates.

Forecast horizon: 6-h ahead							
Low-resolution dataset (1-h resolution)							
Method	Strategy	NMAE (%)	NRMSE (%)	MAPE (%)	NBias (%)	NSDE (%)	IA
EEMD-FFNN	MIMO	10.40	14.33	29.80	4.12	13.72	0.936
EEMD-FFNN	Recursive	10.80	14.66	31.45	-2.09	14.51	0.935
VMD-FFNN	MIMO	**1.77**	**2.440**	**12.05**	**-0.58**	**2.37**	**0.998**
VMD-FFNN	Recursive	4.51	5.45	20.25	4.38	3.23	0.992
FFNN	MIMO	21.08	28.41	44.91	5.98	27.97	0.706
FFNN	Recursive	21.87	30.12	48.61	2.35	30.02	0.730
High-resolution dataset (10-min resolution)							
Method	Strategy	NMAE (%)	NRMSE (%)	MAPE (%)	NBias (%)	NSDE (%)	IA
EEMD-FFNN	MIMO	11.16	15.50	28.69	**1.03**	15.46	0.929
EEMD-FFNN	Recursive	24.91	33.02	56.97	23.51	23.19	0.675
VMD-FFNN	MIMO	**8.27**	**10.77**	**25.19**	-3.61	**10.14**	**0.969**
VMD-FFNN	Recursive	13.65	19.02	35.50	9.49	16.48	0.890
FFNN	MIMO	22.17	30.55	46.84	7.48	29.62	0.689
FFNN	Recursive	34.26	46.11	74.19	34.05	31.10	0.478
Forecast horizon: 24-h ahead							
Low-resolution dataset (1-h resolution)							
Method	Strategy	NMAE (%)	NRMSE (%)	MAPE (%)	NBias (%)	NSDE (%)	IA
EEMD-FFNN	MIMO	18.18	22.33	35.84	**1.54**	22.27	0.765
EEMD-FFNN	Recursive	24.69	33.03	55.89	20.71	25.73	0.582
VMD-FFNN	MIMO	**5.95**	**7.57**	**18.99**	1.66	7.38	**0.982**
VMD-FFNN	Recursive	8.81	10.48	25.78	-7.68	**7.13**	0.969
FFNN	MIMO	26.66	34.93	49.68	12.66	32.56	0.379
FFNN	Recursive	28.74	36.92	58.63	12.50	34.74	0.434
High-resolution dataset (10-min resolution)							
Method	Strategy	NMAE (%)	NRMSE (%)	MAPE (%)	NBias (%)	NSDE (%)	IA
EEMD-FFNN	MIMO	**20.28**	**25.43**	38.52	**1.06**	**25.32**	0.689
EEMD-FFNN	Recursive	43.49	53.73	**36.47**	-39.05	36.91	0.522
VMD-FFNN	MIMO	20.39	25.94	40.41	4.90	25.48	**0.713**
VMD-FFNN	Recursive	29.40	35.77	38.97	-11.20	33.97	0.589
FFNN	MIMO	28.62	38.25	56.42	17.63	33.95	0.389
FFNN	Recursive	63.79	71.43	31.31	-63.79	32.14	0.411

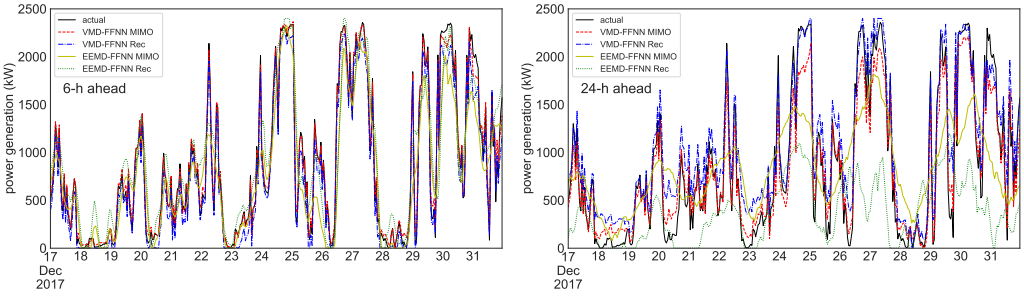

Figure 3.2: Performance of the selected models for 6-h and 24-h ahead point estimates. Data are shown with a temporal (60-min) resolution.

6-h and 24-h ahead forecasts using the low-resolution dataset, indicate that the forecasts have a low degree of error. Compared to the rest of the metrics, *MAPE* shows unsteady values that are not consistent with the rest of metrics evaluating the overall forecast accuracy (*NMAE*, *NRMSE*, and *IA*). As observed in Figure 3.2, the 6-h ahead forecasts are more accurate than 24-h ahead forecasts, as the forecast accuracy is lower for larger prediction horizons. Figure 3.3 shows the forecast errors in the 1-h resolution testing set. In both scenarios, the prediction error shows less variability when the VMD technique is applied to decompose the wind power time series. Spikes are visible when using EEMD, as the prediction error is larger due to the lower ability of EEMD to predict sudden changes in wind power correctly.

In terms of the effect of time resolution on the forecasts, the larger volatility existing in higher resolution wind power data reduces the forecast accuracy of WPF models. Figure 3.4 depicts the distribution of the forecast errors for every scenario. The larger spread observed for the high-resolution data comes mostly from the volatility of these data, as their intrinsic characteristics are harder to capture for the model, and it is affected as well by the number of steps ahead to predict as the errors accumulate for every step. Low-resolution forecast errors show a normal distribution in most of the cases, except for the EEMD models for 24-h ahead, that are slightly skewed to the right. Furthermore, the 6-h ahead predictions present a few outliers, as most of the residuals are centred around the median. The prediction errors from the high-resolution dataset have different patterns. A right-skewed distribution is observed in the EEMD-FFNN Recursive and the VMD-FFNN Recursive models for 6-h ahead forecasts, and in the EEMD-FFNN MIMO, VMD-FFNN MIMO and VMD-FFNN Recursive models for 24-h ahead forecasts. A normal dis-

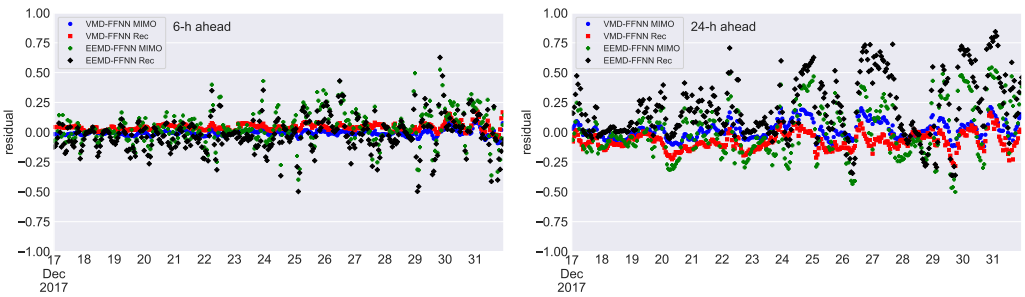

Figure 3.3: Errors for 6-h and 24-h ahead point estimates. Data are shown with a temporal (60-min) resolution.

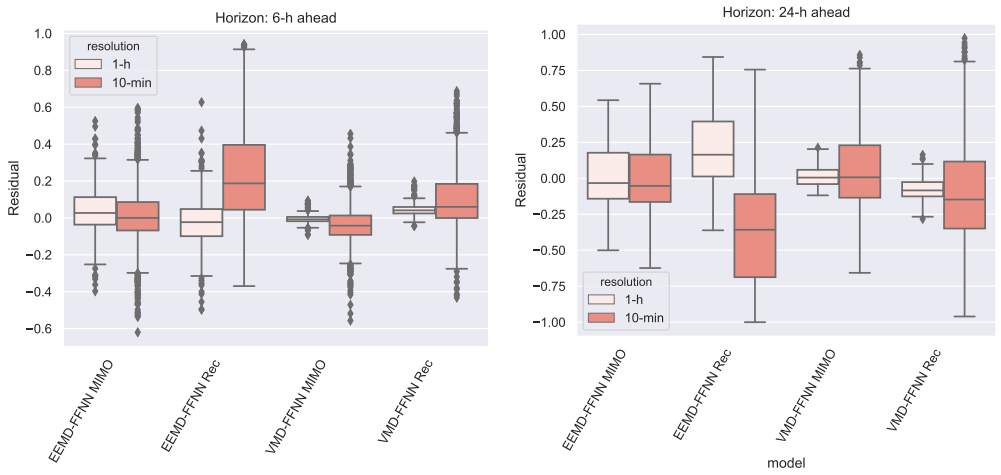

Figure 3.4: Distribution of forecast errors for 6-h and 24-h ahead predictions.

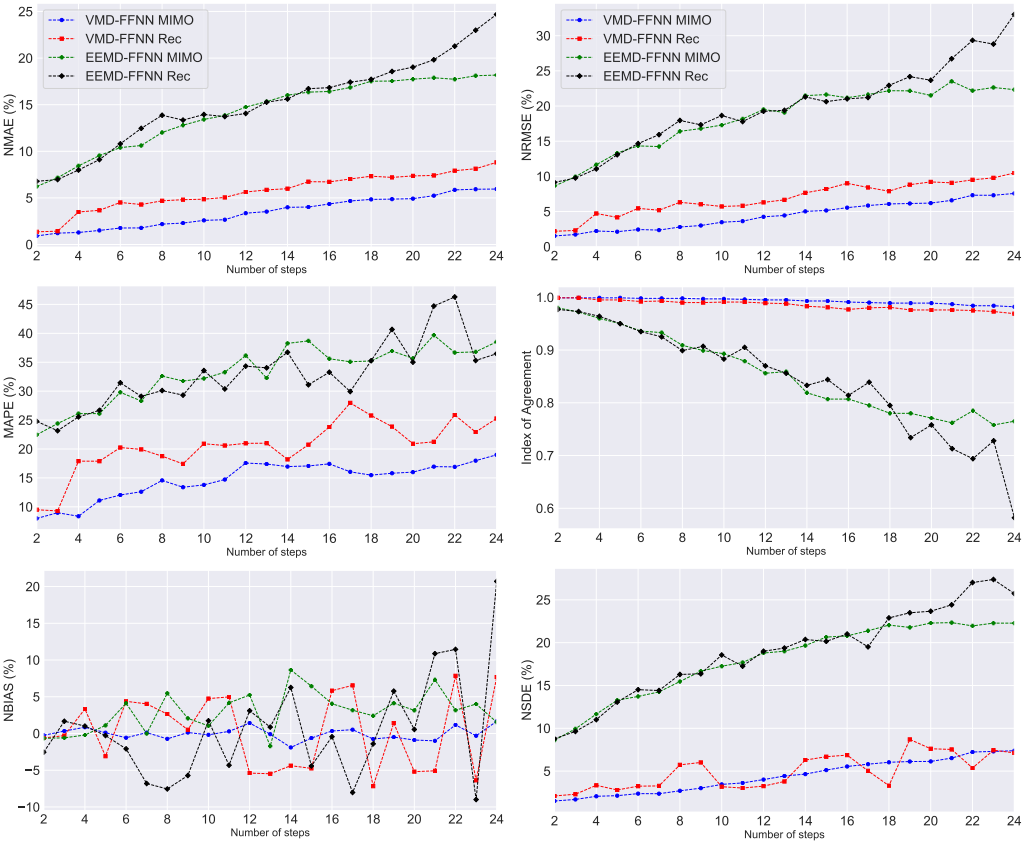

Figure 3.5: Performance evaluation metrics for point estimates with respect to the number of steps ahead (low resolution dataset).

tribution is visible for the EEMD-FFNN MIMO and VMD-FFNN MIMO models, although there are a considerable amount of outliers at the end of both tails. Only the EEMD-FFNN Recursive model shows a skewness to the left.

The evolution of the performance evaluation metrics with respect to the number of steps ahead predicted for the low-resolution dataset is shown in Figure 3.5. As

every step represents a 60-min interval, the number of steps is equivalent to the hours ahead predicted. As expected, the *NMAE* and the *NRMSE* values increase as the number of steps increases, since the quality of the predictions shrinks with the prediction horizon. The same expected behavior is observed for the *IA*, indicating a lower performance for larger prediction horizons. The *NBIAS* is not affected by the prediction horizon, producing approximately regular values for the VMD-FFNN MIMO model and high variance for the other three models. The *SDE* tends to increase with larger prediction horizons. *MAPE* scores are higher for a larger number of steps, although this relationship is not linear.

Considering the techniques and available dataset, the VMD-FFNN MIMO model performs overall better than the rest of the models, and is used hereafter to discuss its robustness by analysing the daily values of the metrics obtained with the forecasts provided by this model (Figure 3.6). Ideally, the forecast accuracy of the model should be as independent as possible from the data used to benchmark its validity, and therefore the numerical values obtained by the metrics should be similar for every subinterval. This performance is achieved for *NMAE* and *NRMSE* for three of the scenarios: 6-h ahead forecasts (both low- and high-resolution sets) and 24-h ahead

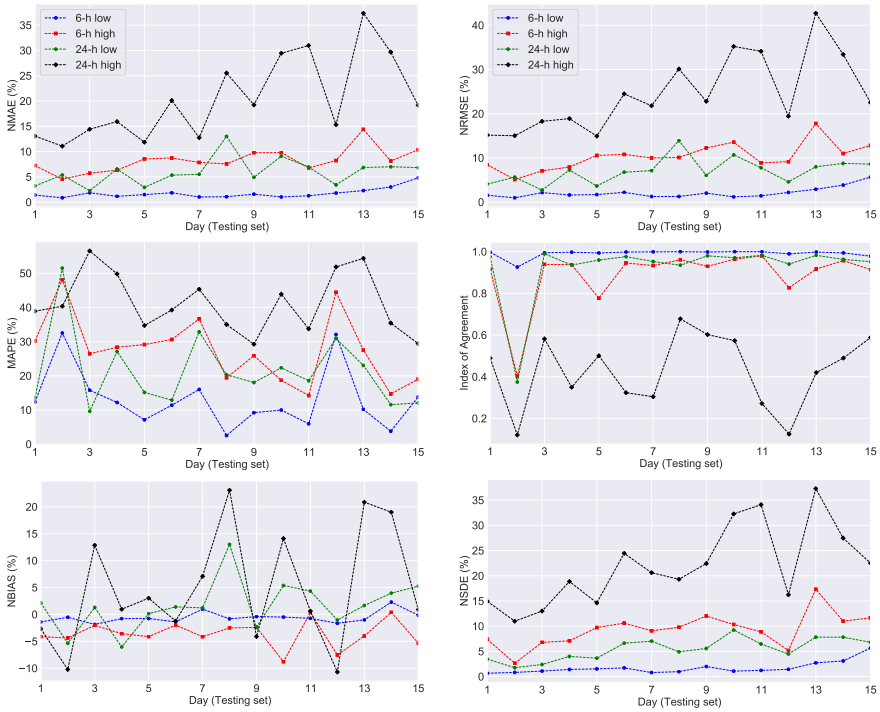

Figure 3.6: Daily performance of metrics for point estimates (VMD-FFNN MIMO model).

forecasts with the low-resolution set. The lack of robustness for the other case (24-h ahead forecasts in the high-resolution set) results from a combination of several aspects: the model itself, the dataset, the prediction horizon length and the time resolution of the data. Therefore, the model should be further calibrated to verify if the model is able to produce accurate forecasts for this case. The daily-averaged *NBIAS* shows low variability in the whole testing set for 6-h ahead forecasts, whereas they do not follow any pattern for 24-h ahead forecasts. The *NSDE* has a similar behavior as the *NMAE* and *NRMSE*, producing robust outcomes for every case but 24-h ahead forecasts in the high-resolution dataset. The *IA* produces similar scores for 6-h ahead predictions in the low-resolution dataset. The results are quite steady for 6-h ahead (high-resolution set) and 24-h ahead (low-resolution set) forecasts, except for day 2, where there is a sudden drop in the performance of the metric. *MAPE* shows a great variability in its values, indicating a large sensitivity to changes of wind power output.

The performance evaluation metrics provide not only information about the general performance of the models in terms of accuracy, but also can act as a tool to analyze the effects of aspects such as the robustness of the models under different conditions. The VMD-FFNN MIMO model show a robust behavior when the forecast accuracy is high since it shows low variations when calculating their values for different periods in the testing set. As expected, the performance evaluation metrics show a decreasing forecast accuracy while the prediction horizon increases. However, the *NBIAS* does not seem to be affected by the prediction horizon and keeps values around zero for the VMD-FFNN MIMO model. Additionally, the *MAPE* values are not in line with the scores obtained by the *NMAE*, the *NRMSE*, and the *IA* in terms of accuracy and consequently its use is not recommended.

3.3.3.2 Prediction intervals

PIs are estimated 6-h ahead for three confidence levels: 99%, 95%, and 80%. The results are shown in Table 3.6. As done previously for the point estimates, two basic FFNN models are used to benchmark the skill of the decomposition-based hybrid models.

Table 3.6: Results for 6-h ahead prediction intervals.

Confidence level: 99 %: Low-resolution dataset (1-h resolution)						
Method	Strategy	PICP (%)	PINAW (%)	CWC	ACE	IS
EEMD-FFNN	MIMO	**100**	99.66	0.997	**0.01**	-47.84
EEMD-FFNN	Recursive	95	76.77	6.440	-0.04	-78.45
VMD-FFNN	MIMO	**100**	77.95	0.780	**0.01**	-37.42
VMD-FFNN	Recursive	**100**	**60.05**	**0.601**	**0.01**	**-28.82**
FFNN	MIMO	93.61	92.68	14.641	-0.05	-55.95
FFNN	Recursive	63.89	80.69	3.3e7	-0.35	-394.49
Confidence level: 99 %: High-resolution dataset (10-min resolution)						
Method	Strategy	PICP (%)	PINAW (%)	CWC	ACE	IS
EEMD-FFNN	MIMO	**100**	96.88	0.969	**0.01**	-46.50
EEMD-FFNN	Recursive	96.25	81.20	1.881	-0.03	-80.02
VMD-FFNN	MIMO	**100**	90.32	**0.903**	**0.01**	**-43.35**
VMD-FFNN	Recursive	88.98	**40.14**	1.495	-0.1	-92.97
FFNN	MIMO	99.8	98.77	0.988	0.01	-47.51
FFNN	Recursive	41.48	67.26	2.1e12	-0.57	-1096.71
Confidence level: 95 %: Low-resolution dataset (1-h resolution)						
Method	Strategy	PICP (%)	PINAW (%)	CWC	ACE	IS
EEMD-FFNN	MIMO	**100**	81.96	0.820	**0.05**	-196.70
EEMD-FFNN	Recursive	88.61	41.29	10.486	-0.06	-184.08
VMD-FFNN	MIMO	**100**	39.75	0.397	**0.05**	-95.39
VMD-FFNN	Recursive	**100**	**31.09**	**0.311**	**0.05**	**-74.61**
FFNN	MIMO	93.06	83.43	3.04	-0.01	-263.53
FFNN	Recursive	43.89	50.51	6.3e10	-0.5	-723.95
Confidence level: 95 %: High-resolution dataset (10-min resolution)						
Method	Strategy	PICP (%)	PINAW (%)	CWC	ACE	IS
EEMD-FFNN	MIMO	**100**	84.76	0.848	**0.05**	-203.42
EEMD-FFNN	Recursive	45.74	32.07	44.521	-0.49	-850.53
VMD-FFNN	MIMO	**100**	68.49	**0.685**	**0.05**	**-164.37**
VMD-FFNN	Recursive	57.78	**26.06**	11.038	-0.37	-346.90
FFNN	MIMO	90.97	85.55	7.265	-0.04	-269.465
FFNN	Recursive	27.78	44.97	1.8e14	-0.67	-1373.64
Confidence level: 80 %: Low-resolution dataset (1-h resolution)						
Method	Strategy	PICP (%)	PINAW (%)	CWC	ACE	IS
EEMD-FFNN	MIMO	96.67	51.59	0.516	**0.17**	-528.03
EEMD-FFNN	Recursive	56.11	18.44	28395.9	-0.24	-526.32
VMD-FFNN	MIMO	**100**	21.34	0.213	0.2	-204.90
VMD-FFNN	Recursive	96.67	**14.26**	**0.143**	**0.17**	**-146.72**
FFNN	MIMO	71.94	56.99	32.57	-0.08	-854.34
FFNN	Recursive	33.61	32.82	3.9e9	-0.46	-1234.63
Confidence level: 80 %: High-resolution dataset (10-min resolution)						
Method	Strategy	PICP (%)	PINAW (%)	CWC	ACE	IS
EEMD-FFNN	MIMO	99.44	58.90	0.589	**0.19**	-567.42
EEMD-FFNN	Recursive	15.88	15.94	81.20	-0.64	-1477.55
VMD-FFNN	MIMO	**99.95**	47.93	**0.479**	**0.19**	**-460.54**
VMD-FFNN	Recursive	36.76	**11.50**	8.798	-0.43	-511.88
FFNN	MIMO	77.18	59.92	3.059	-0.03	-905.04
FFNN	Recursive	21.67	16.84	7.8e11	-0.58	-1751.05

The *PICP* and *ACE* measure exclusively the coverage of the PI. Considering that, these metrics indicate that the VMD-FFNN MIMO model presents the best results in terms of coverage, meaning that a larger number of observations fall within the interval. The *PINAW* and the *IS* quantify the width of the interval. The first metric only considers the width of the interval in every time step, while the *IS* penalises incorrect intervals as shown in Eq. 3.17. In this study, narrower intervals are usually built when the recursive strategy is applied, at the expense of reducing the coverage of the interval. Therefore, better scores for *PINAW* are obtained for narrower intervals (60.05%, 31.09% and 14.26% for the VMD-FFNN Recursive model given 99%, 95%, and 80% confidence levels respectively). Furthermore, since the coverage of the intervals given by this model is high for all confidence levels, it also has the best scores for the *IS*. The *CWC* takes into account both *PICP* and *PINAW* to consider both features simultaneously. As stated in Eq. 3.15, those intervals where the *PICP* is lower than μ will be penalised and will produce equal measurements as the *PINAW*. Looking at the metrics' scores, the decomposition-based hybrid models present more skilled intervals that the FFNN models. Additionally, even if the EEMD-FFNN MIMO model has a high *PICP*, the intervals are not informative as they are also very wide, and therefore not useful for applications in the industry.

For the high-resolution dataset, the metrics also provide information regarding the multi-step ahead forecast strategy used. For instance, even if the *PINAW* is only 26.06% for the VMD-FFNN Recursive model compared to a 68.49% when the MIMO strategy is used (95% confidence level), the interval covers only a 57.78% of the data points of the testing set. The *IS* provides additional details in terms of the coverage-width relation: the VMD-FFNN MIMO model scores better in every scenario, meaning that this relation is balanced, as fewer intervals are penalised by not covering the actual values. In terms of skill, the behavior is identical as the observed for the low-resolution dataset.

Figure 3.7 displays the PIs obtained for the four models using the low-resolution and high-resolution set respectively. As the number of steps to predict is lower for the low-resolution dataset, intervals are more accurate and the boundaries are closer to the observations. In the low-resolution dataset, all models can produce reliable

intervals, although the intervals are too wide for some of the models.

The evolution of the PI performance metrics with respect to the prediction horizon length is shown in Figure 3.8. The coverage of the PI decreases with the number of steps only for EEMD-FFNN Recursive model since the rest maintain all the values within the interval. This behavior is replicated for the *ACE*, where the value decreases with a larger number of steps. The *PINAW* is barely influenced by the increase of the prediction horizon and it does not reveal high changes except for one of the models. Lastly, the *IS* decreases with the forecast horizon, indicating a lower forecast accuracy for longer horizons.

Figure 3.9 shows the daily performance of the interval metrics for both sets. The

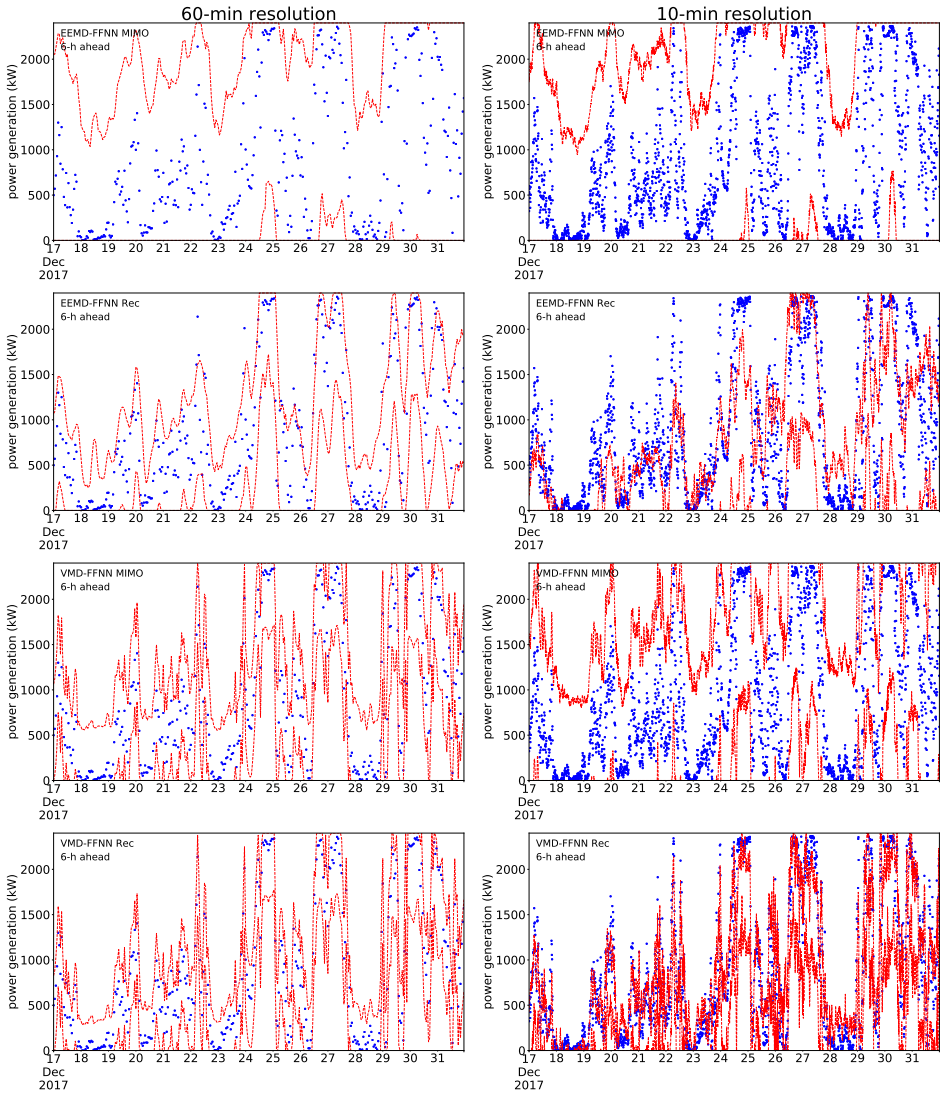

Figure 3.7: 6-h ahead prediction intervals (95% confidence level) using the low-resolution (left) and high-resolution dataset (right).

PICP has reached its maximum value for some of the models, meaning that all the observations fall inside the interval. Therefore, the daily-averaged values of the *PICP* indicate that the models are robust in these cases. Otherwise, the metric shows the sensitivity of the models to changes in the data as the *PICP* acts as a control depending on the percentage of values within the PI. The second metric (*PINAW*) shows a more robust performance for the models in the testing set, although it generates narrower intervals in days 2, 7 and 12 for every model. This behavior takes place with low wind power generation, so these values seem justified as the interval will not grow any longer in its lower boundary. The *CWC* shows spikes whenever the *PICP* is lower than the parameter μ (set to the confidence level). It provides a control system to know the coverage of the interval and to what degree is good, as a larger *CWC* implies a greater penalisation by the metric. However, as the *CWC* is very sensitive to this penalisation, evaluating the robustness of the models with this metric is not advised. The *ACE* provides similar results than the

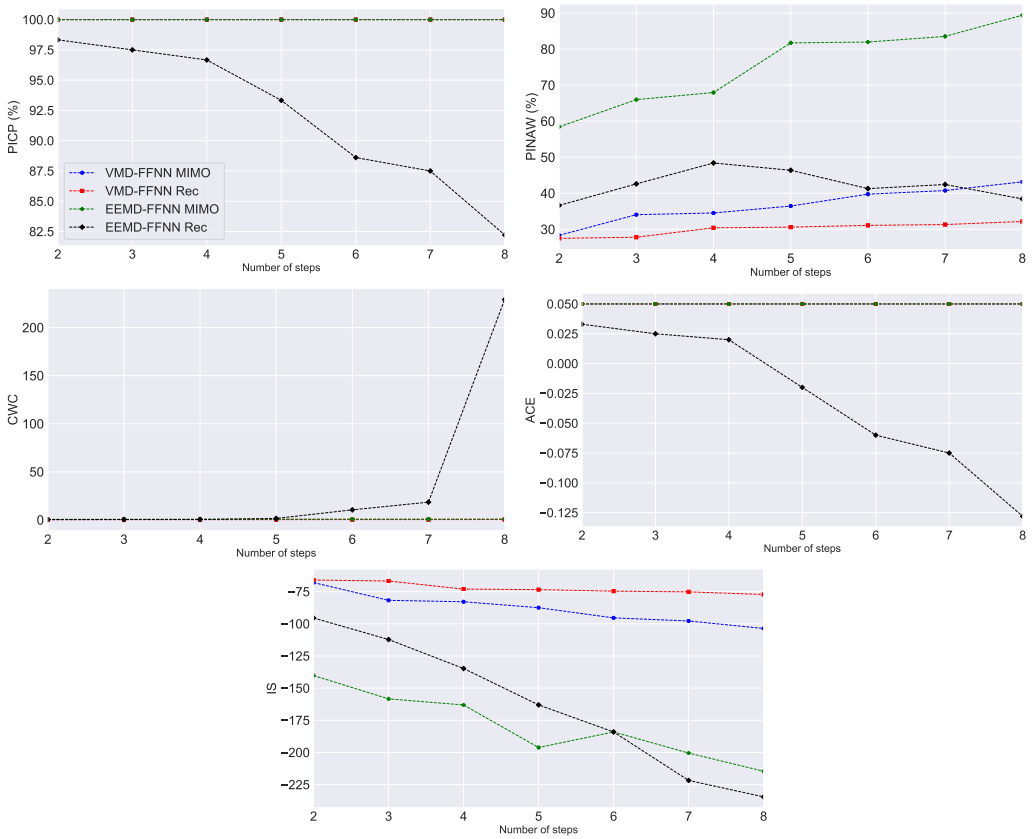

Figure 3.8: Performance evaluation for interval metrics with respect to the number of steps ahead (low resolution dataset).

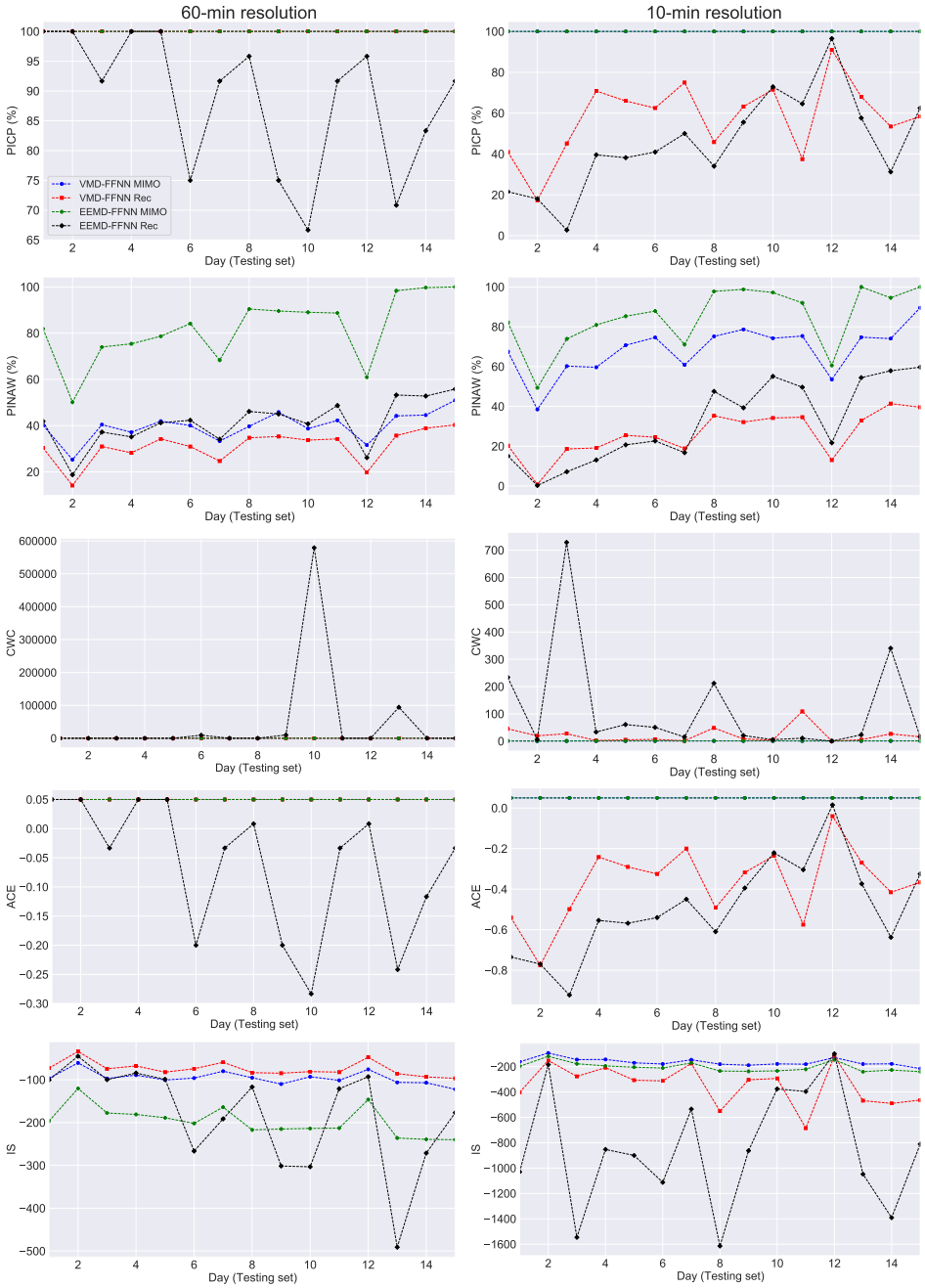

Figure 3.9: Daily performance of metrics for PIs (95% confidence level).

PICP, only it takes into account the significance level. For this reason, except for exceptional cases, it is enough to choose only one of them to analyze the robustness of the interval. The daily-averaged *IS* produces robust values with low variability for the VMD-FFNN models in the low-resolution dataset. Additionally, the *IS* provides supplementary information about the forecast accuracy of the interval, since the prediction error by itself accounts only for the size of the PI. The more negative is the IS for a time step, the further away the actual value is from the interval.

The *PICP* and *PINAW* provide direct knowledge in terms of coverage and width. The *CWC* is highly sensitive to the coverage of the interval and the confidence level, and therefore not suitable to evaluate the robustness of the models. The *ACE* provides very similar information as the *PICP*, so its assessment is not necessary if the *PICP* is already estimated. The *IS* allows to determine reliably the robustness of a model, and provides information about the width interval while penalising incorrect PIs in terms of coverage.

3.4 Conclusions

Performance evaluation metrics capture different aspects of model performance. Most of the performance evaluation metrics identified for the assessment of point estimates analyze all sources of error together (*MAE*, *RMSE*, *MAPE*, *IA*), while others evaluate a specific source of error such as the *BIAS*, that accounts for the systematic component of error, or the *SDE*, where only the random error is analyzed. Probabilistic estimates account for both accuracy and precision, therefore they are preferred for model comparability. These metrics evaluate the coverage provided by the interval, such as the *PICP*, or the width, such as the *PINAW*, whereas the *IS* provides additional information in terms of the overall quality of the interval. These three metrics give enough information to address the forecast accuracy of the interval. On the other hand, the *ACE* does not provide any additional information if the *PICP* is already estimated, consequently is not deemed necessary to evaluate the coverage of the interval. The *CWC* is highly sensitive to both the tuning parameters and the nature of the training set. Therefore, this metric is recommended as a parameter to train the data in methodologies such as the LUBE method, but not to evaluate the accuracy of a forecasting model.

The different performance evaluation metrics are also applied to evaluate the robustness of the models over the testing set, while simultaneously considering aspects of relevance such as the prediction horizon and the time resolution of the data, since intra-hour wind data shows higher volatility than hourly averaged data. In addition to that, the forecasting models for higher resolution sets can be further calibrated to provide more accurate forecasts.

All these insights, as well as the use of unified metrics to evaluate WPF models, are further expanded in the next chapter to propose standards to assess such models. This is achieved by developing a novel benchmarking framework to evaluate statistical WPF models, taking into consideration the insights developed in this chapter combined with the concepts of fair and representative evaluation of WPFs as suggested by the IEA Wind Task 36 guidelines.

Chapter 4

A benchmarking framework to evaluate wind power forecasting models

In the previous chapter, we have identified the main performance evaluation metrics used for both point estimates and prediction intervals. Such a variety of metrics is one of the reasons that lead to issues with respect to the standardization of WPF model evaluation. Together with an unified selection of performance evaluation metrics, other elements must be standardized throughout the model evaluation stage in order to assess WPF models in a fair and representative manner, such as data, time resolution, and prediction horizon (González Sopeña et al., 2022c). These ideas are fleshed out in this chapter.

4.1 Introduction

As found in the literature review conducted in section 1.4, one of the gaps in the WPF literature is the diversity of evaluating conditions found during the model development stage, making it difficult to have a clear idea of the actual improvement in forecasting accuracy provided by a new WPF model compared to existing ones, preventing industry users to choose any possible best model to implement their forecasting tools for electricity market operations (Wang et al., 2016c), as well as

preventing researchers to evaluate their models fairly against cutting-edge forecasting models. The lack of standardized criteria to evaluate WPF models becomes more significant as the number of WPF models proposed in the literature keeps steadily increasing (Giebel and Kariniotakis, 2017).

Therefore, developing benchmarks of WPF algorithms is essential to implement standards and best practices for the assessment of forecasts and their uncertainty (Giebel et al., 2016), as well as model evaluation (Lago et al., 2021). In preparation of a benchmark, some key points are often overlooked, such as (Möhrlen et al., 2018): 1) Forecasting horizons to be considered, 2) historical data available, 3) the representativeness of the operational conditions, and 4) standardized performance evaluation metrics. Additionally, the evaluation of WPF algorithms must be fair, meaning that it should not be expected to evaluate curtailment periods or human actions such as maintenance operations, and representative, meaning that the forecasting model should be tested under significant operation conditions (Möhrlen et al., 2018). Regarding the first point, classification of WPF models with respect to the prediction horizon is somewhat unclear in the literature (Roungkvist and Enevoldsen, 2020). One widely used convention is proposed by Soman et al. (2010), where four different time horizons are defined: very short-term, short-term, medium-term, and long-term. WPF models meet different needs depending on the forecasting horizon (Table 4.1). Statistical models are preferred for very short- and short-term horizons as they can be quickly built and updated as new data arrive, whereas physical approaches are preferred for longer prediction horizons. On another note, the type of forecast considered is significant for the benchmarking process. As pointed out in section 3.1.1, point (*deterministic*) estimates of WPFs have been predominantly used

Table 4.1: Applications of WPFs depending on the forecasting horizon (Soman et al., 2010).

Forecasting horizon	Timeframe	Applications
Very short-term	Up to 30 min ahead	Electricity market clearing Real-time grid operations Turbine control
Short-term	30 min to 6 h ahead	Load increment/decrement decisions Economic load dispatch planning
Medium-term	6 h to 1 day ahead	Generator decisions Operational security in Day-Ahead Market (DAM)
Long-term	More than 1 day ahead	Maintenance planning Unit commitment decisions Reserve requirement decisions

for industry applications, and they are evaluated measuring the discrepancy between the forecast and the measurement. Reducing the forecast error (i.e., increasing the forecast accuracy) improves the O&M of wind farms (Pakrashi et al., 2020) and has a positive impact on the total system costs. Mc Garrigle and Leahy (2015) simulated the Irish electricity system with 33% wind penetration to prove that reducing the forecast error can save 1.64% and 0.50% of total system costs for deterministic and stochastic scheduling respectively. However, point estimates are incomplete as the uncertainty of the forecast is not measured (González-Aparicio and Zucker, 2015; Zhang et al., 2019b). This gap is filled by probabilistic forecasts, which are gradually moving to the industry thanks to projects such as the IEA Wind Task 36 (Giebel et al., 2020). As described in section 3.1.2, PIs are a representation of probabilistic forecasts where the uncertainty is measured by estimating a range of values where future wind power observations are expected to fall with a certain confidence level (Wan et al., 2017), and can be tuned for making decisions in industry applications such as reserve quantification (Zhao et al., 2021). The goodness of a PI is assessed by measuring the coverage of the interval, that is, its ability to correctly spot the future observation within the PI, and the width of the interval.

Among the publications reviewed in the literature, decomposition-based hybrid models (initially introduced in section 1.4.1.4) have attracted the attention of the wind energy forecasting community due to an improved performance with respect to other statistical models (Qian et al., 2019). The evaluation of these models are also affected by the lack of benchmarks mentioned in the previous paragraph. For instance, Wang et al. (2019c) propose a wind power interval prediction method where data are decomposed using VMD as decomposition algorithm, and forecasting is performed using GRU neural networks. The approach is tested using 10-minute interval data from an offshore wind farm located in California and an onshore wind farm in Washington. Decomposition performed with EMD is for instance applied by Naik et al. (2018c), using this algorithm together with ANN-based forecasting models to predict wind power generation outputs in a wind farm located in Wyoming (USA) from 10 minutes to 3 hours ahead. A wavelet-based hybrid model is proposed by Li et al. (2020d), where the decomposed data are trained with SVM structures. Hourly

data collected in a French wind farm are used to benchmark the performance of the model. In general, we have identified 57 publications in the recent literature (Table 1.5) using decomposition-based hybrid models for WPF, but we are unable to compare them (nor any other model) as a result of not having any standardized procedure to evaluate them.

Following the conclusions drawn from the previous chapter, where we have provided a standardized view of performance evaluation metrics for statistical WPF models (González Sopeña et al., 2021a), we propose a benchmark framework to evaluate the performance of such models. The experimental set-up of this benchmark is designed considering the main guidelines of the IEA Wind Task 36: 1) Very short-term and short-term horizons are considered as these are the scenarios where statistical models thrive, 2) historical data collected in two Irish wind farms are used to evaluate the models, 3) varied and realistic operational conditions of the wind farms are evaluated to maximize the representativeness of the results, and 4) performance evaluation metrics for both deterministic WPFs and PIs are standardized throughout the experiment. The usefulness of this benchmark is proven with an example case testing decomposition-based hybrid models, where state-of-the-art decomposition techniques are used to decompose wind power time series data and the main ANN-based structures (Wang et al., 2021b) are applied to build forecasting models. Note that this benchmark framework is not limited to decomposition-based hybrid models, but other statistical models (especially ML/DL type of models) could be applied as well.

The rest of this chapter is structured as follows: section 4.2 describes the main features of decomposition-based hybrid models, the data and the experimental design followed to benchmark the models. Section 4.3 presents the results and discussion, and section 4.4 the concluding remarks of this chapter.

4.2 Methods

Decomposition-based hybrid models have two distinguishing elements: a preprocessing step where a decomposition algorithm divides the wind power time series into a set of more stationary modes, and the forecasting models implemented for every

mode. ANNs represent the most widespread structure to build forecasting models for wind power (Table 1.3). Different types of ANNs allow us to exploit diverse features of the data, such as recurrent or convolutional neural networks. Regardless of the decomposition algorithm or the forecasting model used, decomposition-based hybrid models follow usually the flowchart shown in Figure 4.1 (González Sopeña et al., 2021c): 1) the wind power time series is decomposed into several modes, 2) a forecasting model is built for each one, and 3) the wind power prediction is obtained aggregating the forecast of each mode.

A description of the main decomposition techniques used for WPF and the neural network based approaches used to build the forecasting models is provided down below. Furthermore, the experimental design developed for the benchmark is described in this section, as well as the data used to evaluate the models (which have been used previously in the numerical study conducted in section 3.3).

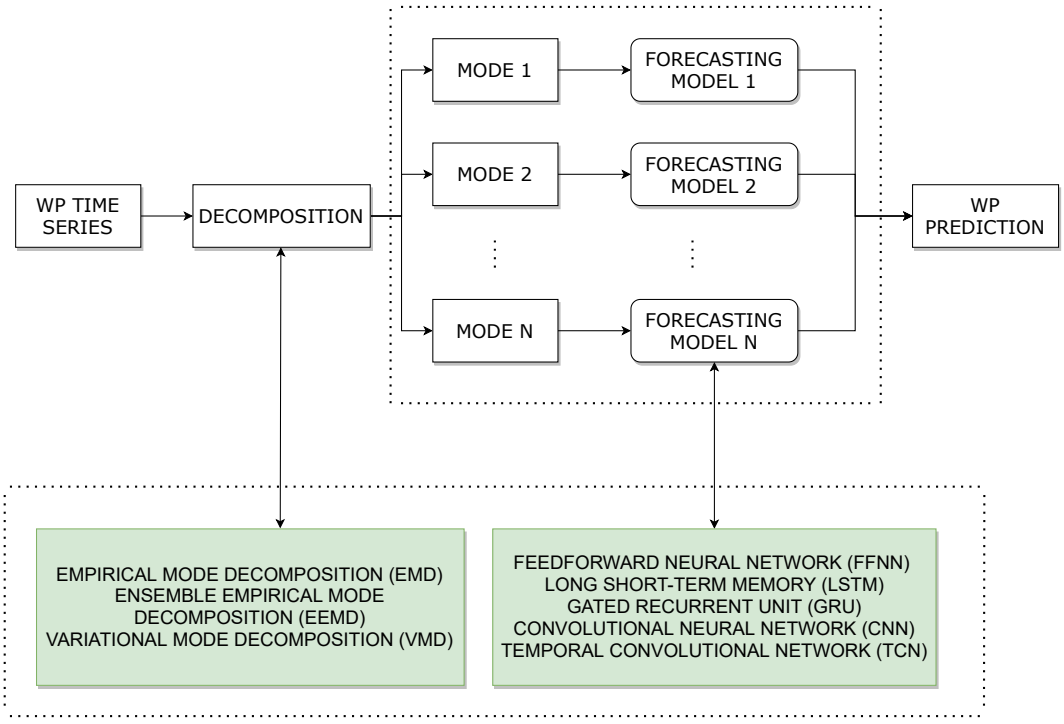

Figure 4.1: Flowchart for WPF using decomposition-based hybrid models.

4.2.1 Decomposition algorithms

Some of the most common techniques to decompose wind power time series are wavelet transform (Meyer, 1992; Daubechies and Bates, 1993), EMD (Huang et al.,

1998), and VMD (Dragomiretskiy and Zosso, 2013). While preprocessing wind power data using wavelet decomposition can lead to an improvement of forecasting performance, the decomposition effect has low adaptability as it depends greatly on the selection of the wavelet basis function, whereas EMD- and VMD-based algorithms present more adaptability to decompose wind power time series. Therefore, we apply VMD, EMD and EEMD (Wu and Huang, 2009) to decompose wind data.

4.2.1.1 Variational mode decomposition

VMD is a non-recursive signal processing method designed for decomposing complex non-stationary signals. The decomposition process into several subseries is performed by a constrained variational problem to determine the bandwidth of each mode. This process involves three steps: 1) the Hilbert transform is used to obtain the unilateral frequency spectrum for each mode, 2) an exponential tuned to the estimated center frequencies is used to shift every mode's frequency spectrum to baseband, and 3) the bandwidth of each mode is identified using the H^1 Gaussian smoothness of the demodulated signal. Thus, the constrained variational problem is defined as

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \tag{4.1}$$

where $\{u_k\}$ the set of all modes, $\{\omega_k\}$ the set of the respective center frequencies, $\delta(t)$ the Dirac function, $*$ denotes a convolution, and $\|\|_2^2$ denotes a squared L^2-norm. As suggested in the original paper (Dragomiretskiy and Zosso, 2013), this constrained variational problem can be transformed into an unconstrained problem introducing a quadratic penalty term and Lagrangian multipliers λ as follows:

$$\begin{aligned} L(\{u_k\},\{\omega_k\},\lambda) = \alpha' \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ + \left\| y(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), y(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \end{aligned} \tag{4.2}$$

This equation can be solved using a sequence of iterative sub-optimizations known as ADMM (Hestenes, 1969; Boyd et al., 2011). By doing so, the modes u_k and their respective center frequencies ω_k are then updated simultaneously with the following

expressions:

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i \neq k} \hat{u}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1 + 2\alpha'(\omega - \omega_k)^2} \tag{4.3}$$

$$w_k^{n+1} = \frac{\int_0^\infty \omega |\hat{u}_k(\omega)|^2 d\omega}{\int_0^\infty |\hat{u}_k(\omega)|^2 d\omega} \tag{4.4}$$

The modes are solved in the spectral domain, and can be transformed back into the time domain taking the real part of the inverse Fourier transform of the signal. In Eq. 4.3, the value α' represents a penalty term defined by the user which will define the shape of the modes.

4.2.1.2 Empirical mode decomposition

EMD is another decomposition method based on identifying the intrinsic oscillatory modes of the signal, known also as IMFs, and a residue. The EMD algorithm can be described in four steps:

1. Local maxima and minima are located in the time series data $y(t)$, and then interpolated to build an upper and a lower envelope respectively.

2. The mean value $m(t)$ of these envelopes is calculated and the first component H_1 is built by subtracting $m(t)$ from the original wind power time series $y(t)$.

3. These two steps are repeated till the stopping criterion is satisfied. Then, H_1 will correspond to the first mode and the residue to $y(t) - H_1$, the difference between the original time series and the first component.

4. Steps 1-3 are repeated with the resulting residues till all the modes and the last residue are computed.

The original wind power time series $y(t)$ can be reconstructed by simply adding up all the IMFs and the residue:

$$y(t) = \sum_{i=1}^{N} \text{IMF}_i + r(t) \tag{4.5}$$

4.2.1.3 Ensemble empirical mode decomposition

EEMD is the ensemble version of the EMD approach to solve mode mixing issues which may occur using the standard EMD algorithm. A set of trials using the EMD algorithm is performed, but the original time series $y(t)$ is mixed with Gaussian white noise ϵ_t. The white noise cancel each other when calculating the mean value of all of the trials. The steps to follow are:

1. Add Gaussian white noise ϵ_t to the original wind power time series data $y(t)$:

$$y'(t) = y(t) + \epsilon_t$$

2. Apply the EMD algorithm to $y'(t)$ to obtain the corresponding IMFs and residue in the same way as shown in Eq. 4.5.

3. Repeat steps 1-2 a preset number of times. Each IMF is computed as the arithmetic average of all these trials as follows:

$$\text{IMF}_i = \frac{1}{N}\sum_{j=1}^{N}\text{IMF}_{ij}$$

where IMF_i is the i-IMF obtained averaging a set of N trials.

4.2.2 ANN-based forecasting models

The most simple type of ANN is the FFNN, which consists of three layers (input, hidden and output layers) where the information is only propagated forward. The backpropagation algorithm (Rumelhart et al., 1986; Chauvin and Rumelhart, 2013) is typically used to train FFNNs.

Neural networks with more complex architectures have the ability to further learn patterns in the data. For instance, RNNs process time series data on a step-by-step basis, keeping an internal state to process a sequence of inputs. However, basic RNN structures suffer from vanishing/exploding gradients when longer-term dependencies are processed (Pascanu et al., 2013). Alternative and more advanced RNN architectures such as LSTM (Hochreiter and Schmidhuber, 1997) and GRU neural networks (Chung et al., 2014; Cho et al., 2014) overcome this issue. Another

type of neural networks are CNNs (Li et al., 2021), which are characterized by the use of filters to extract spatial features from the data. CNN- and RNN-based cells can be combined to extract both spatial and temporal features of the data at the same time (Chen et al., 2019b), resulting in forecasting models such as CNN-GRU (Liu et al., 2021c) or CNN-LSTM (Wu et al., 2021). Temporal convolutional networks (Gan et al., 2021) also regard temporal and spatial features, as the convolutions are causal, meaning that the output values are linked to the current and previous inputs (Bai et al., 2018). Let us provide a more complete description of these architectures down below.

4.2.2.1 Feedforward neural network

A simple FFNN consists of neurons (the processing units of any neural network model) divided into three layers: input, hidden, and output layers (Figure 4.2). This type of ANN is characterized by the flow of information only in a forward direction: the data arrive to the input layer and pass through the network in a layer by layer

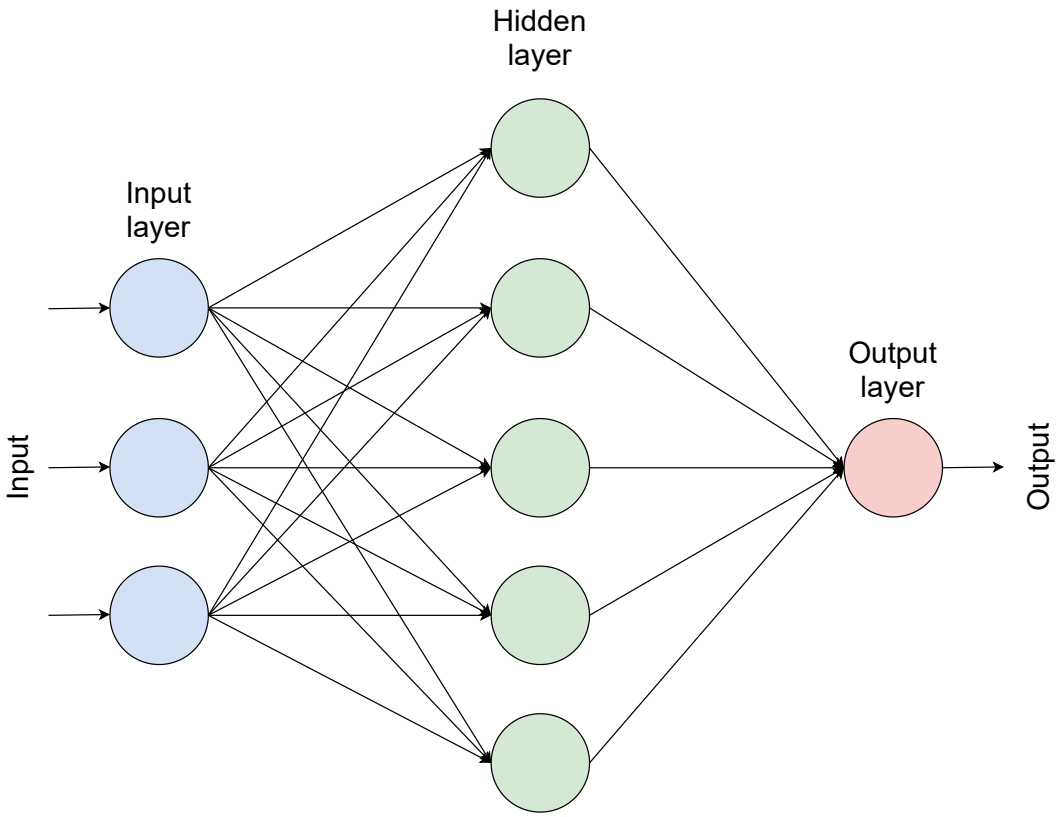

Figure 4.2: Architecture of a feedforward neural network.

basis till the output is computed. The backpropagation algorithm (Rumelhart et al., 1986) is de facto training algorithm for FFNNs.

Let us frame the learning process of a FFNN as a supervised learning problem where a set of inputs and outputs are known, so a given set of inputs and outputs can be used to set the parameters of the network. Two steps can be identified using the backpropagation algorithm, known as the forward and backward sweeps:

- *Forward sweep.* A known input is fed to the network, passes through every layer and finally computes an output. This output is then compared with the actual known (and desired) output.

- *Backward sweep.* The error between the output values is fed back to the network, adjusting the parameters of the network using a nonlinear optimization method known as gradient descent algorithm (Guo and Gelfand, 1990).

This process is repeated a certain number of iterations, with the expectation that the network converges in such a way that the error is small enough, meaning that the network learns the relation between inputs and outputs and is able to generalize the results using the testing data (Mitchell et al., 1986).

4.2.2.2 Long short-term memory

LSTM networks (Hochreiter and Schmidhuber, 1997) regulate the flow of information using three gates (called forget gate f_t, input gate i_t, and output gate o_t), and a reservoir of long-term memory known as cell state c_t to determine the hidden state h_t of the network, which corresponds to the output determined at every time step (Figure 4.3). The following equations indicate how the information is transmitted through the network.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4.6}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4.7}$$

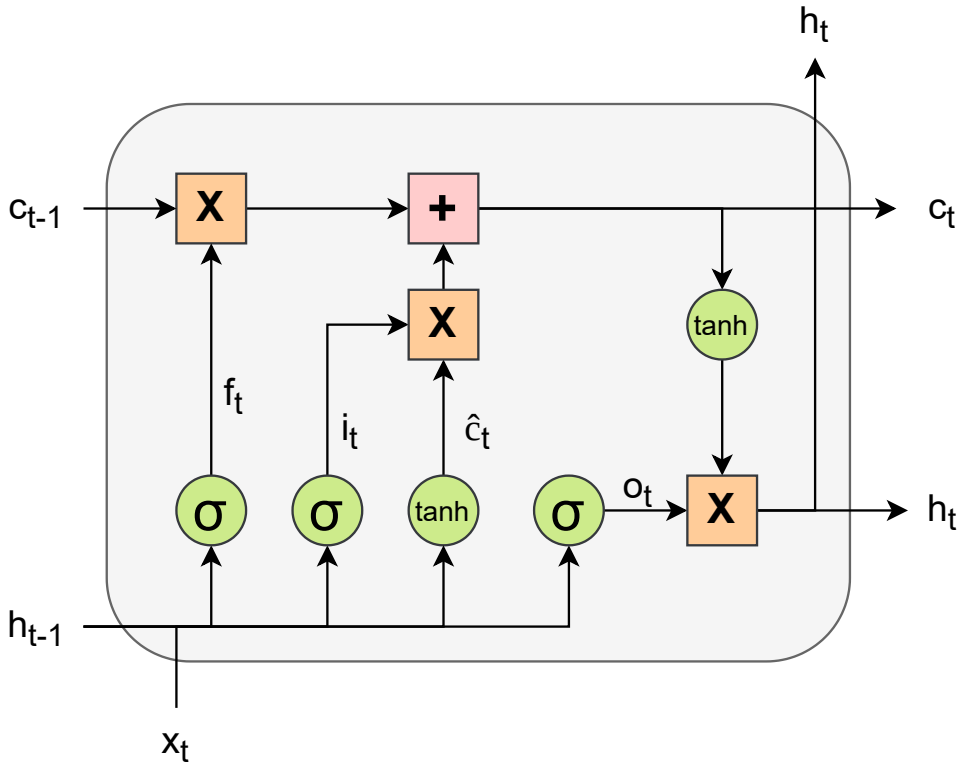

Figure 4.3: Structure of a LSTM network.

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{4.8}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4.9}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4.10}$$

$$h_t = o_t \odot \tanh(c_t) \tag{4.11}$$

Firstly, the LSTM network decides whether the information from the previous

timestep is discarded or maintained by means of the forget gate f_t (Eq. 4.6), where x_t is the input, h_{t-1} the previous hidden state, W_f and U_f are the weights for the input and previous hidden state respectively, b_f the bias, and σ represents a sigmoid activation function.

The next step is to renew the information contained in the cell state based on the input and the previous hidden state. The new memory network is determined by the candidate cell state $\tilde{c}_t$ (Eq. 4.8), whereas the input gate i_t (Eq. 4.7) acts a filter to decide whether this new information is worth to be added to the cell state, or otherwise such information should be filtered. In these equations, W_c and U_c are the weights for the input and previous hidden state for the candidate cell state, b_c the bias of the same candidate cell state, W_i and U_i the weights for the input gate, and b_i the bias of the input gate. In this case, the candidate cell state uses a hyperbolic tangent as the activation function, while the input gate is activated with a sigmoid activation function.

The cell state of the LSTM network is updated as shown in Eq. 4.10, combining the element-wise product $\odot$ of the forget gate and the previous cell state with the element-wise product of the input gate and the candidate cell state. At this stage, the new hidden state can be computed using the output gate (Eq. 4.9) and the updated cell state of the network, as shown in Eq. 4.11.

4.2.2.3 Gated recurrent unit

GRU networks constitute another advanced type of recurrent neural network with the ability of capturing long-term dependencies existing in data. In a GRU neuron (Figure 4.4), the flow of information is controlled by two gates instead of three as LSTM networks: an update gate z_t and a reset gate r_t. Data are transmitted following the next equations:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{4.12}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{4.13}$$

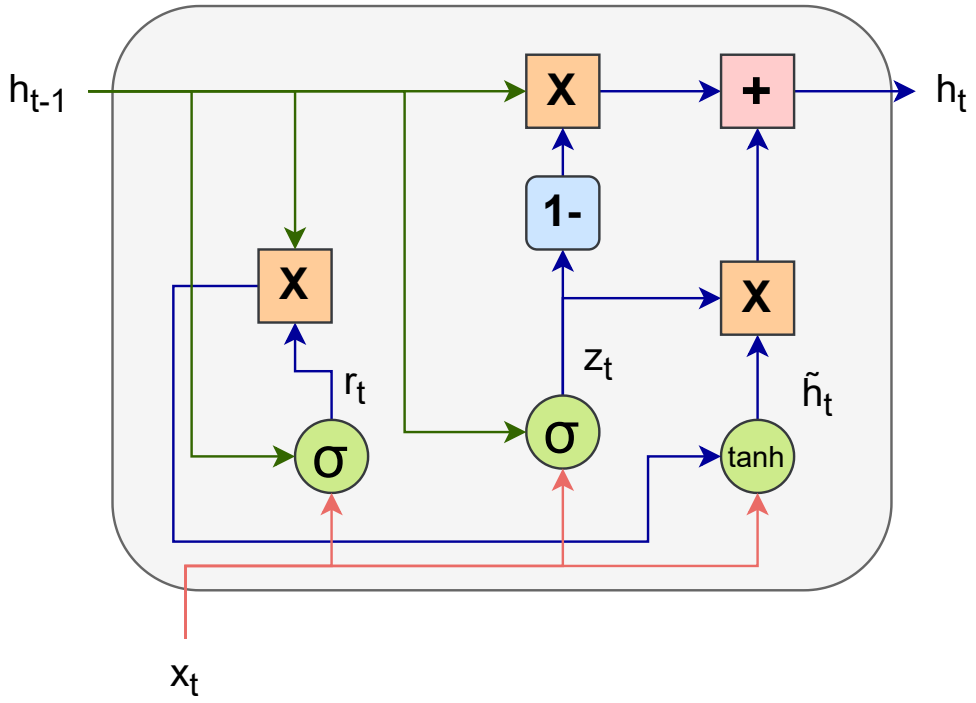

Figure 4.4: Structure of a GRU network.

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h\right) \tag{4.14}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4.15}$$

First, the input x_t and the previous hidden state h_{t-1} are used to compute the update gate (Eq. 4.12) and the reset gate (Eq. 4.13). W_z and U_z are the weights for these inputs along with the bias b_z in the update gate, while W_r, U_r and b_r represent their counterparts for the reset gate. Both gates are activated using a sigmoid activation function σ.

Afterwards, the candidate hidden state $\tilde{h}_t$ (Eq. 4.14) is computed. An element-wise product $\odot$ is computed between the reset gate r_t and the previous state h_{t-1} to decide what information from the previous hidden state should be filtered. An hyperbolic tangent is now applied as the activation function.

Finally, the current hidden state h_t (Eq. 4.15) is calculated transmitting the

information gathered so far as a linear interpolation between the previous state h_{t-1} and the candidate state $\tilde{h}_t$ (Chung et al., 2014). Thus, the previous information is brought to the current state by the update gate, whereas the reset gate acts as filter to erase superfluous information.

4.2.2.4 Convolutional neural network

CNNs are a type of neural network architecture in which at least one of the layers perform a convolution to find features invariant across spatial dimensions (Goodfellow et al., 2016). A convolution is a mathematical operation defined as follows:

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{4.16}$$

where $(f * g)$ indicates the convolution between the functions f and g, in which the function f can be considered as a *filter* or *kernel* and g as the input data. On the right hand side, $g(t-\tau)$ indicates that the input data g is reversed and shifted to a certain time t. In the context of CNNs, the convolutional layer is formed by a certain number of filters which are convoluted with the input data to detect any existing hidden features.

TCNs represent a special case of convolutional networks where the convolution operations keep causality by restricting the operation to elements from the current and previous time step in the previous layer (Bai et al., 2018).

4.2.2.5 Quantile regression based neural networks

Usually, ANN-based models provide a single output for every time step. Uncertainty information can be provided estimating PIs. In particular, the lower and upper boundaries of the PI can be estimated using ANN-based models. For instance, Khosravi et al. (2010) proposed the LUBE method to calculate the boundaries of the interval, where a neural network is trained using a loss function based on metrics to determine the coverage and width of PIs. Implementations of this methodology can be found in the WPF literature (as described in section 1.4.1.5), such as Kavousi-Fard et al. (2015), where a fuzzy based loss function is considered to facilitate the

adjustement of the ANN parameters, and Shi et al. (2017), where a RNN model is used to build PIs with the LUBE method. The main drawback of this method is that gradient descent based algorithms cannot be applied to minimize the LUBE loss function due to its complexity (Khosravi et al., 2010), which is impractical as it is the standard method to train ANN models. An alternative method to obtain PIs is to implement a quantile regression cost function (Koenker and Hallock, 2001), which allows us to determine directly the quantiles corresponding to the boundaries of the interval (Yu et al., 2020), and depends on the quantile τ to be forecast:

$$\rho_\tau(\epsilon) = \begin{cases} \tau\epsilon, & \text{if } \epsilon \geqslant 0 \\ (\tau - 1)\epsilon, & \text{otherwise} \end{cases} \tag{4.17}$$

The error function to minimize is:

$$E_\tau = \frac{1}{N} \sum_{i=1}^{N} \rho_\tau(y(i) - \hat{y}_\tau(i)) \tag{4.18}$$

where *y(i)* is the target, and $\hat{y}_\tau(i)$ is the conditional τ-quantile. Thus, the quantiles corresponding to the upper and lower boundaries can be calculated to estimate PIs. For instance, a prediction interval with 90% confidence can be built setting the loss function to calculate the 0.05- and 0.95-quantiles:

$$\text{PI} = [\hat{y}_{0.05}, \hat{y}_{0.95}] \tag{4.19}$$

where $\hat{y}_{0.05}$ is the forecast of the 0.05-quantile, corresponding to the lower boundary of the interval, and $\hat{y}_{0.95}$ is the forecast 0.95-quantile which represents the upper boundary. Using a decomposition-based hybrid model, these values are obtained after aggregating the corresponding lower and upper boundaries quantities of each mode in the following way:

$$\hat{y}_{0.05} = \sum_{k=1}^{K} \hat{y}_{0.05}^{k} \tag{4.20}$$

$$\hat{y}_{0.95} = \sum_{k=1}^{K} \hat{y}_{0.95}^{k} \tag{4.21}$$

where k represents the total number of modes. Furthermore, multi-step ahead forecasts are estimated by using the MIMO strategy (Taieb et al., 2010), as defined in Eq. 3.24.

All these decomposition algorithms and neural network based forecasting models can be combined to build a complete decomposition-based hybrid model. In order to make the study as comprehensive as possible, 21 models are implemented in the experimental set-up, arising from combining the 3 decomposition algorithms (VMD, EMD, and EEMD) and the 7 forecasting models (FFNN, GRU, LSTM, CNN, CNN-GRU, CNN-LSTM, and TCN) introduced in this section. Regardless of the decomposition algorithm used, the wind power data are divided into six modes. Then, the forecasting models are implemented to estimate three vector outputs, corresponding to the 0.5-quantile for point predictions, and the lower and upper boundaries of the PIs, which are set at the 0.05- and 0.95-quantiles to build intervals with 90% confidence level. In regard to model parameters, the training is performed using a batch of size 64 for 100 epochs, although early stopping (Prechelt, 1998) is implemented to stop the training in order to avoid overfitting (Ying, 2019). The hidden layer of the FFNN and RNN-based models have 50 neurons in total, the CNN layers are set with 50 filters with a kernel size = 6, and the TCN layers are formed by 50 filters with dilation factors d = 1, 2, 4 and a filter size k = 6.

4.2.3 Data

NWP data are composed by forecasts of meteorological variables and represent one of the main sources of predictability for WPF models (Foley et al., 2012; Yan et al., 2015b). However, NWP data underperform for shorter-term scales due to the computational requirements to obtain such data and the uncertainty associated to these meteorological variables (Tawn and Browell, 2022). In addition, short-term phenomena such as turbulence or gusts can lead to fluctuations of the wind power production in a range below 1 hour to seconds (Peinke et al., 2004). While the effects of such

phenomena have not been directly considered during the development of the implemented forecasting models, they are still present in the wind power signal as any turbulent fluctuation will have an impact on the recorded wind power production. Even if the accuracy of these forecasting models could be partially improved in some particular scenarios (such as ramp events) if wind gusts were to be detected (Pichault et al., 2022), the uncertainty added by the required additional variables would potentially limit the improvement in model performance. In the same line, microscale flows could be computed including turbulence parameterization, but at the cost of additional uncertainty of the associated parameters (Yan et al., 2022). On the other hand, a fair representation of adjacent orography in shorter-term scales is limited to the availability of high-resolution terrain data, which is difficult to estimate in practice (Al-Yahyai et al., 2010).

Alternatively, recent observations of the target variable (wind power generation in this case) can be used as the predictors for shorter-term scale WPF models to reduce computational cost and avoid the uncertainty associated to NWP data and short-term weather phenomena (Messner and Pinson, 2019). This is the approach applied in the WPF forecasting models used in this thesis, thus wind power production observations gathered with SCADA are used as the predictor of the implemented WPF forecasting models. These SCADA data, collected in two Irish wind farms, are described down below.

4.2.3.1 Data description

SCADA data from two Irish wind farms were provided by a third party under a non-disclosure agreement. The data from both wind farm I (WF-I) and wind farm II (WF-II) have been collected from 2017-01-01 to 2019-06-30 at a 10-min resolution. WF-I has a total of 11 turbines, whereas WF-II is formed by 17 turbines. In both cases, SCADA data are recorded at turbine level and contain historical measurements of wind power (Figs. 4.5-4.6), as well as other variables such as wind speed, rotation speed and nacelle direction.

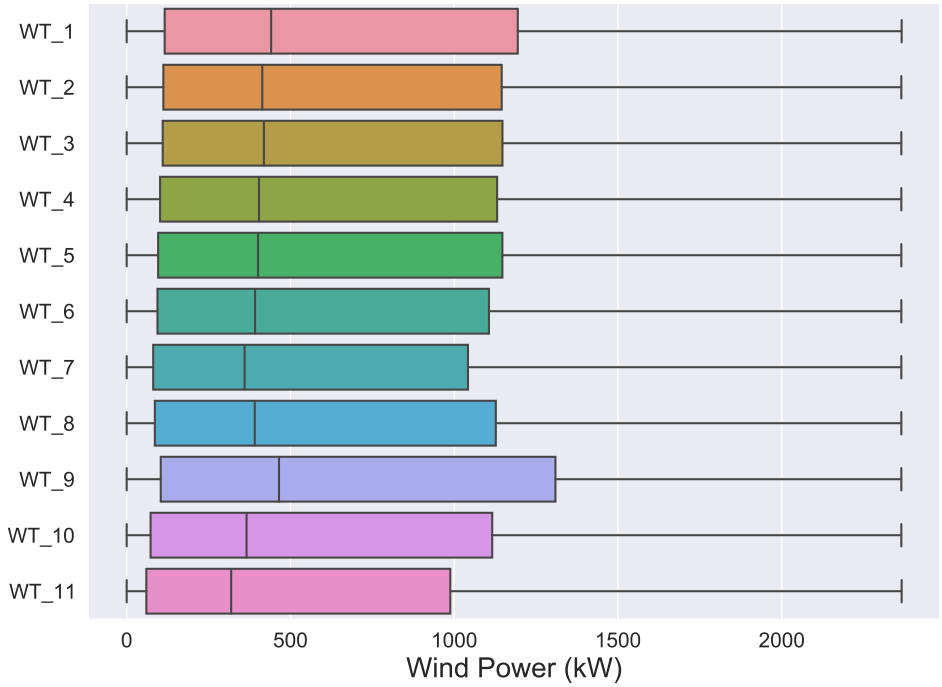

Figure 4.5: Wind power distribution at turbine-level for wind farm I.

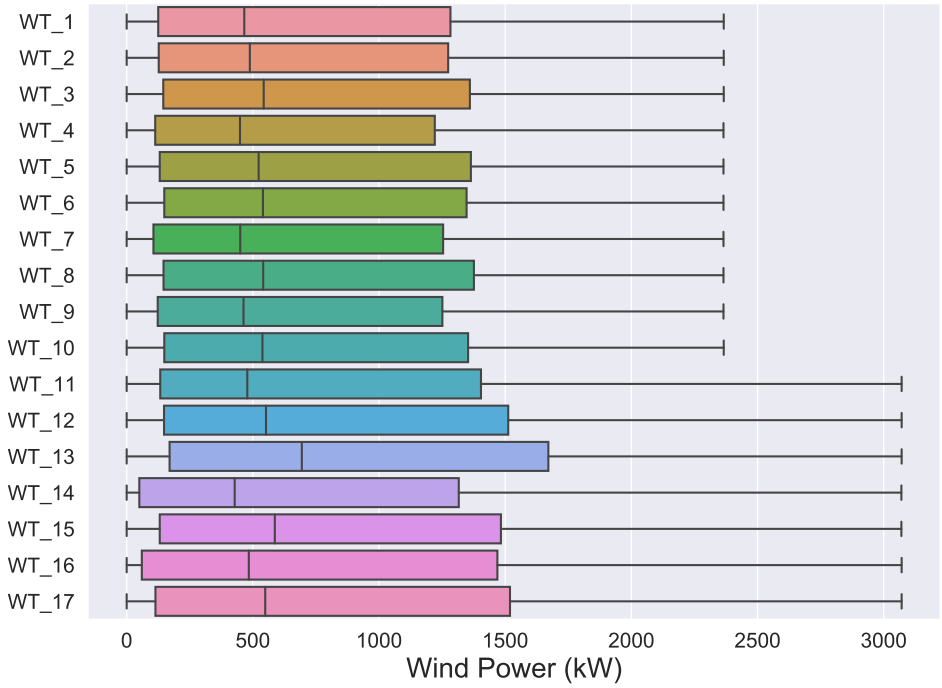

Figure 4.6: Wind power distribution at turbine-level for wind farm II.

4.2.3.2 Data preparation

Turbine-level wind power measurements are aggregated to build a forecasting model to consider the whole wind farm as a single entity. In order to maximize the representativeness of the results, we have first divided the data into a total of ten 1-year long subsets for very short-term forecasts, where the records from first eleven months have been used for training and validation, and records from the last month have been used to benchmark the models for very short-term WPFs. Data continuity has been kept in the subsets to prevent any leakage of information from the future to the past. The testing sets for WF-I and WF-II are shown in Figures 4.7 and 4.8 respectively. In both cases, the testing sets exhibit a large fluctuation of wind power generation, which shows the volatile (nonstationary) nature of wind power. Aside from that, it can be observed that there are periods where the wind farms have been temporarily halted (DS-9 and DS-10 for WF-I, and DS-1 and DS-2 for WF-II). In particular, the DS-9 for the WF-I is almost completely halted in the whole testing set. To provide a representative assessment, these subsets will be discarded for the evaluation of the models, as long curtailment periods should be excluded to keep the comparison fair (Möhrlen et al., 2018).

For longer prediction horizons, the chosen datasets are downsampled to 1-hour resolution. As the number of samples is lower in this case, five 18-month long datasets have been used instead to compensate the reduction in the sample size. The first 17 months of data have been used for training and validation while the data from the last month were used for testing. The test data corresponding to these subsets are shown in Figure 4.9 for WF-I (left) and WF-II (right).

4.2.4 Experimental design

The experimental design (Figure 4.10) is set up to explore the effectiveness of WPF algorithms considering a variety of operational conditions to maximize the representativeness of the results, as well as account for data uncertainty by considering multiple realizations of the original wind power time series (section 2.1.1). To achieve this, the original wind power datasets are divided into smaller continuous subsets to evaluate the performance of various models using different testing data. Further-

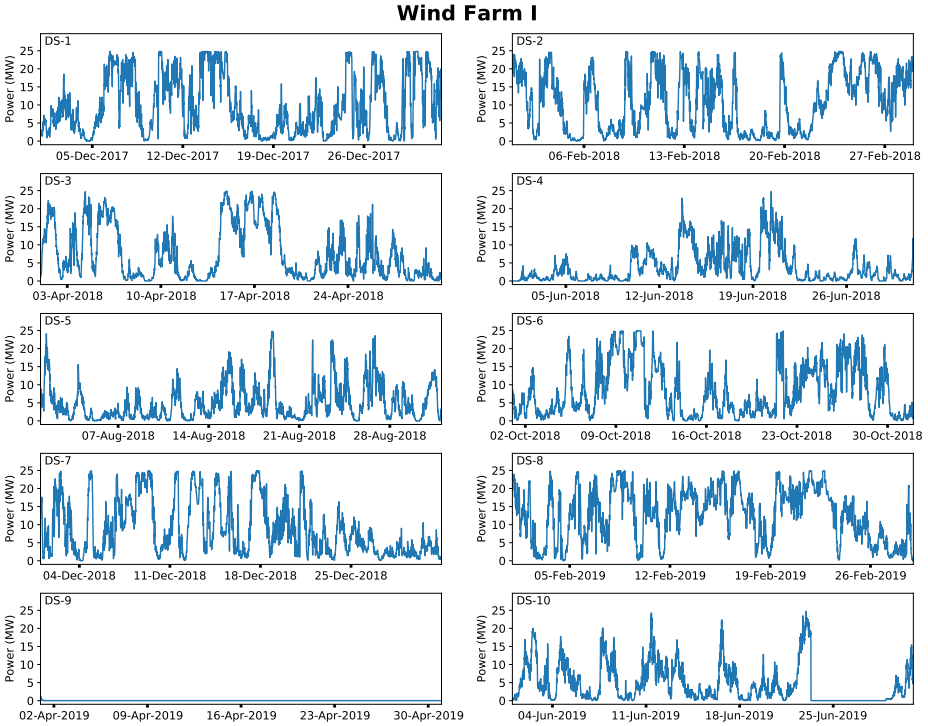

Figure 4.7: Testing data subsets from Wind Farm I to evaluate hybrid WPF models (10-minute resolution data).

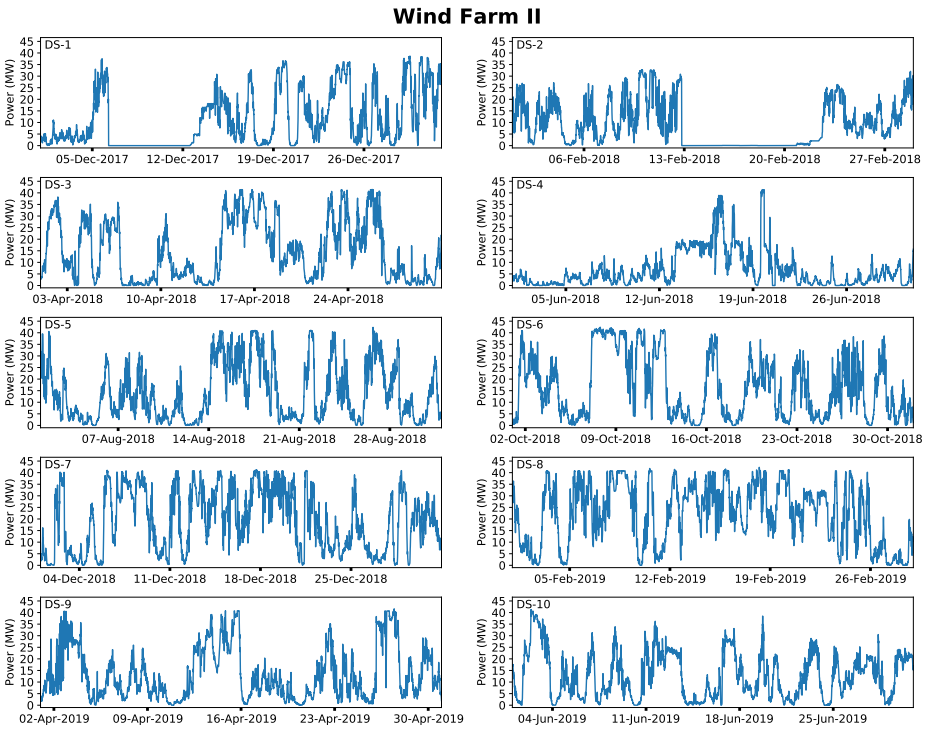

Figure 4.8: Testing data subsets from Wind Farm II to evaluate hybrid WPF models (10-minute resolution data).

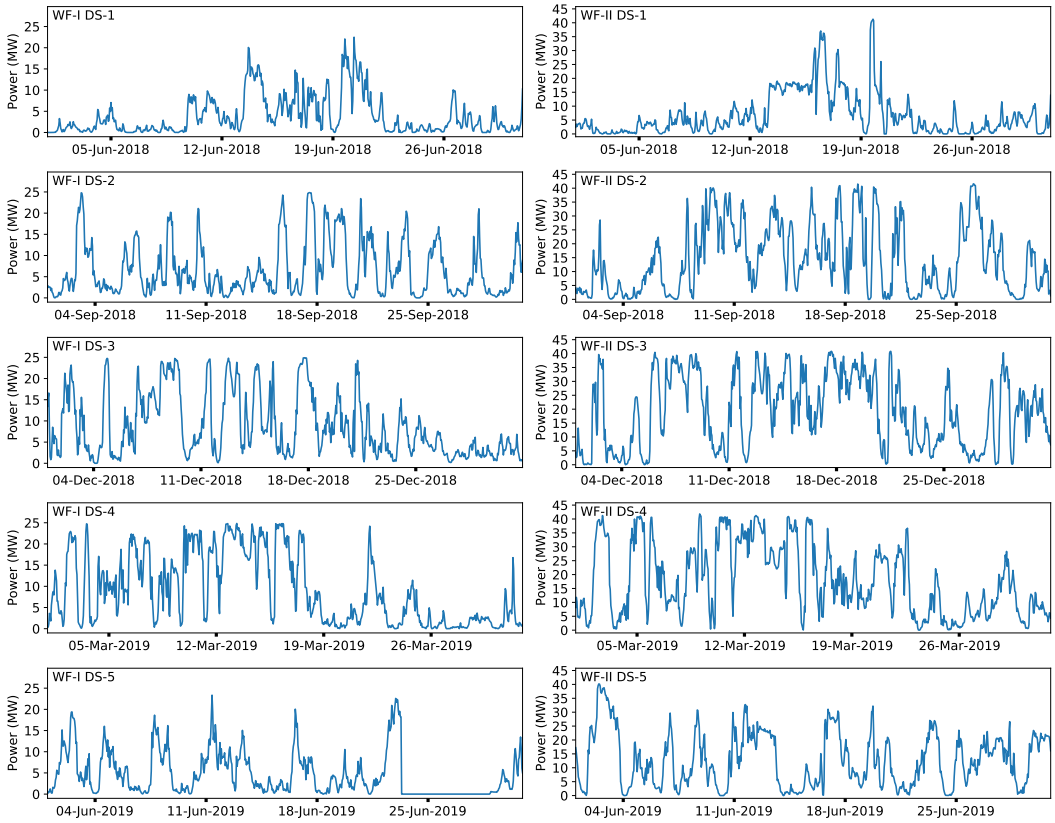

Figure 4.9: Testing data subsets from Wind Farm I (left) and Wind Farm II (right) to evaluate hybrid WPF models (1-hour resolution data).

more, the testing data should not include long curtailment periods to ensure a fair representation of the operational conditions of the wind farm (Möhrlen et al., 2018). For different forecasting horizons, the length and the resolution of the data has to be specified, as these factors influence the forecast results (González Sopeña et al., 2021a). Once the forecasts are estimated with all the WPF models, the accuracy of point forecasts is verified using NMAE, and the coverage and width of PIs using PICP and PINAW respectively. The robustness of the forecasting models is analyzed with the spread of the metrics over all the simulations.

As mentioned previously, a total of 21 models are tested using this benchmark, resulting from combining the three algorithms shown in section 4.2.1 (VMD, EMD, and EEMD), and the 7 ANN-based models described in section 4.2.2, which are used to build forecasts for every mode (Figure 4.1). The training of the ANN models will be repeated for every subset five times for very short-term and ten times for short-term prediction horizons, as forecasts are expected to be different even if the same set of data is used to train the forecasting model, as the weights of the ANN

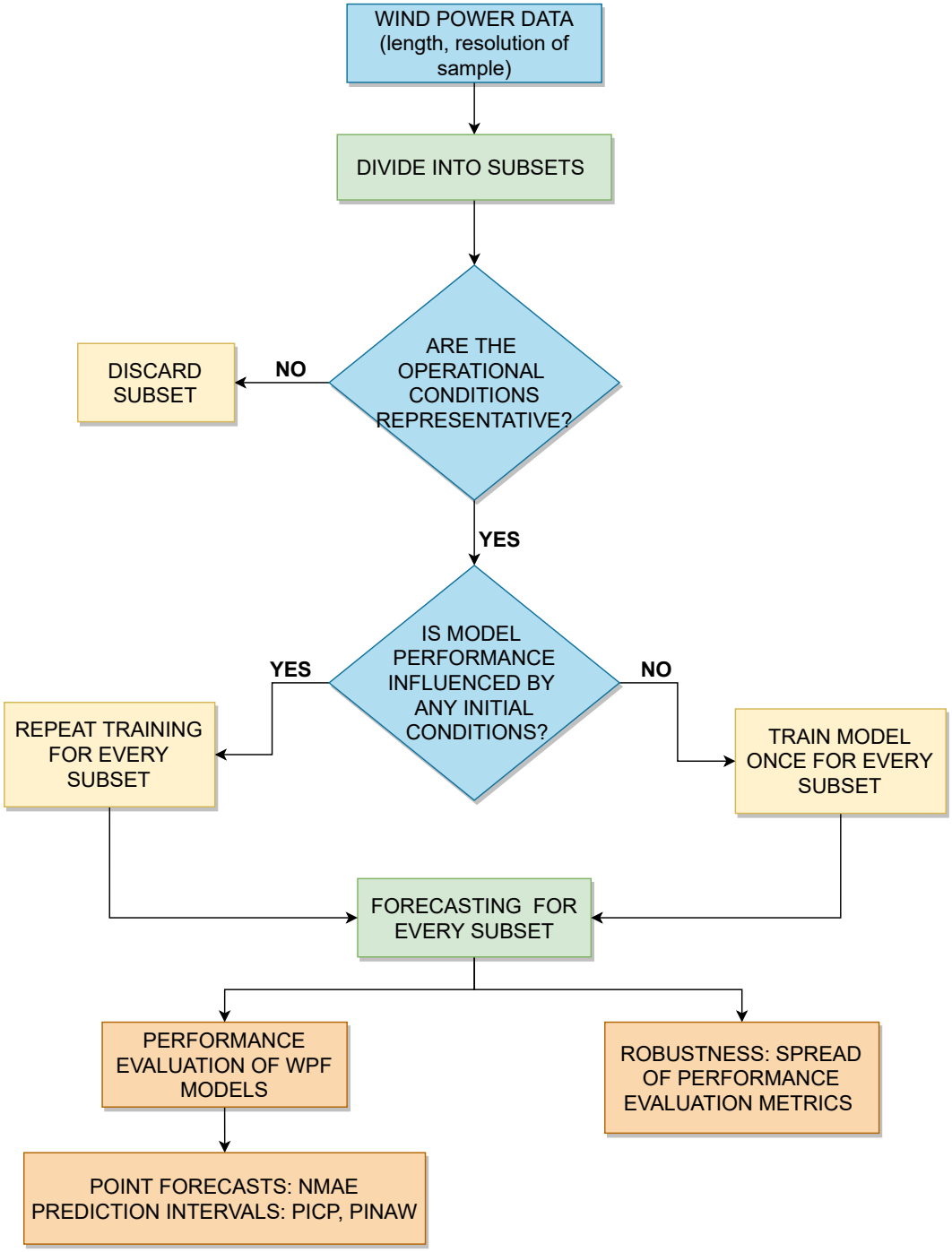

Figure 4.10: Proposed benchmarking process for evaluating statistical WPF models.

models are randomly initialized during the training stage (Pollack, 1990). This way of proceeding must be considered for any model built under random initial conditions to reduce the bias of initialization.

4.3 Results & Discussion

The experimental set-up is first established for very short-term forecasts, which have been estimated using the original 10-minute resolution datasets. Afterwards, the resampled 1-hour resolution data have been used to obtain WPFs for short-term horizons. In both cases, the benchmark is applied to 21 decomposition-based hybrid models to evaluate and compare their performance.

4.3.1 Very short-term forecasts

As very short-term forecasts take into account horizons up to 30 minutes ahead, here we will consider 10-, 20-, and 30-minute ahead WPFs, which correspond to 1- to 3-step ahead forecasts using the 10-minute interval data. The previous 72 steps, which correspond to the previous 12 hours in 10-minute intervals, are used as input for the models.

Datasets DS-9 and DS-10 for WF-I (Figure 4.7) and DS-1 and DS-2 for WF-II (Figure 4.8) have been discarded, as the corresponding testing sets contain large periods where the wind farm was halted, which might bias the results as they represent anomalous operation conditions. In addition to that, the parameters learned by ANN-based models will change from one simulation to another due to randomly initializing the weights of the networks, so every model will be trained five times to reduce the bias of initialization, as pointed out in section 4.2.4. Thus, a total of 40 simulations have been performed for all models considering all the representative subsets of data in every wind farm. Some results are displayed in Figure 4.11, showing point forecasts and PIs in some of the datasets (DS-1 and DS-8 for WF-I, and DS-4 and DS-6 for WF-II) using the VMD-GRU model.

The performance evaluation of point forecasts is shown in Table 4.2, where the average values of the NMAE over all simulations for WF-I and WF-II are displayed. Decomposition-based hybrid models using VMD to decompose wind power data have a higher performance than those using either EEMD or EMD. This gap of performance is explained by the empirical nature of EMD-based algorithms (Wang and Markert, 2016). Among the models using the VMD algorithm, those based on RNN architectures exhibit a higher accuracy. For 10-minute ahead WPFs, the VMD-CNN-

GRU models has a average NMAE value of 0.42% for WF-I, whereas the VMD-GRU model has a score of 0.4% for WF-II. For 20-minute ahead forecasts, a value of 0.59% is obtained by the VMD-GRU model for WF-I, and a score of 0.56% is shared by the VMD-GRU and VMD-CNN-GRU models for WF-II. Lastly, a value of 0.91% is obtained by the VMD-GRU, VMD-CNN-GRU and VMD-CNN-LSTM models for 30-minute ahead WPFs for WF-I, and a value of 0.83% for the VMD-GRU and VMD-CNN-GRU for WF-II. Hence, implementing an additional CNN layer before the GRU or the LSTM layer does not generate any substantial increase of performance for point forecasts. This pattern is observed for the EMD- and EEMD-based models as well. In the WF-I, an average NMAE value of 1.35% is observed for 10-minute ahead WPFs using the EMD-GRU model, compared to an 1.30% using the EMD-CNN-GRU model, whereas a 1.23% is quantified for the average NMAE value of the EEMD-GRU model, similar to the EEMD-CNN-GRU model (1.22%).

With reference to PIs, Tables 4.3 and 4.4 show the average values of PICP and PINAW over all the simulations. In terms of coverage, all the models exhibit a

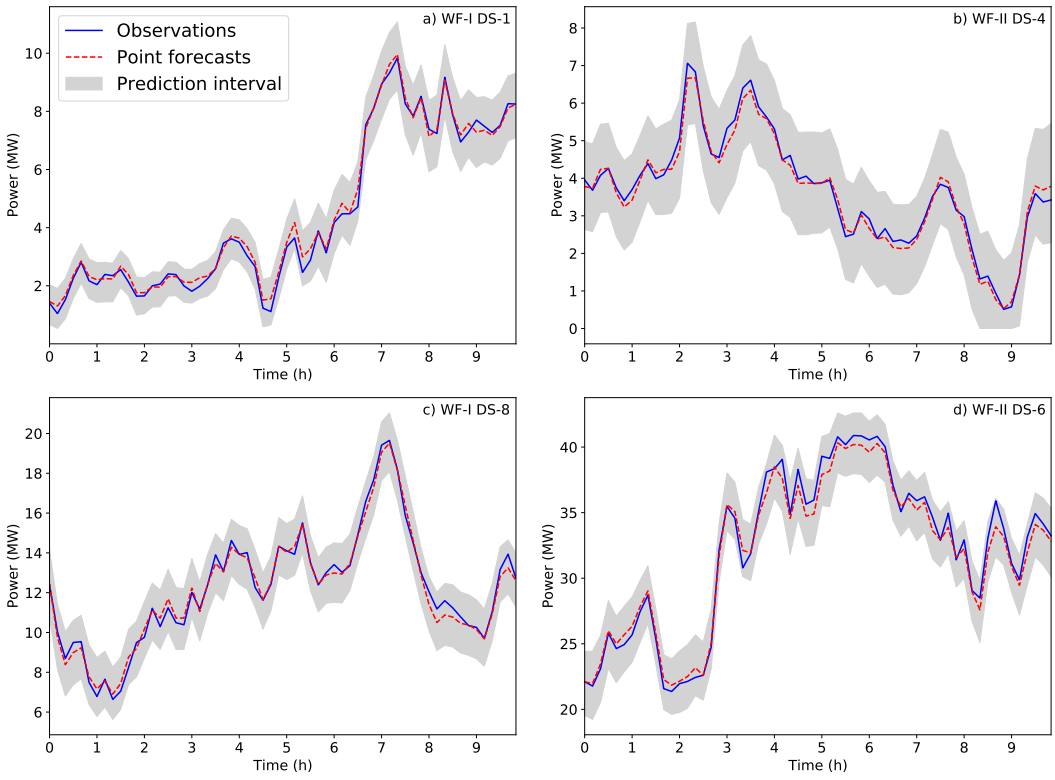

Figure 4.11: Examples of point forecasts and prediction intervals for 30 minute forecasting horizon from Wind Farm I and Wind Farm II utilizing the VMD-GRU model (10-minute resolution data).

Table 4.2: Average NMAE (%) for very short-term forecasts.

	WF-I			WF-II		
Model \ Forecast horizon	10-min	20-min	30-min	10-min	20-min	30-min
VMD-FFNN	0.77	0.97	1.13	0.7	0.86	1
VMD-GRU	0.43	**0.59**	**0.91**	**0.4**	**0.56**	**0.83**
VMD-LSTM	0.46	0.66	0.92	0.43	0.59	0.84
VMD-CNN	0.82	0.91	1.1	0.73	0.78	0.95
VMD-CNN-GRU	**0.42**	0.61	**0.91**	0.41	**0.56**	**0.83**
VMD-CNN-LSTM	0.43	0.61	**0.91**	0.43	0.59	0.86
VMD-TCN	0.57	0.8	1.05	0.5	0.74	0.97
EMD-FFNN	1.69	2.13	2.58	1.73	2.16	2.61
EMD-GRU	1.35	1.8	2.18	1.76	2.09	2.42
EMD-LSTM	1.31	1.71	2.1	1.4	1.75	2.04
EMD-CNN	1.62	2.02	2.31	1.76	2.08	2.34
EMD-CNN-GRU	1.3	1.72	2.08	1.33	1.72	2.03
EMD-CNN-LSTM	1.3	1.69	2.07	1.4	1.78	2.08
EMD-TCN	1.38	1.7	2.04	1.35	1.61	1.91
EEMD-FFNN	1.38	1.75	1.93	1.28	1.58	1.77
EEMD-GRU	1.23	1.56	1.71	1.15	1.41	1.56
EEMD-LSTM	1.21	1.54	1.69	1.15	1.41	1.56
EEMD-CNN	1.37	1.69	1.85	1.32	1.59	1.75
EEMD-CNN-GRU	1.22	1.54	1.7	1.18	1.45	1.61
EEMD-CNN-LSTM	1.23	1.57	1.74	1.17	1.42	1.57
EEMD-TCN	1.28	1.59	1.75	1.2	1.46	1.6

similar behavior, including around 98-99% of the observations of the testing set and therefore exceeding the confidence level set for the intervals. On the other hand, narrower intervals are preferred as they will be more beneficial for end-users. The lower PINAW values show that the RNN-based models using the VMD algorithm (VMD-GRU, VMD-LSTM, VMD-CNN-GRU, and VMD-CNN-LSTM) are capable of building narrower intervals in comparison to the rest of models. In particular, the VMD-CNN-GRU model builds slightly narrower intervals in most cases: a score of 3.11% for 10-minute ahead intervals for WF-II, scores of 6.13% and 5.55% for 20-minute ahead intervals for WF-I and WF-II, and scores of 9.28% and 8.5% for WF-I and WF-II respectively. As expected, the intervals are wider as the forecast horizon increases.

The robustness of the models to varied operational conditions is evaluated with the spread of the performance evaluation metrics over all the simulations (Figure 4.12). Since the same model outputs both the point predictions and the boundaries of the interval, the ideas extracted from studying the NMAE distribution (which

Table 4.3: Average PICP (%) for very short-term forecasts.

	WF-I			WF-II		
Model \ Forecast horizon	10-min	20-min	30-min	10-min	20-min	30-min
VMD-FFNN	98.33	99.55	**99.98**	99.65	99.84	**99.95**
VMD-GRU	99.51	**99.93**	99.91	99.43	**99.92**	99.88
VMD-LSTM	99.64	99.89	99.92	99.54	99.9	99.91
VMD-CNN	98.39	99.58	99.42	98.95	99.5	99.82
VMD-CNN-GRU	99.52	99.9	99.9	99.16	99.91	99.89
VMD-CNN-LSTM	99.69	99.91	99.91	99.45	99.88	99.9
VMD-TCN	**99.86**	99.71	99.93	**99.8**	99.91	99.94
EMD-FFNN	99.68	99.84	99.64	99.66	99.8	99.78
EMD-GRU	99.67	99.82	99.81	99.31	99.68	99.76
EMD-LSTM	99.8	99.85	99.82	99.33	99.7	99.77
EMD-CNN	99.45	99.71	99.73	99.71	99.82	99.89
EMD-CNN-GRU	99.69	99.82	99.86	99.39	99.73	99.79
EMD-CNN-LSTM	99.78	99.89	99.86	99.34	99.7	99.77
EMD-TCN	99.85	99.92	99.92	99.84	99.89	99.88
EEMD-FFNN	99.63	99.77	99.8	99.59	99.7	99.77
EEMD-GRU	98.56	99.35	99.53	98.49	99.24	99.44
EEMD-LSTM	98.55	99.39	99.54	98.46	99.21	99.49
EEMD-CNN	99.61	99.78	99.78	99.62	99.67	99.73
EEMD-CNN-GRU	98.48	99.35	99.53	98.42	99.14	99.44
EEMD-CNN-LSTM	98.66	99.44	99.59	98.48	99.28	99.5
EEMD-TCN	99.2	99.55	99.65	99.3	99.48	99.56

Table 4.4: Average PINAW (%) for very short-term forecasts.

	WF-I			WF-II		
Model \ Forecast horizon	10-min	20-min	30-min	10-min	20-min	30-min
VMD-FFNN	5.97	8.54	11.66	6.05	8.29	11.08
VMD-GRU	**3.34**	**6.13**	9.3	3.17	5.63	8.66
VMD-LSTM	3.66	6.37	9.48	3.43	5.93	8.96
VMD-CNN	6.4	8.71	11.26	6.53	8.05	10.68
VMD-CNN-GRU	3.35	**6.13**	**9.28**	**3.11**	**5.55**	**8.5**
VMD-CNN-LSTM	3.6	6.24	9.49	3.45	5.86	8.83
VMD-TCN	4.94	7.36	10.42	4.58	6.82	9.52
EMD-FFNN	12.49	16.66	20.67	14.75	18.63	22.31
EMD-GRU	9.33	13.43	17.26	13.43	17.15	20.58
EMD-LSTM	9.16	13.07	16.59	11.21	15.16	18.4
EMD-CNN	12.49	16.4	19.61	14.72	18.23	21.62
EMD-CNN-GRU	8.84	12.77	16.18	10.72	14.85	18.3
EMD-CNN-LSTM	9.06	12.91	16.47	11.49	15.4	18.79
EMD-TCN	9.76	13.39	16.1	9.86	13.28	15.68
EEMD-FFNN	10.53	14.39	16.3	9.75	13.27	15.4
EEMD-GRU	7.78	11.47	13.52	7.23	10.66	12.47
EEMD-LSTM	7.45	11.17	13.09	7.06	10.45	12.48
EEMD-CNN	11.97	15.12	16.38	11.11	14.02	15.9
EEMD-CNN-GRU	7.51	11.22	13.16	7.44	10.83	12.79
EEMD-CNN-LSTM	7.83	11.53	13.55	7.32	10.73	12.7
EEMD-TCN	8.91	12.47	14.54	8.64	11.74	13.65

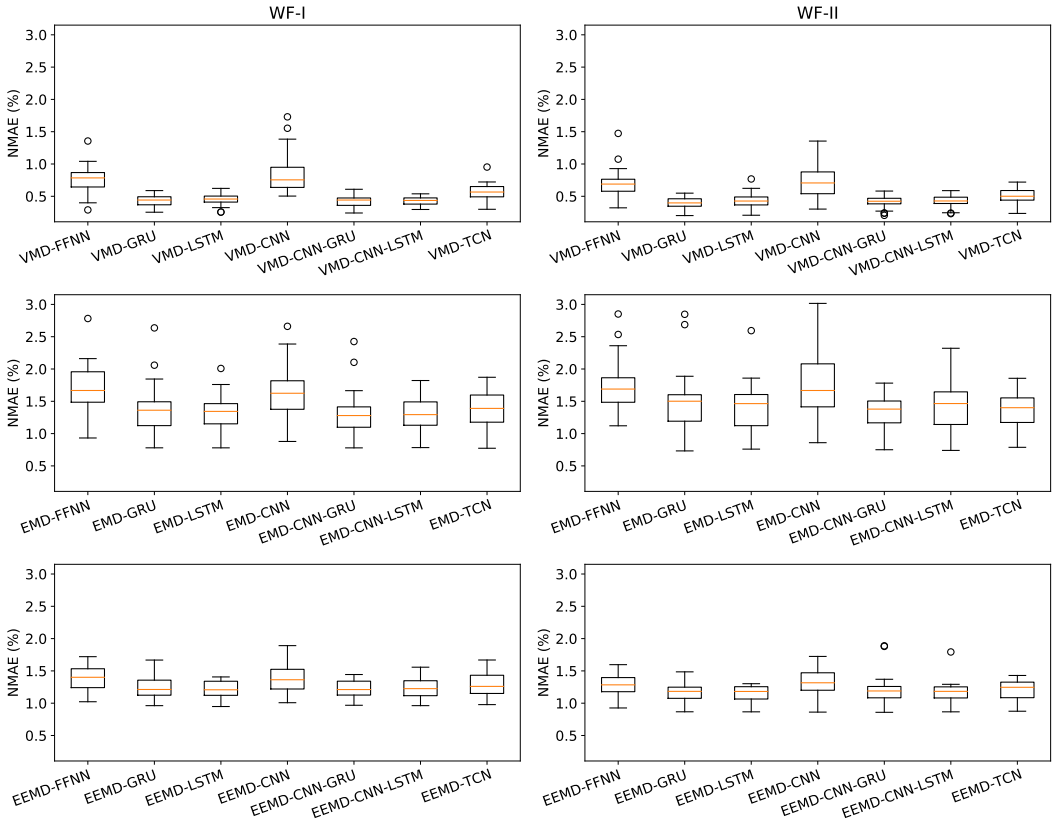

Figure 4.12: Distribution of NMAE values for 10-minute ahead forecasts of all studied models from Wind Farm I (left) and Wind Farm II (right) (10-minute resolution data).

correspond to the performance evaluation of point forecasts) are valid for PIs as well, as the network parameters are shared by all the output vectors. In this case, the combination of the VMD algorithm with RNN-based structures, including the VMD-CNN-GRU and VMD-CNN-LSTM models, is the most robust among all the tested models, as the spread of NMAE values is very low for both farms. Besides that, the low variability of the NMAE values for these models demonstrates their ability to adapt to different training/testing conditions for wind power time series considered in the experimental design, indicating that the network parameters are able to capture all of them with a similar degree of accuracy. On the other hand, preprocessing wind power time series using the EMD and EEMD algorithms shows not only a lower accuracy for point forecasts, but a lower degree of robustness as they variability is higher in both cases.

In light of the simulations performed for very short-term WPFs, the benchmarking process indicates that a combination of the VMD algorithm with RNN-based models (either a GRU or a LSTM layer) is the most adequate among the evaluated

decomposition-based hybrid models. The VMD-CNN-GRU and VMD-CNN-LSTM models do not show a significant increase in performance with respect to the VMD-GRU and VMD-LSTM models, so the CNN layer can be omitted if implementing a model for an individual wind farm. The benchmark is both fair and representative, as anomalous events have been discarded from the evaluation process while considering different realistic operational conditions. Furthermore, evaluating the models with this variety of operational conditions also reduces the influence of the data on the forecast results, which allows to extrapolate these results to other farms within the same region.

4.3.2 Short-term forecasts

Short-term horizons consider WPFs from 1- to 6-h ahead. As we use 1-hour resolution data, forecasts will correspond consequently to 1- to 6-step ahead forecasts in the models. The input vector is still formed by the previous 72 data points, but in this case representing wind power measurements recorded in the previous 72 hours.

In the benchmark process, only the dataset DS-5 for the WF-I (Figure 4.9) has been omitted not to bias the results, as the wind farm has been halted in a significant period of the testing set. Once again, the effect of randomly initializing the weights of the networks is minimized by training every subset ten times. This way, 40 simulations for WF-I and 50 simulations for WF-II are considered respectively. Some examples showing 2-h ahead point forecasts and PIs using the VMD-GRU model are shown in Figure 4.13.

The average NMAE values to evaluate the performance of point estimates of WPFs are shown in Table 4.5. As occurred for very short-term forecasts, VMD outperforms the other decomposition algorithms due to the empirical features of EMD. Furthermore, the use of advanced recurrent structures provides once again the best model performance in terms of accuracy. In particular, the VMD-GRU model has the lower NMAE values in most cases: 0.96% in WF-II for 1-h ahead forecasts, 1.58% in WF-I and 1.61% in WF-II for 2-h ahead forecasts, 2.34% in WF-I and 2.35 in WF-II for 3-h ahead predictions, 2.96% in WF-I and 3.35% in WF-II for 4-ahead forecasts, and 4.25% and 5.32% for 6-ahead WPFs. Only for 5-h ahead

Table 4.5: Average NMAE (%) for short-term forecasts.

Wind Farm I						
Model \ Forecast horizon	1-h	2-h	3-h	4-h	5-h	6-h
VMD-FFNN	1.2	1.95	2.74	3.54	4.22	4.89
VMD-GRU	0.96	**1.58**	**2.34**	**2.96**	3.71	**4.25**
VMD-LSTM	**0.95**	1.65	2.38	3.01	3.68	4.33
VMD-CNN	1.09	1.62	2.41	3.2	3.82	4.44
VMD-CNN-GRU	0.96	1.61	2.37	2.98	3.74	4.29
VMD-CNN-LSTM	0.96	1.63	**2.34**	2.97	**3.66**	4.26
VMD-TCN	1.13	1.86	2.63	3.49	4.05	4.72
EMD-FFNN	3.02	3.95	4.94	5.87	6.56	7.18
EMD-GRU	2.71	3.66	4.39	5.22	5.94	6.53
EMD-LSTM	2.77	3.69	4.54	5.4	6.09	6.63
EMD-CNN	2.96	3.82	4.61	5.44	6.18	6.81
EMD-CNN-GRU	2.71	3.62	4.37	5.13	5.87	6.46
EMD-CNN-LSTM	2.76	3.72	4.54	5.39	6.1	6.63
EMD-TCN	2.73	3.58	4.4	5.18	5.83	6.34
EEMD-FFNN	2.25	3.1	3.53	4.34	5.13	5.49
EEMD-GRU	2.03	2.85	3.11	3.81	4.64	5
EEMD-LSTM	2.03	2.86	3.15	3.88	4.7	5.08
EEMD-CNN	2.23	3.02	3.35	4.1	4.98	5.39
EEMD-CNN-GRU	2.05	2.85	3.13	3.83	4.68	5.03
EEMD-CNN-LSTM	2.04	2.84	3.15	3.9	4.74	5.12
EEMD-TCN	2.2	3	3.34	4.1	4.8	5.07
Wind Farm II						
Model \ Forecast horizon	1-h	2-h	3-h	4-h	5-h	6-h
VMD-FFNN	1.18	1.95	2.89	4.12	5.18	6.04
VMD-GRU	**0.96**	**1.61**	**2.35**	**3.35**	4.52	**5.32**
VMD-LSTM	0.98	1.62	2.39	3.36	**4.5**	5.34
VMD-CNN	1.13	1.71	2.53	3.72	4.7	5.6
VMD-CNN-GRU	0.97	1.63	2.37	3.37	4.55	5.33
VMD-CNN-LSTM	**0.96**	1.64	2.42	3.4	4.55	5.38
VMD-TCN	1.15	1.92	2.8	3.98	5.02	5.9
EMD-FFNN	3.18	4.21	5.05	5.84	6.71	7.49
EMD-GRU	2.97	3.95	4.58	5.21	5.98	6.57
EMD-LSTM	3.32	4.13	4.68	5.4	6.22	6.79
EMD-CNN	3.22	4.12	4.82	5.53	6.34	7.01
EMD-CNN-GRU	2.98	3.94	4.58	5.21	5.98	6.61
EMD-CNN-LSTM	3.29	4.12	4.68	5.45	6.25	6.82
EMD-TCN	3	3.83	4.5	5.04	5.62	6.2
EEMD-FFNN	2.28	3.36	3.85	4.71	5.6	6.04
EEMD-GRU	2.11	3.07	3.42	4.12	4.95	5.38
EEMD-LSTM	2.11	3.07	3.43	4.2	5.06	5.53
EEMD-CNN	2.26	3.26	3.67	4.43	5.29	5.71
EEMD-CNN-GRU	2.11	3.07	3.4	4.12	4.96	5.41
EEMD-CNN-LSTM	2.11	3.07	3.44	4.22	5.06	5.49
EEMD-TCN	2.26	3.21	3.62	4.36	5.15	5.51

predictions other models have a slightly higher performance than the VMD-GRU model: the VMD-CNN-LSTM models for WF-I, which has an average NMAE value of 3.66%, and the VMD-LSTM models for WF-II, with a value of 4.5%. Furthermore, in contrast to the results shown for very short-term forecasts, the VMD-CNN model has a similar performance to the RNN-based models for WF-I considering all forecast horizons, whereas the performance is almost as good using EEMD instead of VMD for WF-II.

The performance evaluation for PIs in terms of coverage and width are shown in Tables 4.6 and 4.7 respectively. All the models provide valid intervals, as their coverage is larger than the confidence level for all of them. As shown in the experimental results for very short-term forecasts, the intervals cover around 98-99% of the wind power generation outputs in the testing sets. Regarding the width of the intervals, the same combination as before (VMD with either GRU or LSTM architectures) build the narrower intervals up to 4 hours ahead. In the WF-I, the VMD-GRU provides the narrower 1-h ahead intervals, as shown by the PINAW values (7.19%),

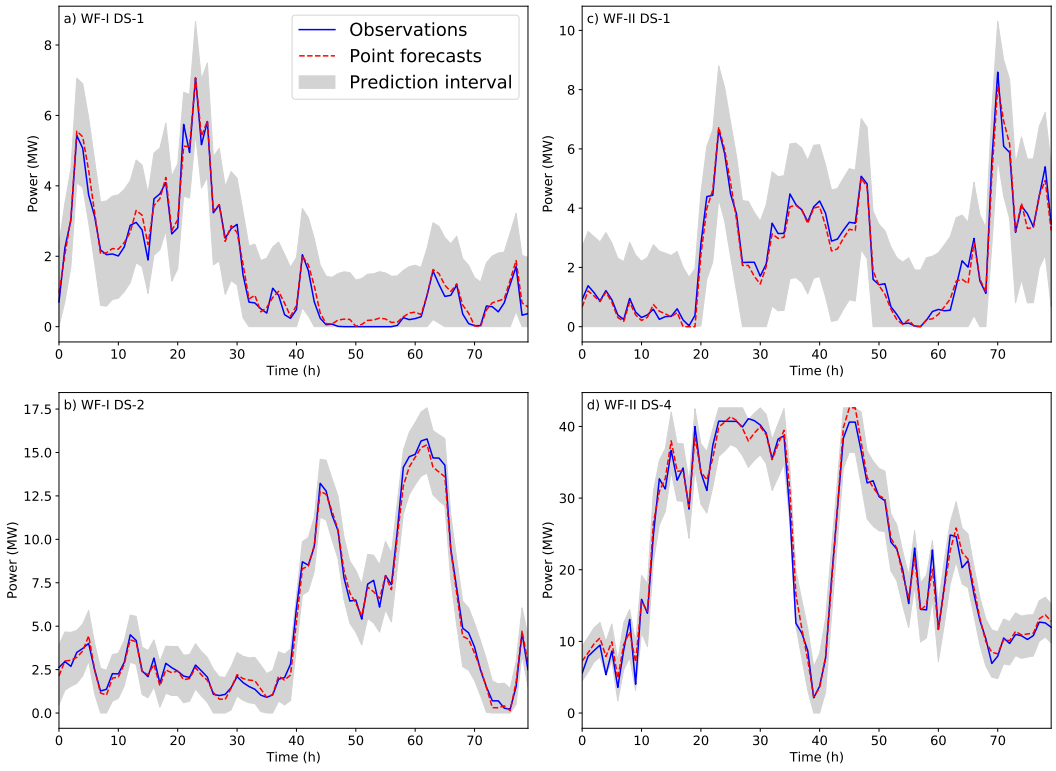

Figure 4.13: Examples of point forecasts and prediction intervals for 2 hour forecasting horizon from Wind Farm I and Wind Farm II utilizing the VMD-GRU model (1-hour resolution data).

Table 4.6: Average PICP (%) for short-term forecasts.

Wind Farm I						
Model \ Forecast horizon	1-h	2-h	3-h	4-h	5-h	6-h
VMD-FFNN	99.85	99.73	99.83	99.81	99.91	99.86
VMD-GRU	99.21	99.72	99.82	**99.95**	99.95	**99.94**
VMD-LSTM	99.7	99.74	99.8	**99.95**	99.96	99.88
VMD-CNN	99.65	99.72	**99.85**	99.89	99.89	99.81
VMD-CNN-GRU	99.22	99.67	99.74	99.94	99.95	99.92
VMD-CNN-LSTM	99.58	99.75	99.78	99.93	**99.97**	99.92
VMD-TCN	99.79	99.7	99.62	99.64	99.83	99.79
EMD-FFNN	99.84	99.84	99.66	99.59	99.62	99.68
EMD-GRU	99.79	99.64	99.64	99.57	99.61	99.57
EMD-LSTM	99.77	99.62	99.48	99.47	99.55	99.57
EMD-CNN	**99.95**	**99.92**	99.76	99.73	99.75	99.72
EMD-CNN-GRU	99.75	99.53	99.61	99.48	99.61	99.56
EMD-CNN-LSTM	99.77	99.6	99.61	99.44	99.5	99.56
EMD-TCN	99.86	99.76	99.53	99.42	99.4	99.45
EEMD-FFNN	99.53	99.54	99.57	99.34	99.26	99.3
EEMD-GRU	99.12	98.88	99.37	99.35	99.14	99.15
EEMD-LSTM	99.23	99.07	99.39	99.36	99.17	99.1
EEMD-CNN	99.61	99.58	99.7	99.52	99.39	99.3
EEMD-CNN-GRU	99.07	98.92	99.33	99.31	99.13	99.16
EEMD-CNN-LSTM	99.22	99.02	99.42	99.35	99.06	99.01
EEMD-TCN	99.47	99.42	99.53	99.19	98.95	98.91
Wind Farm II						
Model \ Forecast horizon	1-h	2-h	3-h	4-h	5-h	6-h
VMD-FFNN	**99.82**	99.72	99.72	99.59	99.64	99.53
VMD-GRU	99.16	99.5	99.7	99.79	99.75	99.76
VMD-LSTM	99.23	99.59	99.71	99.77	99.74	99.73
VMD-CNN	99.56	99.59	99.7	99.41	99.58	99.46
VMD-CNN-GRU	99.03	99.52	99.73	99.77	99.72	99.76
VMD-CNN-LSTM	99.31	99.54	99.7	99.71	99.69	99.7
VMD-TCN	99.79	99.61	99.57	99.27	99.28	99.26
EMD-FFNN	99.73	99.79	99.79	99.83	99.75	99.74
EMD-GRU	99.42	99.71	99.71	99.75	99.78	99.79
EMD-LSTM	98.83	99.47	99.6	99.7	99.72	99.7
EMD-CNN	99.77	**99.83**	**99.86**	**99.9**	**99.87**	**99.86**
EMD-CNN-GRU	99.46	99.69	99.71	99.73	99.79	99.79
EMD-CNN-LSTM	99.09	99.53	99.65	99.68	99.74	99.74
EMD-TCN	99.68	99.66	99.68	99.69	99.67	99.74
EEMD-FFNN	99.59	99.47	99.58	99.45	99.41	99.33
EEMD-GRU	98.66	98.43	99.05	99.05	99.06	99.11
EEMD-LSTM	98.83	98.75	99.11	98.94	98.88	98.81
EEMD-CNN	99.69	99.37	99.39	99.36	99.14	99.06
EEMD-CNN-GRU	98.69	98.55	99.01	99.07	99.07	99.07
EEMD-CNN-LSTM	98.95	98.83	99.15	99.1	99.01	98.94
EEMD-TCN	99.48	99.34	99.26	99.05	98.81	98.81

Table 4.7: Average PINAW (%) for short-term forecasts.

Wind Farm I						
Model \ Forecast horizon	1-h	2-h	3-h	4-h	5-h	6-h
VMD-FFNN	10.1	16.04	22.72	29.08	35.62	39.58
VMD-GRU	**7.19**	13.37	19.99	26.25	32.83	37.12
VMD-LSTM	8.07	13.85	20.31	26.42	33.09	37.53
VMD-CNN	10.13	15.17	21.63	27.89	33.82	37.98
VMD-CNN-GRU	7.35	**13.23**	**19.96**	**26.09**	32.74	37.11
VMD-CNN-LSTM	7.83	13.65	20.15	26.2	32.98	37.22
VMD-TCN	10.15	15.35	20.89	26.32	**32.06**	36.2
EMD-FFNN	19.88	27.96	34.37	40.17	44.98	49.4
EMD-GRU	17.17	24.28	30.34	36.25	40.91	45.06
EMD-LSTM	17.8	24.94	30.98	36.67	40.9	44.91
EMD-CNN	21.72	28.42	33.8	38.88	43.71	47.6
EMD-CNN-GRU	16.73	23.84	30.04	35.8	40.44	44.44
EMD-CNN-LSTM	17.86	25.13	31.24	36.66	41.01	45.03
EMD-TCN	18.89	24.57	29.23	33.25	37.08	40.26
EEMD-FFNN	16.07	23.19	27.46	32.68	37.41	40.68
EEMD-GRU	12.77	19.72	23.58	28.63	33.74	37.08
EEMD-LSTM	13.35	20.03	23.88	28.85	33.94	37.05
EEMD-CNN	17.7	23.86	27.78	32.54	37.16	40.11
EEMD-CNN-GRU	13.07	20.21	24.14	29.17	34.38	37.89
EEMD-CNN-LSTM	13.25	19.99	24.04	28.84	33.81	37.01
EEMD-TCN	16.03	22.45	25.31	29.32	33.23	**35.74**
Wind Farm II						
Model \ Forecast horizon	1-h	2-h	3-h	4-h	5-h	6-h
VMD-FFNN	10.19	16.23	23.56	30.99	38.32	42.63
VMD-GRU	7.2	**13.26**	20.42	27.7	35.39	40.35
VMD-LSTM	7.52	13.64	20.54	27.7	35.46	40.33
VMD-CNN	10.33	15.58	22.49	29.61	36.81	41.2
VMD-CNN-GRU	**7.11**	13.36	**20.37**	**27.59**	35.3	40.34
VMD-CNN-LSTM	7.47	13.52	20.55	27.71	35.51	40.47
VMD-TCN	10.77	15.64	21.89	28.07	35.02	39.81
EMD-FFNN	22.4	30.79	37.24	43.34	48.68	53.1
EMD-GRU	19.4	27.69	33.28	39.03	44.27	48.72
EMD-LSTM	22.14	29.28	34.61	40.18	44.63	48.36
EMD-CNN	23.1	30.77	36.34	41.7	46.6	50.64
EMD-CNN-GRU	19.61	27.88	33.48	39.33	44.44	48.87
EMD-CNN-LSTM	22.81	29.58	35.1	40.61	45.11	48.86
EMD-TCN	20.61	26.78	31.25	35.66	39.27	43.13
EEMD-FFNN	17	24.42	28.98	33.98	39.06	42.48
EEMD-GRU	13.24	20.42	24.58	29.73	35.17	38.78
EEMD-LSTM	13.5	20.45	24.77	29.39	34.42	37.75
EEMD-CNN	18.03	24.08	28.54	32.97	37.51	40.64
EEMD-CNN-GRU	13.28	20.64	24.68	29.95	35.49	39.11
EEMD-CNN-LSTM	13.57	20.71	24.75	29.66	34.82	38.22
EEMD-TCN	16.77	23.04	26.33	30.4	**34.32**	**36.89**

and the VMD-CNN-GRU model outperforms the rest from 2- to 4-h ahead intervals, having average PINAW values of 13.23%, 19.96% and 26.09% respectively. In the WF-II, the VMD-GRU models has the best results for 2-h ahead PIs in terms of coverage with a PINAW value of 13.26%, and the VMD-CNN-GRU model for 1-h, 3-h, and 4-h ahead PIs. For 5-h and 6-h ahead forecasts, the interval width of most models using VMD and EEMD even out, showing an average PINAW value between 35-40%.

The robustness of the models is analyzed likewise with the distribution of NMAE values over all the simulations (Figure 4.14). Overall, the VMD based models are more robust compared to the EMD and EEMD based models, as the NMAE scores are more similar over all the simulations, indicating that these models have the ability to capture more accurately the features of wind power data regardless of the operational conditions featured on the datasets. The models using GRU and LSTM networks present the lower variability among the VMD models for both wind

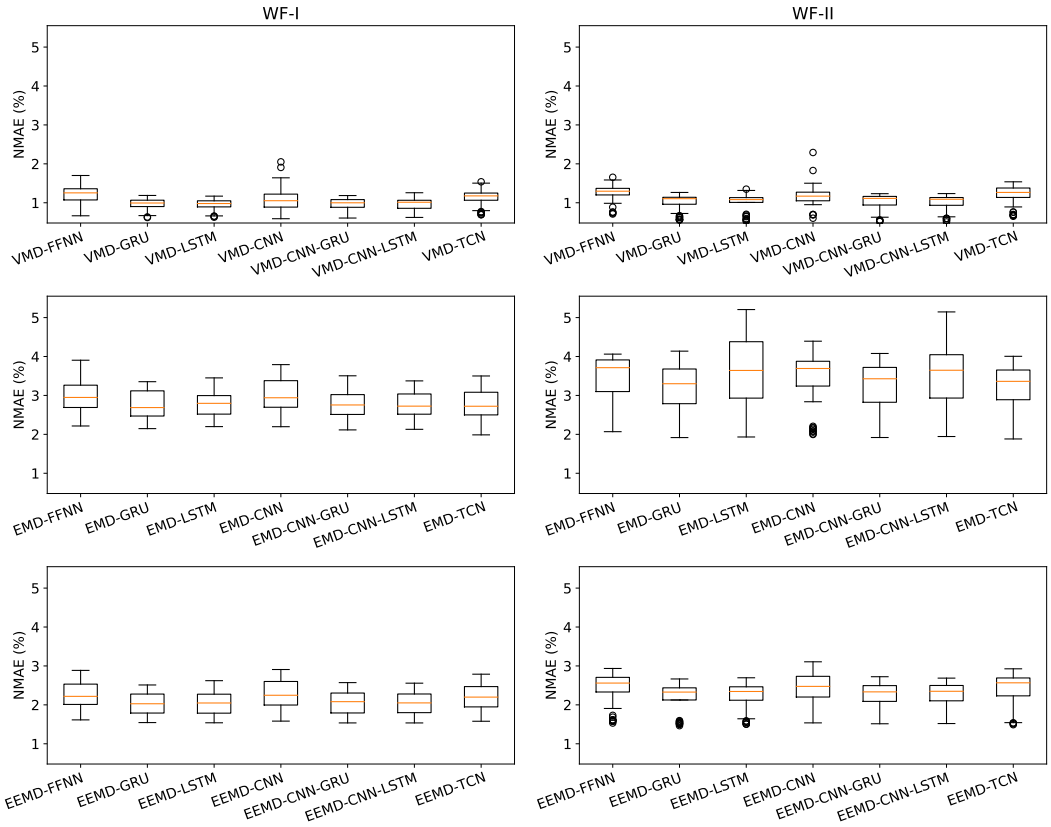

Figure 4.14: Distribution of NMAE values for hourly forecasts of all studied models from Wind Farm I (left) and Wind Farm II (right) (1-hour resolution data).

farms. In the WF-II, the NMAE values for the VMD-CNN are mostly found within a narrow interquartile range, but cannot be considered as robust as the outliers in the right tail indicate that the model performance is significantly worst for some of the simulations. Once again, the models using EMD and EEMD display lower accuracy and robustness.

Just as demonstrated using the proposed benchmark for very short-term forecasts, combining the VMD algorithm with GRU and LSTM neural networks provide the best results among all the decomposition-based hybrid models evaluated in the experiment. The only exception takes place when longer prediction horizons from short-term PIs are considered (5 and 6 hours ahead forecasts), as most models using the VMD and EEMD algorithms provide similar intervals in relation to both their width and coverage.

4.4 Conclusions

For very short-term prediction horizons, the models have been evaluated using 10-minute resolution datasets over periods of one month from two Irish wind farms. The results indicate that combining the VMD algorithm to decompose the wind data with advanced RNN structures to build the forecasting models (in particular the GRU and LSTM neural networks) provide the best performance (around 0.42-0.91% with NMAE for point forecasts). Furthermore, this same combination is the more robust among all the models, as they show a low degree of variability considering their performance over all the simulations carried out in the study. The same conclusions are drawn for short-term prediction horizons, where the models have been evaluated using data divided in hourly intervals. Additionally, evaluating the performance of the models under varied operational conditions reduces the influence of the dataset on the results. This way, any proposed statistical WPF model can be easily compared by just following the same set-up described even with a different wind power dataset belonging to the same region, as long as relevant features such as the resolution and length of data remain unchanged. If this type of information were publicly available, it could offset errors and bias of in-house models, which are not typically shared, and consequently improve the practice of WPFs and reduce epistemic uncertainty in

modelling.

The potential of this benchmark can be further improved by considering farms located in other wind regions to study the scalability of the results (for instance, using publicly available data of Sotavento wind farm (Sotavento Wind Farm, 2022)), as well as considering existing spatio-temporal relations between farms (Lenzi et al., 2018; Sun et al., 2019) and the implementation of these models with turbine-level data (González Sopeña et al., 2021b). Furthermore, strategies for decomposing wind data in real time can be developed to study the effectiveness of decomposition-based hybrid models for near real-time or online applications (Wang and Wu, 2016; Lu et al., 2021).

In the next chapter, turbine-level SCADA data are used as the starting point in order to leverage the spatio-temporal relationships of wind turbines by clustering relevant features, and later apply such clustering process for a VMD-GRU model in which the decomposition is analyzed from a practical point of view to ensure avoiding the leakage of information during the model training stage. The main guidelines of this benchmark are applied for the evaluation of this cluster-based approach.

Chapter 5

Enhancing wind power forecasting using turbine-level data

The benchmarking framework developed in the previous chapter has shown that variational mode decomposition combined with advanced recurrent network architectures provide the best model performance among the tested models considering the data extracted from the Irish wind farms. In this chapter, we first show how turbine-level forecasting models can lead to more accurate farm-level forecasts for short-term horizons. However, this approach can increase considerably the total computational cost required to predict the power generation of the whole wind farm (especially for larger ones). Therefore, a middle ground between forecast skill and computational cost can be found by grouping turbines to build cluster-level forecasting models using a VMD-GRU model as baseline, following the guidelines presented in chapter 4. Thus, this model (and in general the decomposition-based hybrid models presented in the previous chapters) is taken into consideration together with aspects to perform such a decomposition realistically to obtain near real-time wind power forecasts.

5.1 Introduction

As presented in the literature review (section 1.4), improvement of WPF models can be achieved following different strategies, for instance, by applying improved

optimization algorithms (Lu et al., 2021), feature selection (Li et al., 2015; Feng et al., 2017), or hybrid methodologies (Qian et al., 2019). However, innovation does not only stem from further enhancing the modelling of WPF models, but by using near real-time data coming from acquisition systems such as remote sensing (Würth et al., 2019; Wilczak et al., 2019) and SCADA systems (Lin and Liu, 2020). Regarding the latter case, SCADA data are collected individually for turbines, but they are frequently not used to build more accurate forecasting turbine-level models (Gilbert et al., 2019). This same reference proposes two different methods two exploit the benefits of turbine-level data: first, a feature engineering approach where forecasts at a turbine-level are used as an additional input for a wind farm-level model, and second, a wind farm-level forecast is produced from the joint predictive distribution of the individual turbines. However, building individual turbine models may come with an increased computational cost (Huang et al., 2020; Mehrjoo et al., 2021), while simultaneously ignoring any possible spatio-temporal relationships between turbines (Ezzat, 2020). Thus, wind turbines can be classified into homogeneous groups using clustering algorithms. For instance, Mehrjoo et al. (2021) use k-means clustering to group wind turbines, and Yang et al. (2021c) fuzzy C-means clustering to classify turbines based on the wind speed-power curve of each turbine.

In order to verify the benefits of using turbine-level data to obtain WPFs, section 5.2 presents a case study where a comparison using turbine- and wind farm-level data is performed. The forecasting models have been implemented under the same conditions to differentiate the effects of the data on the forecast skill from other possible sources. Later, a cluster-based approach using mode decomposition models is developed in section 5.3 to leverage near real-time wind power data (González Sopeña et al., 2022a). The clustering process is performed applying the *density-based spatial clustering of applications with noise* (DBSCAN) algorithm (Ester et al., 1996). To the best of our knowledge, this algorithm has been applied in the WPF literature to detect anomalies in wind power data (Yan et al., 2019; Zhou et al., 2019), but not to group turbines in a wind farm. Furthermore, to effectively implement mode decomposition models to obtain near real-time forecasts, practical aspects such as new input generation are also addressed in this chapter. An example case using SCADA

data collected in one of the Irish wind farms is presented to show the benefits of using the cluster-based approach.

5.2 Improving wind power forecasting with turbine-level data

Turbine-level data can be leveraged to improve the skill of WPFs. Let us show the veracity of this statement with a case study run in the two wind farms comparing forecasts using turbine- and aggregated wind farm-level data. To make this comparison as comprehensive as possible, the forecasting models are implemented under the same conditions to isolate the effects of the data with respect to any other variables which might influence model performance.

An ensemble consisting of five VMD-ELM models (González Sopeña et al., 2020) is chosen to train the models. As stated in section 1.4.1.2, ELM is a variation of a single hidden-layer FFNN where the first weight matrix is randomly assigned instead of using gradient-based learning methods (Huang et al., 2006). ELMs are chosen to speed up the training process, as they have a faster training speed compared to FFNNs trained with the backpropagation algorithm (Tang et al., 2015). This is relevant as the number of models to be trained are proportional to the number of turbines found in every wind farm. Even if the five VMD-ELM models of the ensemble are trained using the same data, the weights learned during the training process of each VMD-ELM model will differ from each other, resulting in slightly different WPFs for every time step.

Considering this forecasting model, the turbine approach is implemented by fitting an independent ensemble of VMD-ELM models for each turbine, and the final wind power generation output will be estimated aggregating the forecasts provided by the model fitted to each turbine (Figure 5.1). Thus, the wind farm I will have a total of 11 turbine-level models, and the WPFs of wind farm II will be computed aggregating the forecasts of 17 turbine-level models. This approach is benchmarked using the same model fitted with data at a wind farm level.

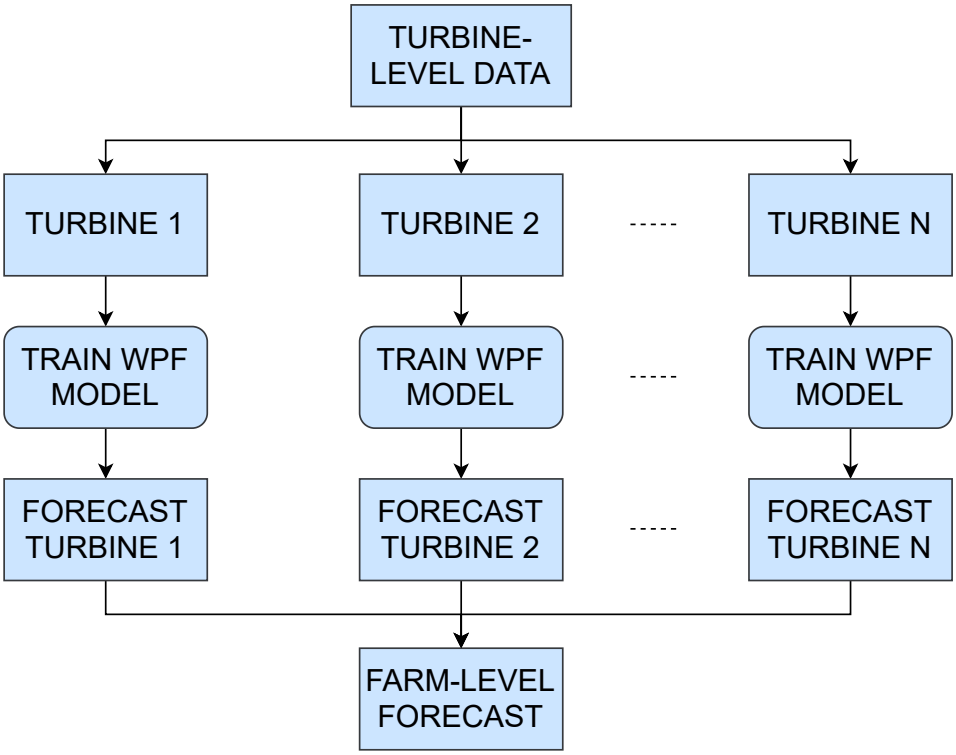

Figure 5.1: Turbine-level approach.

The performance of the approach is measured by means of the improvement (Eq. 3.7) of point WPFs with respect to NMAE and NRMSE, and calculated on a daily basis with the aim of capturing in as much detail the influence of the different power generation outputs displayed in the testing set.

Figure 5.2 shows the improvement using both NMAE and NRMSE for the wind farm I. In this case, even if the use of turbine-level data increases the overall prediction performance, there are a few days where the turbine-data approach does not outperform the wind farm-data approach. For instance, this pattern can be observed for the NMAE in periods such as the day 7 for 2-h ahead predictions (around -7%), and the day 5 for 8-h ahead predictions (around -13%). On the other hand, when the NRMSE is measuring the performance, this behavior is more significant for the day 4 for 6-h ahead forecasts, and once again the day 5 for 8-h ahead predictions. Nonetheless, the underperformance observed in these situations is compensated by a larger improvement in other subintervals over the testing set. Using the NMAE as the metric, an improvement over 30% is found in the day 1 of the testing set for 7-h ahead forecasts, and slightly over 25% in the day 6 for 4-h ahead forecasts, day

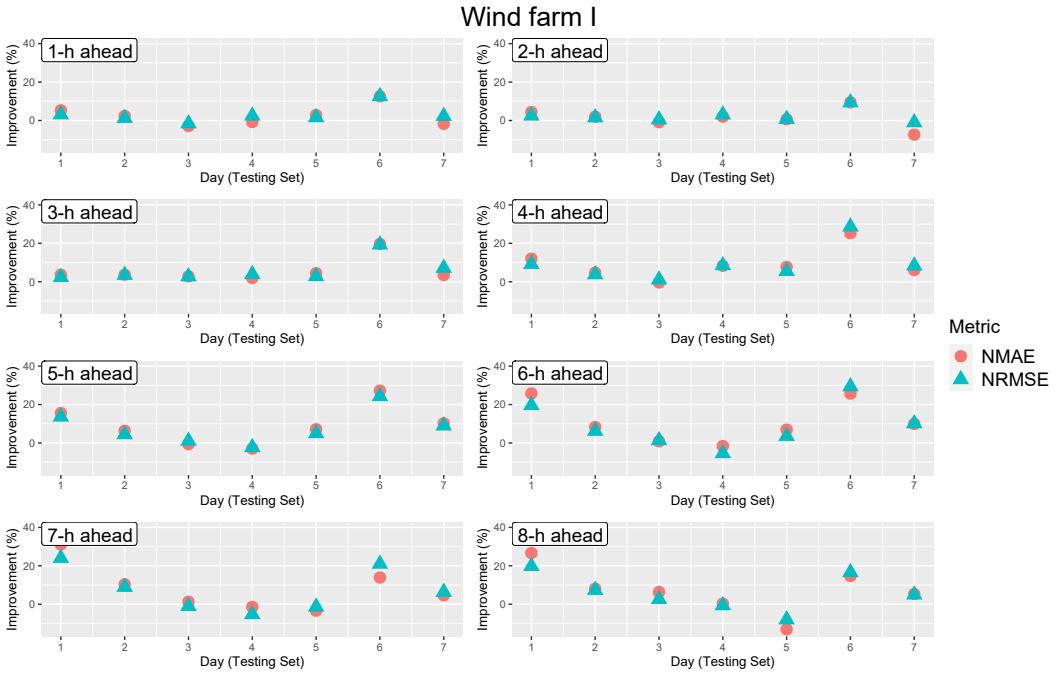

Figure 5.2: Improvement (%) provided by the turbine approach (Wind farm I).

6 for 5-h ahead forecasts, days 1 and 6 for 6-h ahead forecasts, and day 1 for 7-h and 8-h ahead predictions. With regard to the NRMSE, an improvement over 25% is observed in the day 6 for 4-h and 6-h ahead predictions, whereas an improvement over 20% can be noticed in the day 5 for 6-h ahead forecasts, and days 1 and 6 for 7-h ahead predictions.

Fig. 5.3 shows the improvement using NMAE and NRMSE for the wind farm II. With the exception of two of the daily-averaged NMAE values in which the farm-level data have provided a slightly larger forecast accuracy (day 4 for 1-h and 3-h ahead forecasts), the models trained by the turbine-level data show a better performance. The larger improvement using the NMAE is observed in the day 7 of the testing set for 4-h ahead predictions and the days 4, 5, and 6 for 6-h ahead predictions, where an improvement of approximately 20% is reached by using turbine-level data. With respect to the daily-averaged scores of NRMSE, the turbine-data approach shows an improvement for every subinterval of the testing set. In this case, the maximum improvements are found in the day 6 for 2-h ahead forecasts, days 5 and 6 for 6-h ahead forecasts, and day 7 for 7-h ahead forecasts, reaching in these subintervals an improvement around 18%.

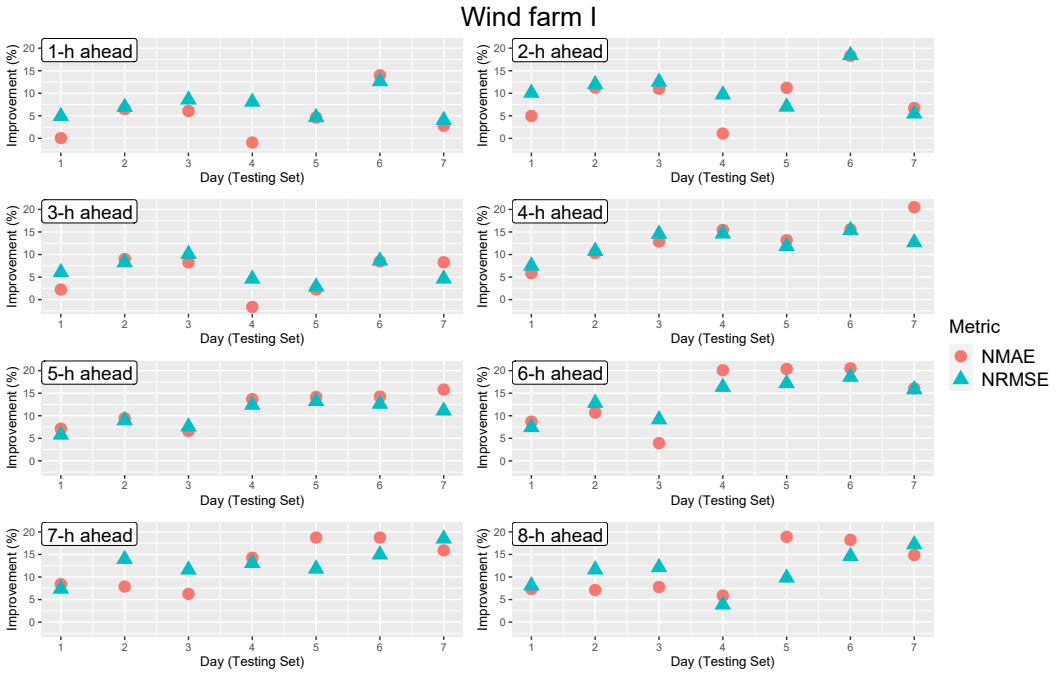

Figure 5.3: Improvement (%) provided by the turbine approach (Wind farm II).

5.3 A cluster-based approach using mode decomposition models

While the turbine-level approach proves to be effective to improve the final forecasting accuracy, it is much more expensive from a computational point of view compared to modelling the wind power generation of the farm as a whole. Furthermore, new inputs can easily be provided to most statistical models as new data arrive (e.g. wind power measurements). However, this is not true for decomposition-based hybrid models (Qian et al., 2019), as the decomposition must be updated with the arrival of new data to generate new inputs. This is a delicate process as the decomposed subseries are sensitive to the original data, potentially leading to significant changes in the subseries modelled with the new data (Wang and Wu, 2016). Therefore, input update strategies must be considered to provide a fair assessment in the evaluation stage (Gendeel et al., 2018; Qian et al., 2019). In addition, decomposition algorithms such as VMD are heavily biased by some initial user-defined parameters, affecting the decomposition and consequently the forecasts. Let us address these points hereunder.

5.3.1 Methodology

The proposed cluster-based approach is shown in Figure 5.4. First, high-resolution data from SCADA systems (such as wind power and wind speed) are collected during the preprocessing stage. Turbines are then grouped using the DBSCAN algorithm (Ester et al., 1996). For every cluster, the wind power measurements recorded from every turbine are aggregated and decomposition using the VMD algorithm (Dragomiretskiy and Zosso, 2013) is applied to the cluster-level wind power measurements.

In the training stage, a forecasting model is built for every mode in each cluster. Depending on the energy explained by the mode, the forecasting model chosen will be either a GRU network, if the mode is explaining a significant amount of the energy identified in the original data, or a simple FFNN otherwise.

Wind farm power forecasts are determined in the implementation stage. For every cluster, the forecasts of every mode are aggregated, and later the cluster-level forecasts are once again aggregated to determine the WPF. When new measurements arrive from the SCADA systems, more WPFs will be provided to the user. The decomposition must then be updated to generate the input vectors for all the forecasting models. The approach is later evaluated following the guidelines presented in chapter 4.

All the elements present in this cluster-based approach are described thoroughly in the remainder of this section.

5.3.1.1 Density-based spatial clustering of applications with noise

DBSCAN (Ester et al., 1996) is a density-based clustering algorithm. Considering a certain radius ϵ, the ϵ-neighborhood of a point p is defined as (Khan et al., 2014):

$$N_\epsilon = \{q \in D / dist(p, q) < \epsilon\} \tag{5.1}$$

where D represents the data. If a point contains at least a user-specified threshold number of neighbors *minPts*, that point is defined as a *core point*. The DBSCAN algorithm tries to find the areas which satisfy this condition. Those points belonging

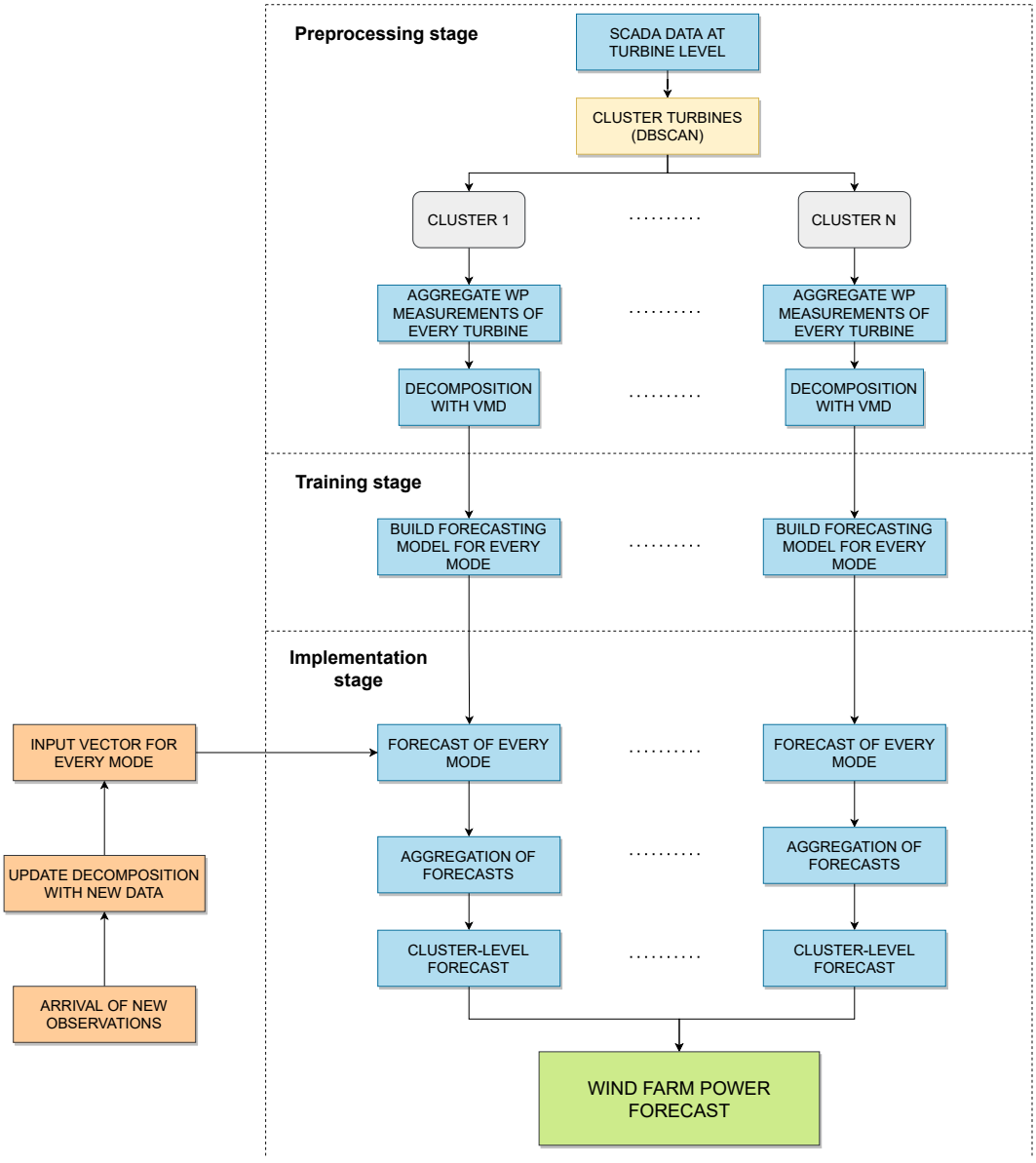

Figure 5.4: Flowchart of the proposed cluster-based approach.

to the ϵ-neighborhood are *direct density reachable*, and if any of this neighbors is a core point as well, the points within its vicinity are defined as *density reachable*. The neighbors of such set are known as *border points* with respect to the initial ϵ-neighborhood. If any given point is not density reachable from any core point, it will not belong to any cluster and will be labelled as *noise*.

Once set a radius ϵ and a minimum number of neighbors *minPts*, the DBSCAN clustering algorithm computes clusters following these definitions, except for those border points which belong to several clusters (Schubert et al., 2017). The algorithm iterates linearly through the dataset looking for points that have not been checked yet. If the ϵ-neighborhood exceeds the value set for *minPts*, it gets labelled as a core point, otherwise as noise. Then, all the neighbors are added as part of the cluster. If any of the neighbors is defined as a core point as well, then the points within its ϵ-neighborhood are also aggregated to the cluster. Those points assigned to a cluster will be in that case skipped during the iteration process, which ends once all the points are labelled.

5.3.1.2 Practical-based variational mode decomposition

In order to implement the VMD algorithm (section 4.2.1.1) to forecast near real-time wind power, two practical aspects have to be addressed during the evaluation (and subsequently implementation) of the model: 1) the sensitivity of the decomposition algorithm to some user-defined parameters, and 2) the generation of new inputs from upcoming data.

Regarding the first practical challenge, the algorithm is highly influenced by the constraint parameter α' and the number of modes k (Hong et al., 2018; Cui et al., 2019). Following up from the description of the VMD algorithm in section 4.2.1.1, the modes u_k and their respective center frequencies ω_k are updated simultaneously with Eqs. 4.3 and 4.4 respectively. The parameters α' and k are found in these equations, so they must be carefully chosen to ensure a decomposition which is beneficial to compute highly accurate WPFs.

The second practical aspect to be addressed is the necessity of updating the decomposition with the influx of new data as the modes are very sensitive to the

original data (Wang and Wu, 2016), as well as the length of the sample (Qian et al., 2019). Let us suppose we have a wind power time series dataset $Y = \{y_1, \ldots, y_L, y_{L+1}, \ldots, y_{L+L'}\}$ conformed by $L + L'$ observations equally spaced in time. The usual strategy would be applied as follows:

1. Decompose the original wind power time series Y to obtain the modes.
2. Keep the first L observations for the training and validation sets, and the following $\{y_{L+1}, \ldots, y_{L+L'}\}$ observations for the testing set.
3. To set up the supervised learning problem, create sets of input and output vectors of length m and n respectively for the training and validation sets. In the same way, create input vectors of length m with the testing data. Repeat this for all modes.
4. Train and validate the model for every mode using the respective pairs of input and output vectors created in the previous step.
5. Obtain forecasts of every mode using the input vectors built with the testing data.
6. Aggregate forecasts of all modes to obtain the WPF. Evaluate the forecasts using adequate performance evaluation metrics.

This strategy would be appropriate if no decomposition algorithm is applied to the data, as the future inputs could simply be provided to the model to generate forecasts. However, this is not the case for decomposition-based hybrid models, as the future observations must be decomposed on a step by step basis while ensuring these data are not leaked into the training stage. To address this issue, the original strategy is modified as follows:

1. Decompose the first L observations of the original time series.
2. Divide the modes into training and validation sets considering these first L observations.
3. The supervised learning problem is set up by creating sets of input and output vectors of length m and n for the training and validation datasets respectively.

4. Train and validate the model for every mode using the pairs of input and output vectors generated in the previous step.

5. To reduce the sensitivity to the original observations and the length of decomposition, we keep the same length of data to be decomposed. Every time a new observation arrives, the oldest observation is removed, and the new observation is added. That is, when the observation y_{L+1} comes, we apply the decomposition algorithm to $\{y_2, \ldots, y_{L+1}\}$, and take the last m observations to create the input vector. We repeat this step till there are no observations remaining in the testing set.

6. Generate forecasts for every mode using the input vectors built with the testing data.

7. Aggregate forecasts of all the modes to predict wind power generation, and evaluate them with the corresponding metrics stated in the experimental design described in section 4.2.4.

Proceeding this way allow us to ensure that no future data are involved during the evaluation stage of the model.

5.3.1.3 Energy-based model selection

Decomposing the original wind power data into simpler modes can increase the accuracy of WPFs, but at the expense of increasing the training time of the model proportionally to the number of modes decomposed. While this can be partially addressed using parallel computing (Wang et al., 2019c), the computational cost would remain unchanged. However, the predictions obtained for those modes explaining less energy from the original wind power time series will be less relevant for the final aggregated WPF. Therefore, the less relevant modes from an energy point of view will be trained with a regular FFNN (Fine, 2006) instead of a GRU network to speed up the training time. Note that energy is chosen as the model selection feature as it is time invariant.

The energy explained for every mode u_k can be determined over the training dataset calculating the energy explained by that mode with respect to the original

wind power time series:

$$EE_k = \frac{\sum_{n=1}^{N} |u_k(n)|^2}{\sum_{n=1}^{N} |y(n)|^2} \tag{5.2}$$

where $\sum_{n=1}^{N} |u_k(n)|^2$ is the energy of the k-mode (as a finite discrete-time signal) and $\sum_{n=1}^{N} |y(n)|^2$ the energy of the original wind power time series.

Furthermore, those modes explaining less variance are representing the noisier and therefore more complex components, meaning that the dynamics of the less representative modes will be more difficult to identify (Dong et al., 2019), so applying a FFNN instead of a GRU neuron will have very little impact in terms of accuracy on the final forecast. To quantify that, the complexity of each mode is calculated using an entropy measure known as approximate entropy (ApEn) (Pincus, 1991). This entropy measure is adequate as ApEn can be applied effectively to raw time series data (in the sense that the data do not have to be preprocessed in any way) without assuming any type of model, and can be applied to equally spaced measurements (Delgado-Bonal and Marshak, 2019).

5.3.2 Results & discussion

First, some practical aspects from the mode decomposition will be examined numerically for one wind turbine to analyze how suitable these models are to estimate near real-time forecasts. These results are later taken into consideration for the implementation of the cluster-based approach, whose efficiency is then benchmarked against a single farm-level forecasting model.

5.3.2.1 Practical aspects of mode decomposition

The first factor to fairly evaluate a mode decomposition model is providing correctly the inputs to the forecasting model, preventing that future data are leaked into the initial decomposition. This is achieved by recurrently decomposing the data with the arrival of new observations (as stated in section 5.3.1.2) to build the updated input vectors. In the first iteration, the whole data are decomposed and divided into training and validation datasets, which are fed to the forecasting model to set up its parameters. After that, we are interested in retrieving those values which will

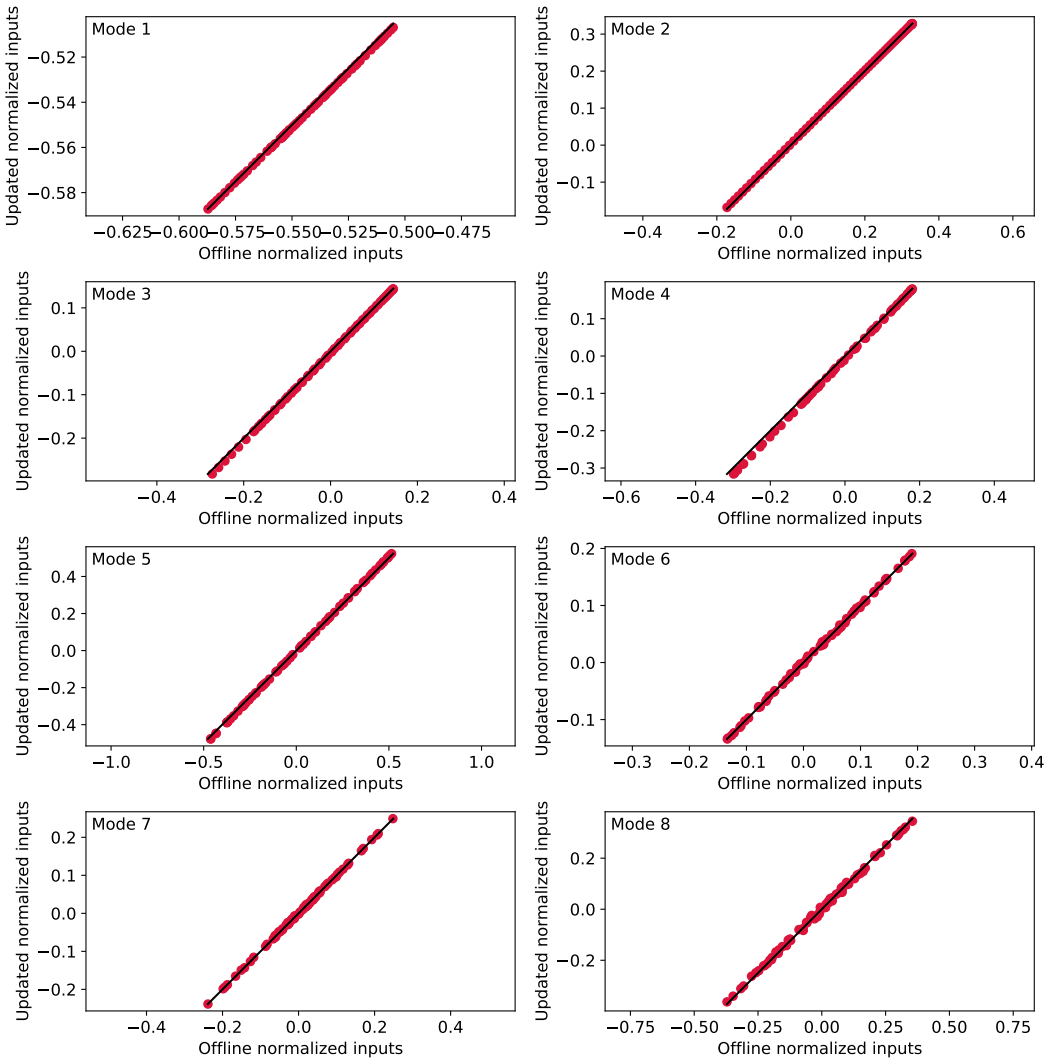

Figure 5.5: Input comparison with offline input generation and updating inputs stepwise.

constitute the input vector for the next set of forecasts. While some deviations are expected, the input vectors remain very similar compared to those obtained offline when the same sample size is kept for the subsequent decompositions (Figure 5.5). Thus, we can ensure that the forecasts provide unbiased results as the information contained in the future observations (i.e., the testing data) is not leaked into the initial decomposition, and therefore not compromising the training of the forecasting models.

On another note, the forecast skill is affected by the parameter α' and the number of modes chosen to decompose the wind power data. To verify that, we set up an initial experiment decomposing the data from one of the turbines and run 10 simulations for a set of reasonable $[\alpha', k]$ values (Figure 5.6). For shorter prediction

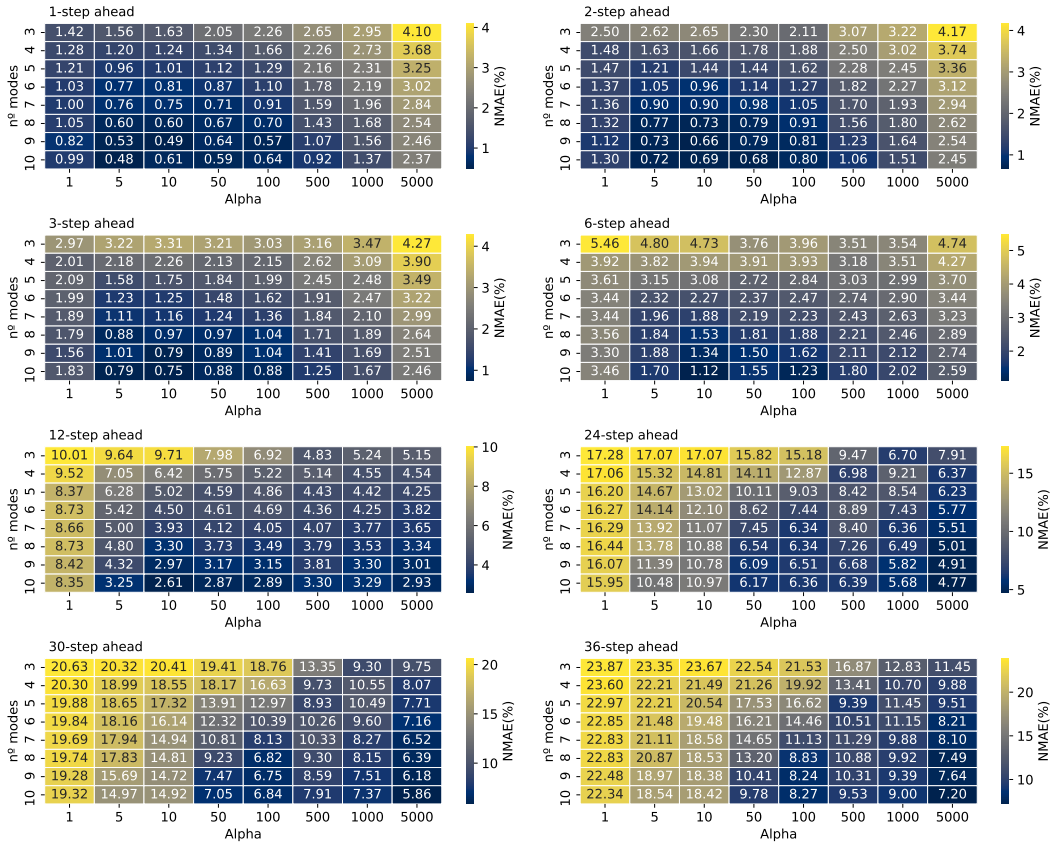

Figure 5.6: Average NMAE (%) for different decomposition levels and values of α'. 5 simulations are run for 1 dataset.

horizons, lower values of α' turn into more accurate WPFs and usually increase with a larger decomposition level. Nonetheless, the forecast accuracy with low α' values deteriorates at a higher rate for larger prediction horizons, causing that those models built with larger α' values ensure better forecasts as we move forward in the forecast vector. Therefore, the pair $[\alpha', k]$ must be chosen in such a way that a) ensures reasonable good forecasts for a certain look-ahead time, and b) keeps a realistic computational complexity, meaning that a very large decomposition level might not be worth if the forecast accuracy only increases slightly compared to a lower decomposition level.

Let us choose three levels of α' (10, 500, and 5000) and a fixed value of $k = 8$ to further analyze the decomposition and the power spectral density (PSD). The decomposition of these three pairs of $[\alpha', k]$ values is shown in Figure 5.7. In all cases, the first mode includes primarily the trend of the original data. After that, the modes are formed by higher frequencies present in the data. This is indeed

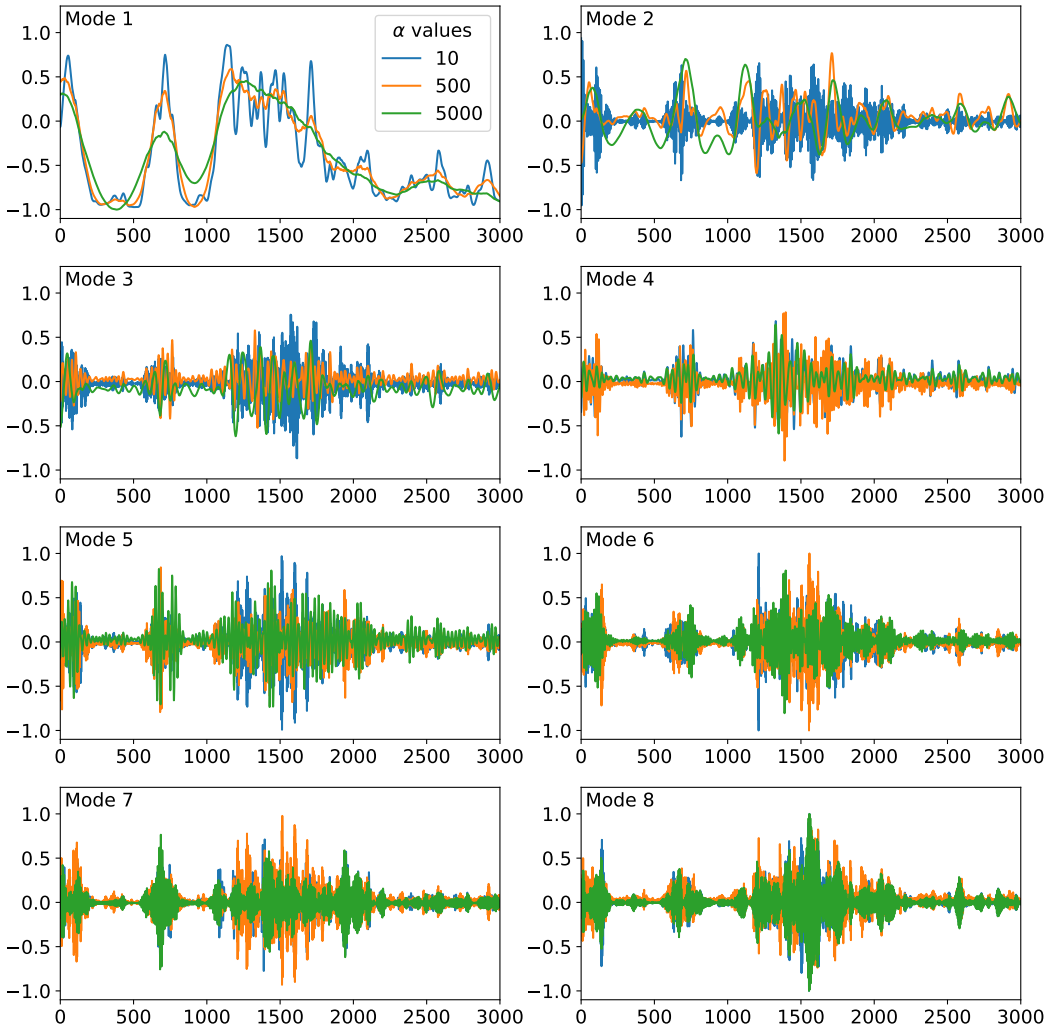

Figure 5.7: Normalized modes for a decomposition level k = 8 and different values of α'.

corroborated when looking at the PSD of every mode (Figure 5.8), even if the energy is unevenly distributed for every level of α'. Thus, the energy for lower frequencies is concentrated in the first mode for a small value of α', whereas it is distributed among several modes for larger values of α': modes 1 and 2 for $\alpha' = 500$, and modes 1-3 for $\alpha' = 5000$. In this two cases, we observe that the energy shifts gradually to higher frequencies for the latter modes. That is not the case for $\alpha' = 10$, as no regular pattern is visible between the energy and the respective modes.

In addition to these practical considerations for mode decomposition, we have to keep in mind that transforming highly complex wind power data into several less complex subseries is the main goal to be achieved. However, some of the modes could be explaining noisier components of the original wind power data, hindering the ability of any model to build highly accurate forecasts, as their level of complexity

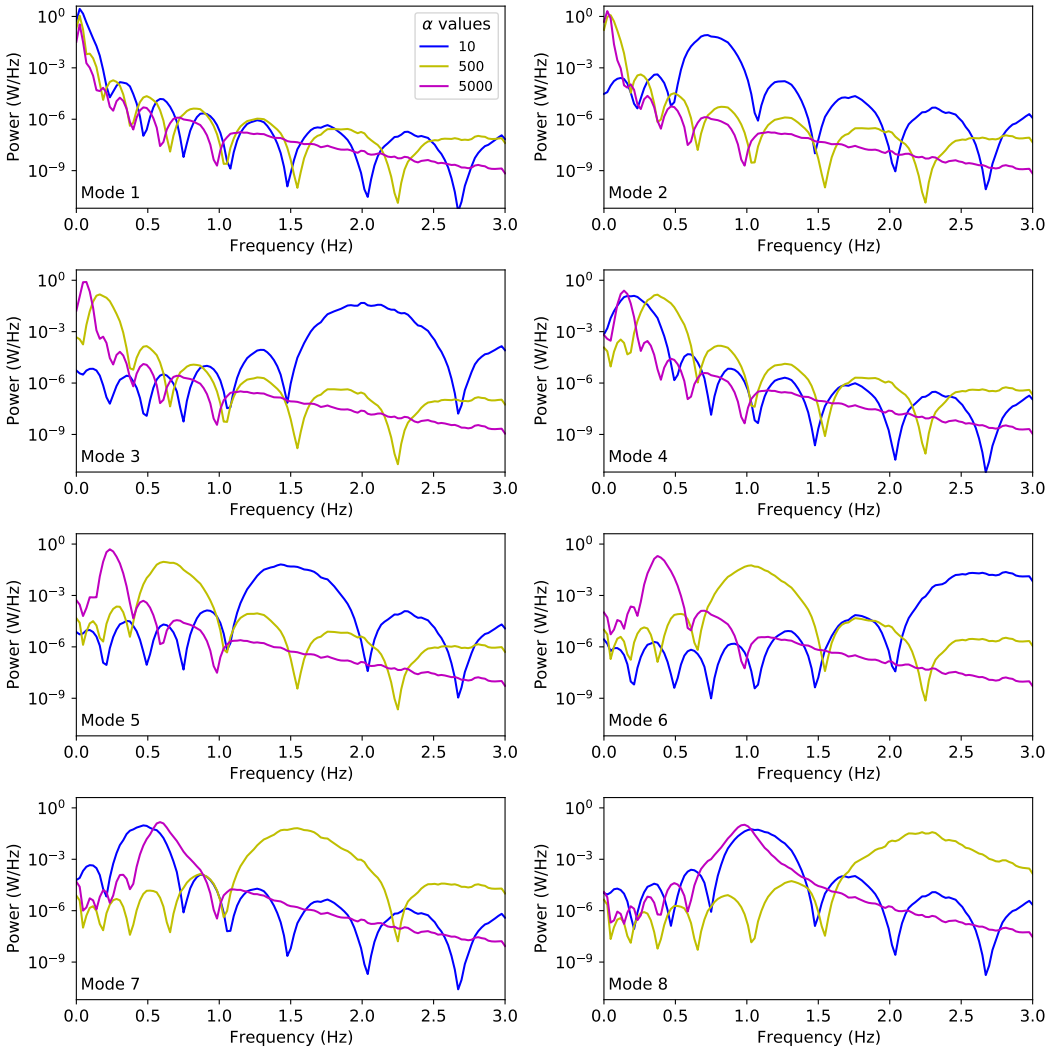

Figure 5.8: PSD plots for a decomposition level k = 8 and different values of α'.

will certainly be larger. Thus, switching to a simpler forecasting model (in this case, a basic FFNN) instead of a GRU neuron to forecast those modes would barely impact the final WPF while reducing the training time of the total network (as explained in section 5.3.1.3). To verify that, we calculate the explained energy with respect to the original signal and ApEn for the three selected α' values, shown in Table 5.1. The first mode explains most of the energy contained in the original data, while reducing significantly its complexity, which has an ApEn value of 0.652 (compared to 0.073, 0.029 and 0.016 respectively for the three values of α'). For $\alpha' = 10$, the rest of modes are more complex with respect to their counterparts for higher α', and explain less than 1% of the energy contained in the wind power signal (with the exception of mode 4, which explains 2.08% of the original signal). Conversely, mode

Table 5.1: Explained energy (%) and ApEn for a decomposition level k = 8 and different values of α'.

	$\alpha' = 10$		$\alpha' = 500$		$\alpha' = 5000$	
Mode	EE (%)	ApEn	EE (%)	ApEn	EE (%)	ApEn
1	92.54	0.073	80.63	0.029	73.30	0.016
2	0.37	0.587	7.63	0.198	9.30	0.082
3	0.15	0.812	1.68	0.519	3.08	0.265
4	2.08	0.530	0.76	0.580	1.22	0.480
5	0.20	0.780	0.45	0.585	0.70	0.566
6	0.11	0.657	0.29	0.649	0.44	0.574
7	0.71	0.582	0.17	0.793	0.31	0.534
8	0.21	0.627	0.11	0.793	0.16	0.483
Total	96.44	-	91.72	-	88.51	-

2 still shows low complexity compared to the original signal and explains a significant amount of energy for $\alpha' = 500$ and $\alpha' = 5000$ (7.63% and 9.30%). The rest of modes show a similar or higher complexity, but describe a small part of the original signal. Therefore, a threshold of 5% is chosen to select either a GRU network or a FFNN to train the mode.

5.3.2.2 Case study

The DBSCAN algorithm is applied using the SCADA data collected at turbine-level for a period of two months. Three variables are used for the clustering process: wind power, wind speed, and rotational speed. Similarly to the previous chapter, NWP data are neither considered for clustering nor as an input for the forecasting models, as the uncertainty introduced by these variables is greater than the variability found on shorter time scales (Tawn and Browell, 2022). First, we need to select what variables are more relevant to cluster the turbines. For that, variables are chosen according to the silhouette coefficient (Rousseeuw, 1987), a metric which indicates how well the turbines are classified (and whose value can range between -1 and +1). Table 5.2 shows the silhouette coefficient where the clustering has been performed with different combinations of the mean and deviation values of the variables over the two-month period. The silhouette coefficient indicates that the strongest clustering comes from using the mean values of wind power, wind speed, and the deviation of wind power values of the whole period. Thus, these variables are selected to classify and cluster the turbines.

Table 5.2: Clustering variable selection (Silhouette coefficient)

Mean WP	Mean WS	Mean RS	WP dev	WS dev	RS dev	Silhouette coefficient
✓	✓	✓	✓	✓	✓	0.532
✓	✓	✓	✓	✓		0.628
✓	✓	✓	✓			0.566
✓	✓	✓		✓		0.524
✓	✓		✓	✓		0.719
✓	✓		✓			**0.808**
✓	✓			✓		0.633
✓	✓	✓				0.592

Afterwards, we find a sufficiently large value of ϵ so at least 90-95% of the turbines have one neighbor (Figure 5.9a). Figure 5.9b shows that two clusters are built with a large enough ϵ, with seven and ten turbines respectively. For every cluster, wind power measurements are aggregated and decomposed using the VMD algorithm. Then, following the same experimental design shown in section 4.2.4 to guarantee a fair evaluation process of the cluster-based approach, every model is run 10 times for every subset of data, so 50 simulations are performed in total, and the accuracy of the forecasts is measured using NMAE as metric (Eq. 3.1). The proposed approach is benchmarked against an individual farm-level model where the wind power measurements of all turbines are added to account for the wind farm as a whole. The decomposition of both farm- and cluster-level approaches are performed with $\alpha' = 500$ and $k = 8$, which have been proven to provide reasonable accurate forecasts as shown previously for one wind turbine.

The clustering is particularly important for shorter horizons, as the forecast errors accumulate the further we move ahead in the future, making the choice of approach (cluster- or farm-level) less relevant for longer prediction horizons. Thus, the analysis is focused on forecasts up to six steps ahead. In other words, WPFs up to 1-h ahead in 10-minute intervals. The distribution of NMAE values for 1- to 6-step ahead WPFs is shown in Figure 5.10. For all prediction horizons, the cluster-based approach provides in average more accurate WPFs, and the forecasts tend to fall slightly more centered within the interquartile range, indicating a higher degree of robustness for the cluster approach. Some outliers are found for both approaches when the prediction horizon is increased, due to the diverse nature of both training and testing sets presented to the models as well as the lower predictive ability of the models when looking further

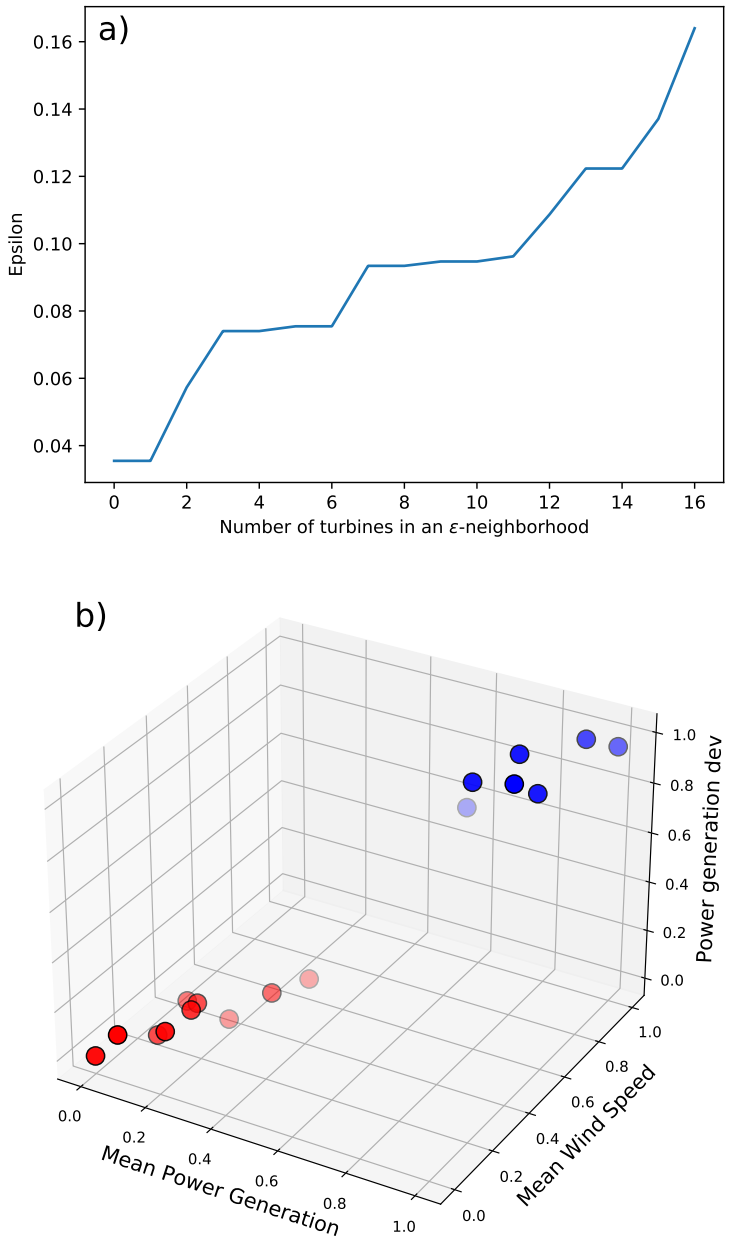

Figure 5.9: a) Relation between ϵ and turbines found in an ϵ-neighborhood, and b) turbine clustering using DBSCAN.

ahead in time. The average improvement with respect to NMAE (following the Eq. 3.7) is shown in Table 5.3. The improvement ranges from 5.55% for 2-step ahead (20-minute ahead) to 10.76% for 1-step ahead (10-minute ahead) forecasts. Beyond that, the forecast error increases for both methods, leading to more similar NMAE values for both approaches.

Therefore, we can assure that the use of high-resolution SCADA data can effectively improve the accuracy of WPFs for shorter time scales while keeping a reasonable computational cost, in the sense that only two cluster-level models have been necessary compared to the seventeen that would have been required if an independent model was built for each wind turbine.

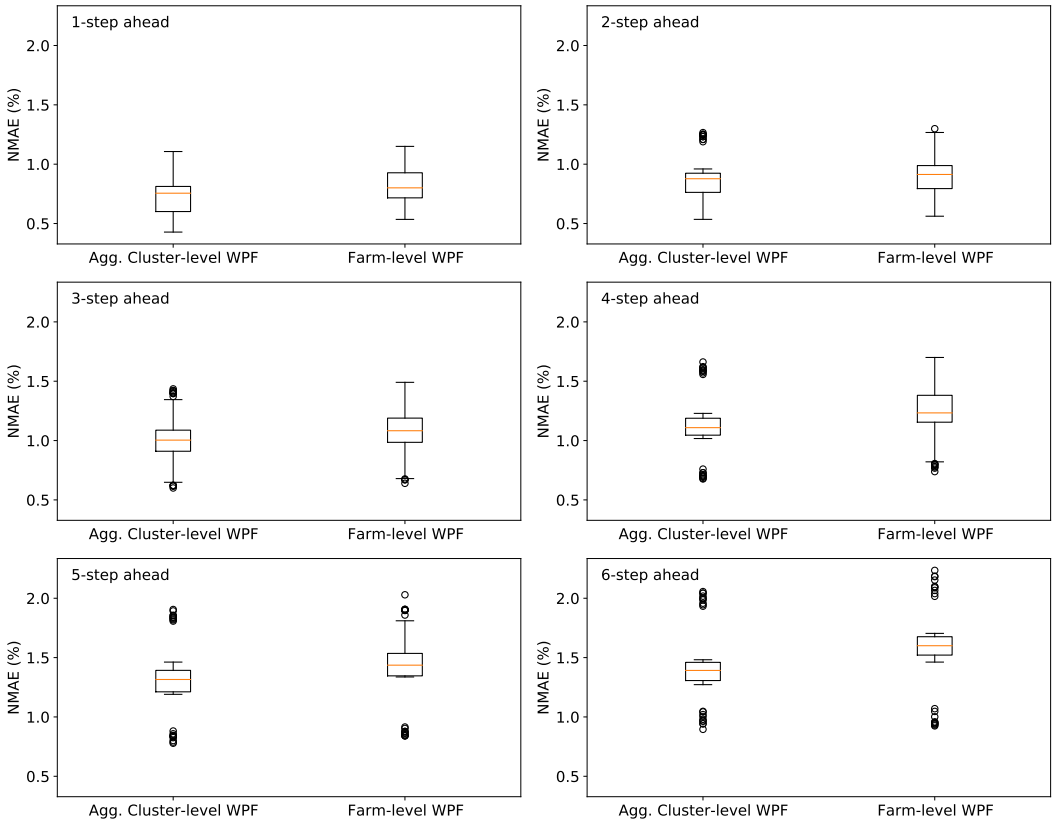

Figure 5.10: Distribution of NMAE values for different prediction horizons.

Table 5.3: Average NMAE values (%) and improvement (%) shown by the cluster approach.

	NMAE (Farm-level)	NMAE (Agg. Cluster)	NMAE Improvement
1-step ahead	0.83	0.74	10.76
2-step ahead	0.92	0.87	5.55
3-step ahead	1.07	1.01	6.35
4-step ahead	1.24	1.13	9.48
5-step ahead	1.42	1.32	6.87
6-step ahead	1.57	1.43	8.59

5.4 Conclusions

WPFs can benefit from new types of high-resolution data available in wind farms, such as SCADA data (Lin et al., 2020) to build more accurate forecasting models. This type of data is generally collected independently for every single turbine, allowing to build turbine-level forecasting models, whose forecasts can be aggregated at a later stage to provide a single forecast for the whole wind farm. However, the improvement of building individual turbine-level models comes with an extra computational cost. A middle ground between computational cost and forecast improvement can be established by classifying the turbines into clusters to build cluster-level WPF models. Thus, a cluster-based approach is implemented grouping the turbines according to the DBSCAN algorithm (Ester et al., 1996). Decomposition-based hybrid models are used in the forecasting stage, addressing practical challenges to correctly evaluate and implement them to produce near real-time WPFs, which can be posteriorly leveraged for trading in the electricity market (Conejo et al., 2010; Baringo and Conejo, 2015).

The main practical decomposition challenges have been addressed and tested in one single turbine, verifying that no future data are leaked into the training stage, by introducing some slight modifications during the decomposition of wind power data (section 5.3.1.2); and analyzing the influence of user-defined parameters required to compute the decomposition using VMD. After that, the effectiveness of the cluster-based approach has been verified using the SCADA data collected in the 17-turbine Irish wind farm. Two clusters have been identified using DBSCAN, and subsequently a forecasting model has been trained for each one of them. The WPFs of each cluster have posteriorly been aggregated to compare them to a farm-level model. Up to 6-step ahead, the accuracy is indeed higher, showing an improvement around 5-10% in terms of NMAE.

Further work should consider the implementation of clustering methods (not only limited to DBSCAN but other algorithms) with larger wind farms, in which the use of turbine-level models will hinder more notably the computational cost to obtain a farm-level WPF. Furthermore, methods to identify and update the clusters in real time can be considered (Bernardoni et al., 2020), as aspects such as the operational

state of the turbines (Yang et al., 2021c) and the weather conditions (Orlov et al., 2020) will change over time. Additionally, the uncertainty of the forecasts introduced by the clusters should be taken into consideration by building PIs (for instance, using QR-based intervals as shown in the previous chapters) or alternatively other probabilistic methodologies (Zhang et al., 2014b; Gneiting and Katzfuss, 2014; Abdar et al., 2021).

In the next chapter, we follow up with the idea of optimizing the available resources, but not in terms of data, rather from the perspective of using algorithms adapted to new and more energy-efficient computational resources represented by neuromorphic devices.

Chapter 6

Neuromorphic computing for wind power forecasting

In terms of modelling, many machine/deep learning based methodologies have been successfully applied for short-term wind power forecasting (Table 1.3). However, the modelling and training of these architectures might take many computational resources, thus reducing the possibility of implementing such algorithms for real-/near real-time forecasting scenarios. Emerging computational architectures, such as neuromorphic computing, have the potential of real-time learning using brain-inspired models characterized by their low latency and low energy consumption compared to more conventional approaches. In particular, we introduce spiking neural networks (also referred as the third generation of artificial neural networks), in which the information is propagated through the network by spikes, imitating the energy-efficient nature of the brain.

In this chapter, we first introduce the neuromorphic paradigm in section 6.1. The experimental design to develop spiking neural networks is described in section 6.2. Afterwards, this procedure is applied to short-term wind power forecasting (González Sopeña et al., 2022d) in section 6.3, and the final remarks of this chapter are found in section 6.4.

6.1 Introduction to neuromorphic computing

A large number of ML/DL models have been developed and applied to time series data of varied nature for tasks such as forecasting (Lim and Zohren, 2021), classification (Fawaz et al., 2019), clustering (Ma et al., 2019), and anomaly detection (Munir et al., 2018). While ML/DL models have high capability to solve these tasks, such models may be associated with a high computational cost, which is particularly critical for edge computing applications (Wang et al., 2020d), such as those implemented for renewable energy sources (Li et al., 2018). Neuromorphic computing aims to develop algorithms inspired in the energy-efficient nature of the brain to overcome this issue, such as SNNs (Maass, 1997). The development of specific hardware for spiking-based algorithms such as Intel's Loihi (Lin et al., 2018a) allow us to implement and evaluate SNN models. Furthermore, SNNs are well-suited for online learning (Lobo et al., 2020), for instance using online approximations of the backpropagation algorithm (Tavanaei and Maida, 2019; Bellec et al., 2020) or leveraging the features of spiking network models such as evolving SNNs (Schliebs and Kasabov, 2013; Kasabov et al., 2016).

Spiking deep learning algorithms can be broadly divided into *online* and *offline* approaches (Davies et al., 2021). Online approaches first implement an SNN in neuromorphic hardware, leveraging on-chip plasticity to train the spiking network and evolve its parameters with the arrival of new data (Stewart et al., 2020). On the other hand, the SNN is trained before deploying the model for offline approaches. These offline approaches can be further divided into two categories, considering how the training stage is performed. On the one hand, a conventional ANN is trained using the backpropagation algorithm, and later the parameters are mapped into an equivalent SNN model (Diehl et al., 2015). This approach is known as *ANN-to-SNN conversion*. On the other hand, a direct training approach uses a variation of error backpropagation to optimize directly the parameters of an SNN (Taherkhani et al., 2015; Wu et al., 2018a).

On top of that, efforts have been made to develop software platforms to facilitate the implementation of SNN-based applications. For instance, Nengo (Bekolay et al., 2014) is a software based on the principles of the Neural Engineering Framework

(NEF), a theoretical framework to implement large-scale neural models with cognitive abilities (Eliasmith and Anderson, 2003). This same software was later extended with the sister library NengoDL (Rasmussen, 2019), aiming to combine the principles of neuromorphic modelling with the well-known deep learning framework TensorFlow (Abadi et al., 2015) to build deep spiking neural models by ANN-to-SNN conversion. Another ANN-to-SNN conversion framework is the SNN conversion toolbox (SNN-TB), based on the theoretical basis proposed by Rueckauer et al. (2017), which describes the approximation of SNN firing rates to ANN activations and the conversion of common neural operations into spiking ones. Alternatively, other frameworks allow to train directly SNNs, such as the Spike Layer Error Reassignment (SLAYER) algorithm proposed by Shrestha and Orchard (2018), and the spatiotemporal backpropagation training framework established by Wu et al. (2018a). Recently, in October 2021, Intel's Neuromorphic Computing Lab released the first version of Lava (Intel's Neuromorphic Computing Lab, 2021), an open-source software framework to implement neuromorphic applications for the Intel Loihi architecture (Davies et al., 2018).

Thus, we explore the opportunities offered by neuromorphic computing to build up more efficient algorithms using spiking-based neural models for WPF. In particular, we use the framework provided by NengoDL to build spiking neuron models, which allow us to extend ML/DL models by mapping the parameters of trained ANNs, and NengoLoihi, a library to implement such models on Loihi hardware.

6.2 Methodology

The design of a non-spiking ANN model is key to successfully map its parameters into an spiking network. The parameters of the network must be carefully tuned to ensure a minimal loss of performance during the conversion, and the architecture of the model must be tailored to subsequently build the network on Loihi hardware. Therefore, we proceed as follows to build and evaluate the performance of our SNN model:

1. Build the non-spiking neural model as usual. The network must be designed taking into account specific requirements for its implementation on Loihi hard-

ware, such as the communication with the chip. Details of the model architecture are found later in this section (see section 6.2.2).

2. Train the equivalent rate-based network with the methodology described by Hunsberger and Eliasmith (2016), the default method implemented in NengoDL to train SNNs. This training process is explained in section 6.2.1.

3. Replace the activation functions for their spiking counterparts (in particular, we will use spiking ReLU activations) for inference. The activation profile of this function is restricted by the discretization required for the Loihi chip (DeWolf et al., 2020), leading to discrepancies compared to the theoretical spiking ReLU activation (Figure 6.2). Such discrepancy increases for higher firing rates due to this discretization. Furthermore, the Loihi chip can only fire a spike once per timestep, limiting its firing rate at a maximum of 1000 Hz. This constraint does not exist otherwise, and multiple spikes could in theory be fired simultaneously and exceed that value (Applied Brain Research, 2021).

4. Run the network using the NengoDL framework, setting parameters such as the number of timesteps that each input will be presented to the spiking model, allowing the network to settle and spike in the given timeframe; and the firing rate scale, letting the network to spike at a higher rate. These preliminary results will help us monitor the neural activities and tune the parameters of the SNN.

5. Once an acceptable model performance is reached, we need to configure some additional parameters to set up the SNN for Loihi and simulate it for either Loihi hardware or alternatively the emulator (Voelker and Eliasmith, 2020) to replicate the chip's behavior.

6. Collect the results to evaluate them. One-step ahead point predictions are calculated, and NMAE (Eq. 3.1) is the metric used to measure the accuracy of these forecasts.

In the remainder of this section, we introduce how the ANN-to-SNN conversion is performed (section 6.2.1) and the model architecture (section 6.2.2) to forecast wind

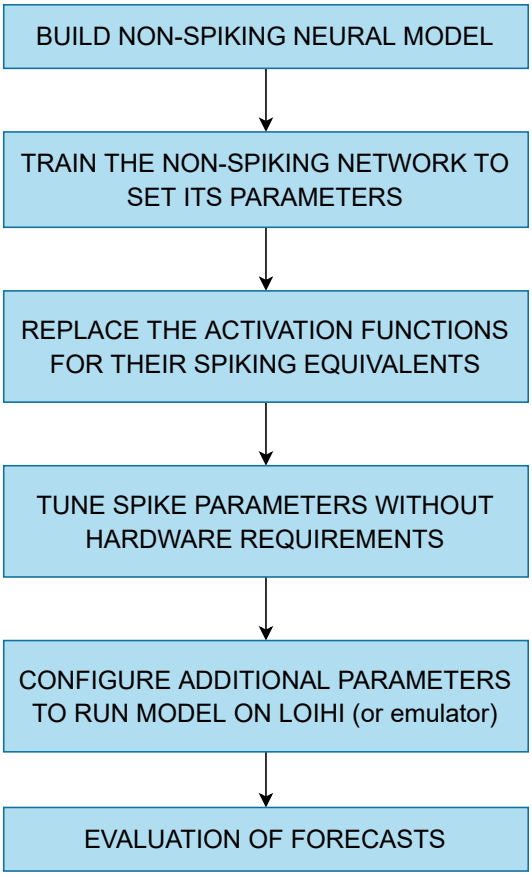

Figure 6.1: Experimental design followed to forecast with a spiking-based model.

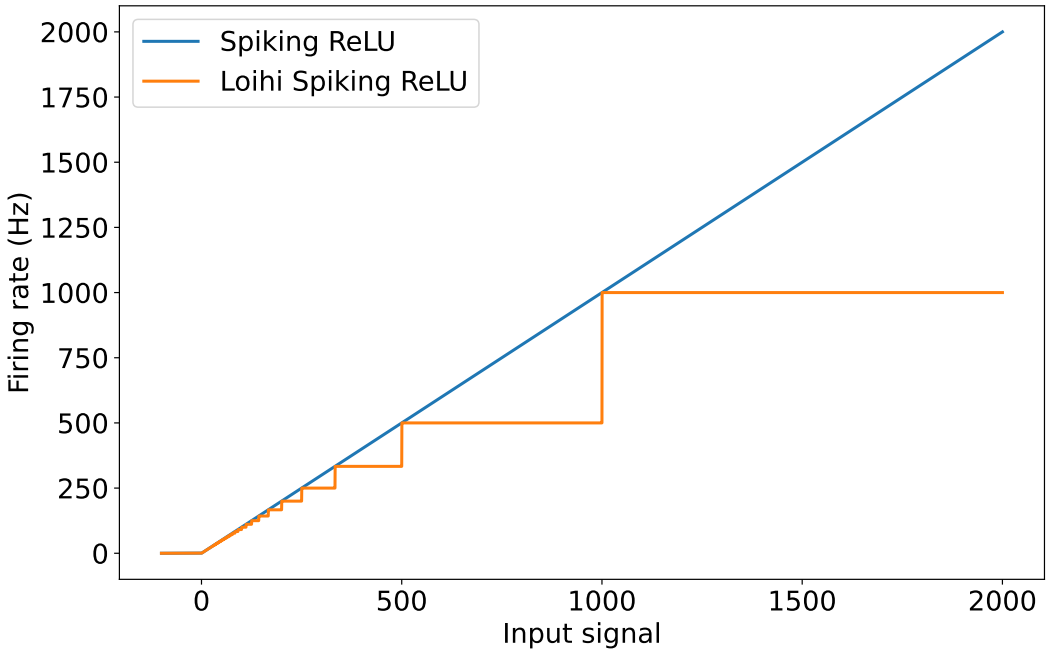

Figure 6.2: Spiking ReLU activation profile (based on DeWolf et al. (2020)).

power. Examples using a synthetic sine wave signal and load data are described in sections 6.2.3-6.2.4 to clarify some details of the steps to follow to successfully convert an ANN model into an spiking one.

6.2.1 ANN-to-SNN conversion

The non-differentiable nature of spikes impedes the use of the backpropagation algorithm to train spiking neurons (Tavanaei et al., 2019). ANN-to-SNN conversion sorts this out by mapping the parameters of a trained ANN to an equivalent SNN. Thus, the main challenge remains as to how the non-spiking model is trained so there is only a small loss of performance in the conversion process. The first point is choosing an adequate spiking activation function. Cao et al. (2015) established an equivalence between the ReLU (Rectified Linear Unit) activation function (Schmidt-Hieber, 2020) and the spiking neuron's firing rate. Hunsberger and Eliasmith (2016) propose a method to train spiking deep networks that can be extended to non-linear activation functions such as leaky integrate-and-fire (LIF), by smoothing the equivalent rate equation employed to train the ANN. To understand this, let us look at the equation governing the dynamics of a LIF neuron:

$$\tau_{RC}\frac{dv(t)}{dt} = -v(t) + I(t) \tag{6.1}$$

where τ_{RC} is the membrane time constant, $v(t)$ is the membrane voltage, and $I(t)$ is the input current. The neuron will fire a spike if it reaches a certain threshold $\mathcal{V}$, and after the potential is reset during a certain period of time (known as refractory period τ_{ref}). The dynamics of the neuron are recovered after the refractory period τ_{ref} is ended. If a constant input current is given to the neuron, the steady-state firing rate (i.e., the time that it takes to the neuron to reach the threshold to fire a spike) can be determined as:

$$r(j) = \left[\tau_{ref} + \tau_{RC}\log(1 + \frac{\mathcal{V}}{\rho(j - \mathcal{V})})\right]^{-1} \tag{6.2}$$

where $\rho(x) = max(x, 0)$. However, this function is not completely differentiable, so the LIF rate equation is softened to address this problem and allow to use the

backpropagation algorithm (Hunsberger and Eliasmith, 2015). The hard maximum ρ is replaced by a soft maximum ρ' defined as:

$$\rho'(x) = \gamma \log(1 + e^{x/\gamma}) \tag{6.3}$$

After training the conventional ANN, the parameters of the SNN are identical to its non-spiking counterpart, only changing the neurons themselves. The performance of the spiking network can be further enhanced by tuning additional parameters. For instance, if using a linear activation function for the spiking forecasting model, the spiking firing rate can be easily increased after training by applying a scale to the input weights of the neurons to make them spike at a faster rate. The output of the network is divided for the same scale to not affect the behavior of the trained network. This way of proceeding is not optimal for non-linear activation functions. Instead, the firing rates can be optimized during training with regularization, so the firing rates are encouraged to spike at a certain firing rate (DeWolf et al., 2020). Furthermore, a synaptic filter can be applied to reduce any possible noise found in the output of the spiking network.

6.2.2 Spiking model architecture

The model architecture (Figure 6.3) is slightly different compared to the ones described in the previous chapters, as it has to be adapted to the requirements of the Loihi hardware. The first distinctive feature of this network is the *off-chip* layer. This layer is a prerequisite to transmit any information with the hardware, as it only communicates with spikes. Thus, this initial layer is run off-chip and converts the input into spikes (Applied Brain Research, 2021). The rest of the network is run on the hardware. A convolutional (*conv-layer*) and a regular fully connected layer (*dense-layer*) are used to process the data and generate the forecast. It is important to notice that not all type of neural networks are currently available in this ANN-to-SNN conversion framework (e.g. LSTM neurons are not supported). The activation function of all these three layers will be a spiking ReLU activation for inference, and the rate-based version is used to train the model.

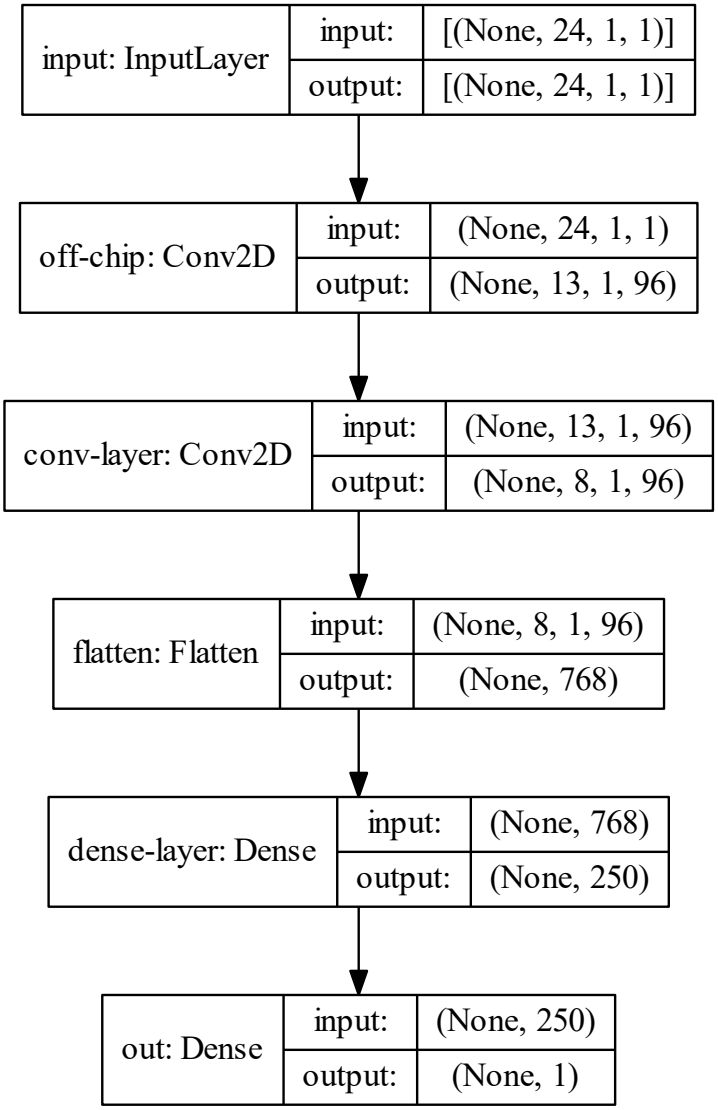

Figure 6.3: SNN model architecture.

To keep the consistency with the previous chapters, this model architecture is applied to decomposed data, which have been obtained with a VMD decomposition with parameters $[\alpha', k] = [500, 8]$, the same parameters selected in the previous chapter (section 5.3.2.2). This will give us the opportunity to examine and adapt the SNN under data with different levels of complexity.

6.2.3 Synthetic signal forecasting

Before applying the methodology to wind power data, let us present an example with a more simple signal (a synthetic sine wave) to clarify and further explain the details of tuning the parameters to reach a good performance with the spiking network. For simplicity, the example using this signal is run within the NengoDL framework, so any additional parameters for implementing the model on Loihi hardware can be dismissed (such as the off-chip layer), so a basic FFNN model is used instead of the model architecture described in section 6.2.2, which suffices to predict accurately such a basic signal.

During the initial evaluation of the spiking network model (Steps 3 and 4), taking into account the discretization of the activation function required for Loihi hardware is of importance to posteriorly transfer our model without a significant drop of performance. Therefore, we must be particularly careful when scaling the firing rate of the spikes, as very high rates will not work on Loihi hardware. Let us examine the implications of disregarding this point with the example shown in Figure 6.4: we build the FFNN model (Step 1) and train it with a rate-based (i.e., non-spiking) ReLU activation (Step 2). Then, we replace the activation for its spiking counterpart, scaling the firing rate with a high enough value (Step 3). The neural activities of three neurons when presenting an input are shown in Figures 6.4a) and 6.4b), having replaced the ReLU activation function by the theoretical spiking ReLU and the discretized version for Loihi respectively. Two of these neurons (shown in green and yellow) fire very fast in the first case, but their behavior is diminished in the second one due to the activation profile, impacting the performance of the model when all the input vectors conforming the testing set are presented to the network (Step 4), as displayed in Figure 6.4c). Thus, the firing rate of this network should be lowered to satisfy the hardware specifications which are required in the following steps to implement the model on neuromorphic devices.

The tuning of the firing rate scale, as well as the amplitude of the spikes, are indeed essential to achieve a good forecasting accuracy, while simultaneously trying to find a balance between the firing rates (enough spikes must be generated to transmit the information to the network) and the sparsity of spiking networks (leveraging the

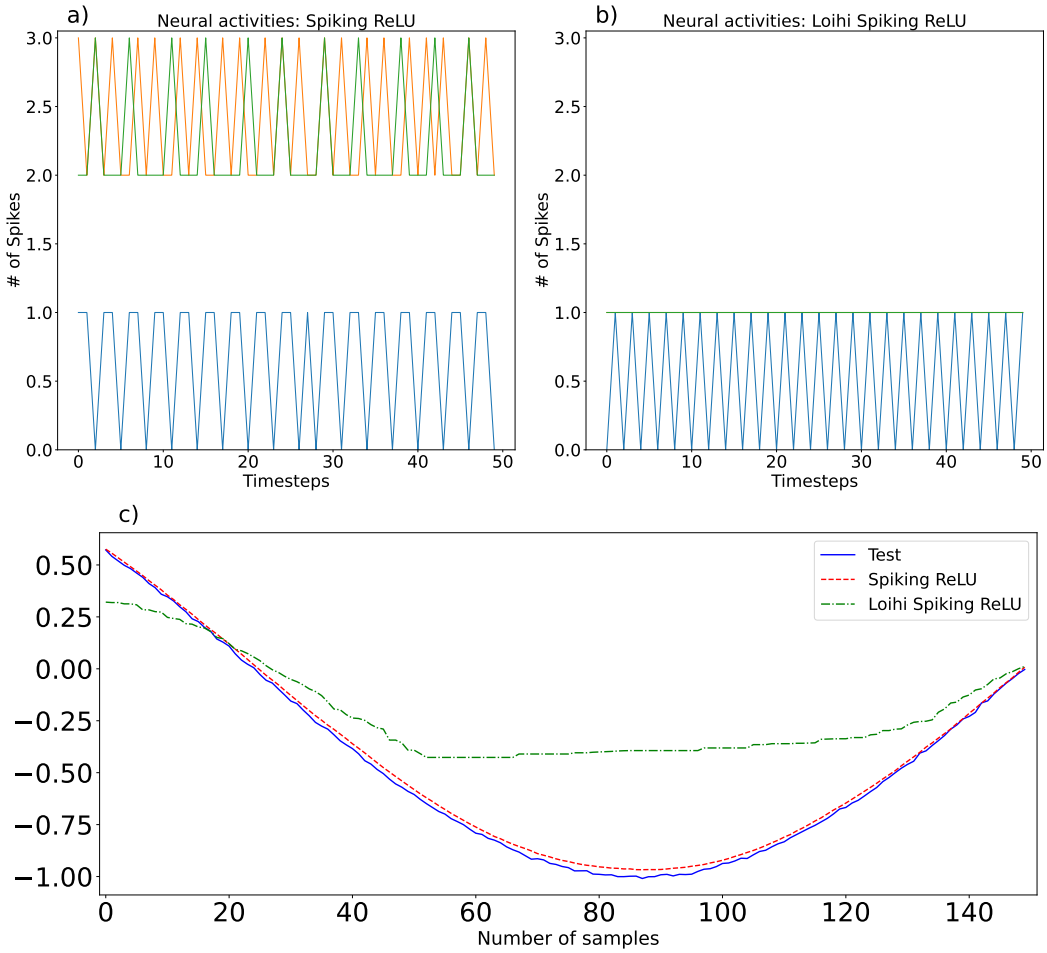

Figure 6.4: a) Neural activities using a spiking ReLU activation for inference (one input vector is shown to the network during 50 timesteps), b) neural activities using the discretized version of the spiking ReLU activation, and c) predictions over the testing set.

promise of low-energy consumption of neuromorphic devices). Following the same example, let us a fix a certain spiking amplitude and experiment with different firing rate scales to find this trade-off, taking into consideration the Loihi-tailored spiking ReLU activation. The neural activities of the same three neurons are shown in Figure 6.5a) for a scale of 1 (i.e., keeping the same input weights as the original SNN), in Figure 6.5b) for a scale of 5 (so a linear scale of 5 is applied to the inputs of the neurons), and in Figure 6.5c) using a scale of 50. As expected, the spikes fire much faster when increasing this parameter, being the spikes almost indistinguishable in the latter case, and thus reducing the sparsity of this network. Between the neural activities a) and b), the mean firing rates are low (6 and 30.9 Hz) and show a more sparse firing rate, meaning that both are in principle better suited parameters for this application. The preliminary results computed within NengoDL (Figure 6.5d))

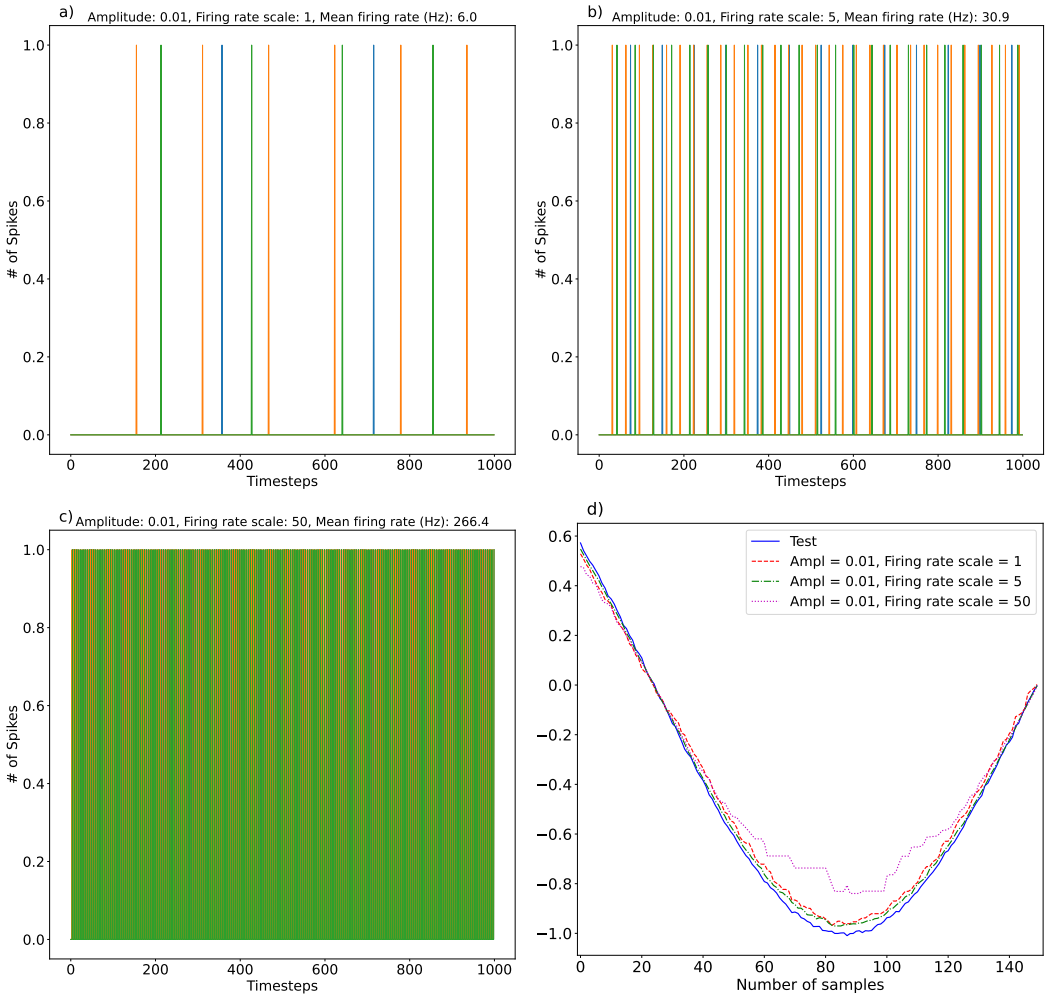

Figure 6.5: a) Neural activities setting an amplitude = 0.01 and a firing rate scale = 1, Neural activities setting an amplitude = 0.01 and a firing rate scale = 5, c) Neural activities setting an amplitude = 0.01 and a firing rate scale = 50, and d) predictions over the testing set.

indicate that a scale of 5 provides a slighlty better performance, being thus the most adequate value for this parameter. Naturally, tuning these parameters is a harder task when dealing with more complex data and more complex spiking architectures, as we will see in the following section.

6.2.4 Load forecasting

Let us set another example using real data to calculate one-step ahead forecasts. In particular, short-term load forecasting is an example of interest due to its close relation with WPF, as both are necessary to operate and maintain the stability of the electrical grid (Hong and Fan, 2016). Furthermore, load demand data show regular daily and weekly patterns which are not observed in wind power data (Quan et al.,

2013), so a model architecture using CNNs (such as the one described in section 6.2.2) is a good candidate to extract such features (Sadaei et al., 2019). Records of aggregated hourly demand data from Ireland (Figure 6.6) can be found on the European Network of Transmission System Operators for Electricity (ENTSO-E) website (ENTSO-E, 2021). The available measurements have been recorded between 2016 and 2018.

As usual, we build and train the rate-based equivalent of the model, and subsequently the activation functions are replaced. Then, the spike parameters are tuned without specifying any hardware requirements, and we monitor the initial results to choose the best values for these parameters. Some of these initial forecasts are shown in Figure 6.7. The existing patterns in load data are captured by the model, and adjusting the spikes parameters is fairly straightforward. The dashed red line (obtained using an amplitude of 0.05 and a firing rate scale of 50) matches closer the test data compared to the rest, so these values are chosen for its implementation on Loihi's emulator (or the hardware itself if available).

As indicated in Step 5, the network must be further adjusted to be run on Loihi. In our particular case, we must indicate what layers are run on- and off-chip, but other adjustments might be needed for more complex networks, such as distributing the connections of the network over multiples cores on Loihi (Applied Brain Research, 2021). Figure 6.8a) shows that neurons are effectively firing in each layer, whereas Figure 6.8b) compares the initial forecasts obtained previously while tuning the spike parameters (the red dashed line) and the load forecasts emulating the Loihi chip (dash-dot green line). It can be observed that the model architecture translates well to the emulator after fine-tuning those hardware specifications, resulting in similar load forecasts with respect to the initial evaluation of Step 4.

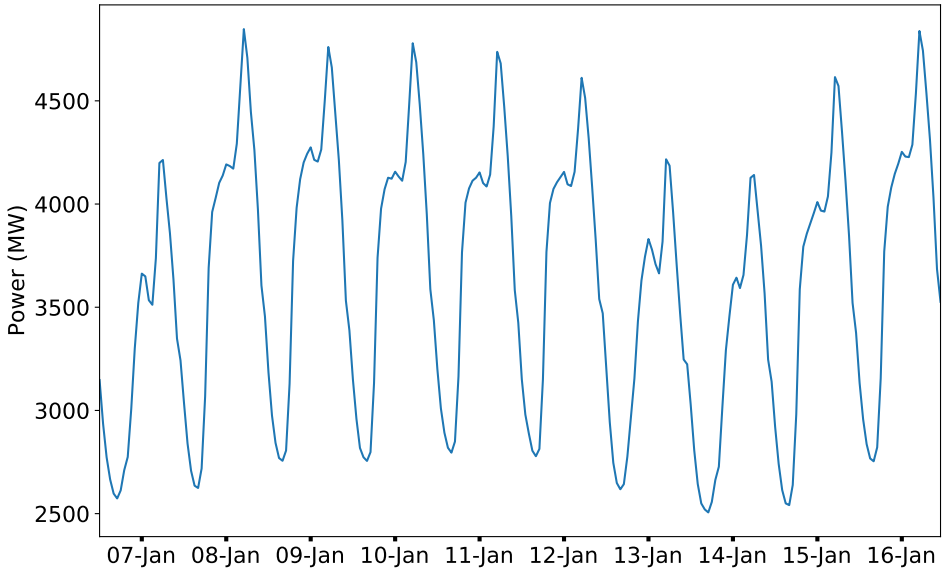

Figure 6.6: Sample of hourly load demand data for Ireland.

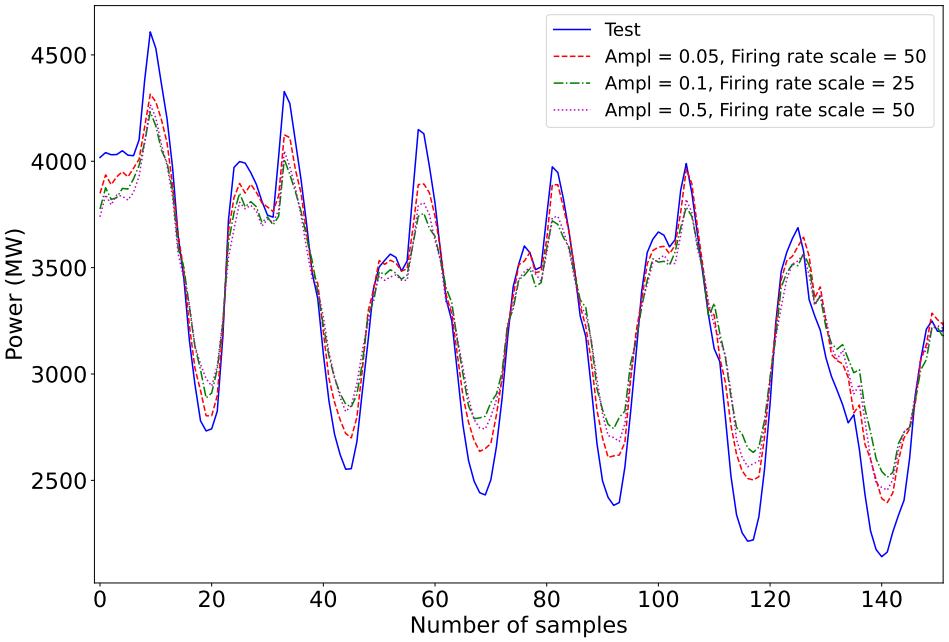

Figure 6.7: Preliminary one-step ahead load forecasts setting different spike amplitudes and firing rates.

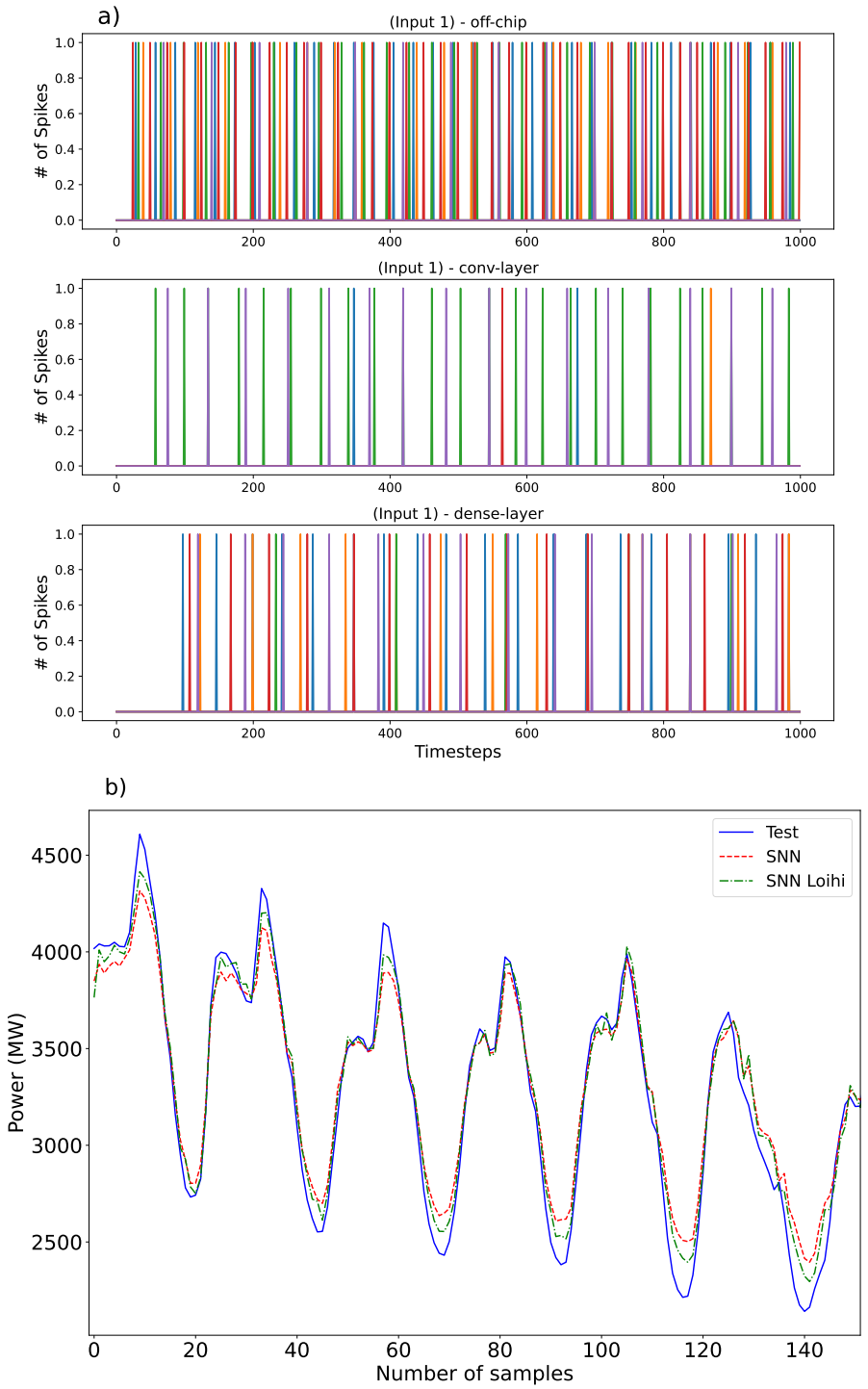

Figure 6.8: Results for one-step ahead load forecasts: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

6.3 Wind power forecasting

First and foremost, we have to keep in mind that neuromorphic computing is still in its infancy, so the goal is to reach an acceptable level of performance to build up our knowledge for implementing spiking-based models for WPF, and not outperforming the current well-established neural network models (Davies et al., 2021). In addition, spiking models could benefit further from data collected with event-based sensors (Leñero-Bardallo et al., 2018), and could contribute to unleash the potential of SNNs for WPF if some type of relevant data could be compiled this way. Bearing these insights in mind, let us apply the model architecture described in section 6.2.2 to build the spiking forecasting models for each mode extracted from Irish wind power data.

Following the methodology shown previously in section 6.2, we build the model presented in section 6.2.2 (Step 1), and we train the rate-based neural network model to set its network parameters (Step 2). Then, we transform it into an SNN by switching the activation functions to spiking ones (Step 3). In the Step 4, we set empirically values for the amplitude and firing rate of the spikes (Table 6.1) within the NengoDL framework till we obtain a reasonable performance of the spiking model. On the one hand, the spiking amplitude modulates the amount of information transmitted to the subsequent layers of the network, whereas the firing rate adjusts how fast the spikes are being fired. If the firing rate is high, the behavior will be closer to the non-spiking model and thus the performance will increase, but at the cost of losing the characteristic temporal sparsity provided by the spikes (Patel et al., 2021). In addition, having a high firing rate will lead to detrimental results on Loihi because of

Table 6.1: Main spiking network parameters.

	Neuron type	Spiking amplitude	Firing rate scale
Mode 1	Spiking ReLU	0.1	25
Mode 2	Spiking ReLU	0.05	40
Mode 3	Spiking ReLU	0.01	70
Mode 4	Spiking ReLU	0.1	90
Mode 5	Spiking ReLU	0.3	200
Mode 6	Spiking ReLU	0.3	200
Mode 7	Spiking ReLU	0.5	400
Mode 8	Spiking ReLU	1.5	500

the discrepancy resulting from discretizing the spiking activation function (as shown in Figure 6.2). The low mean firing rates of these preliminary results (Table 6.2) suggest that the selected parameters are potentially good to be implemented on Loihi. Afterwards, we configure some additional parameters to run the model on the Loihi emulator (Step 5). In particular, we must indicate what part of the model is run off-chip (in this case, the *off-chip* layer we use to communicate with the chip) and how long each input vector is presented to the network (in our case, we will show each one for 0.4 seconds).

The information recorded in Steps 4 and 5 are shown in Figures 6.9-6.16 for modes 1 to 8 respectively. Part a) of these figures shows the neural activities of each layer (limited to 5 neurons for illustrative purposes). These neural activities correspond to the first input vector fed to the model, and produce the first point forecast shown on part b). Furthermore, the neural activities help us visualize the mean firing rates shown in Table 6.2: modes 1, 3, and 8 exhibit higher firing rates, which translates into a large number of spikes generated during this timeframe, whereas the rest of modes present a more sparse behavior, resulting in a lower generation of spikes. In some cases, such as mode 4 (Figure 6.12) and mode 5 (Figure 6.13), the neurons of the *off-chip* layer need a large time to settle and thus start to spike, delaying the neural response of subsequent layers. Although the temporal sparsity is not necessarily adverse in terms of forecasting accuracy (it is indeed one feature we are looking for when building a SNN), it might occasionally be advisable to fine tune the firing rate of the *off-chip* layer to propagate the information faster to the rest of the network.

Table 6.2: Mean firing rates (Hz) for each layer.

	Off-chip layer	Conv layer	Dense layer
Mode 1	8.1	8.3	12.0
Mode 2	1.9	1.8	2.1
Mode 3	7.2	4.9	2.5
Mode 4	1.6	1.2	1.0
Mode 5	1.2	1.3	1.0
Mode 6	1.1	1.0	1.0
Mode 7	1.4	1.1	1.0
Mode 8	3.7	9.6	11.5

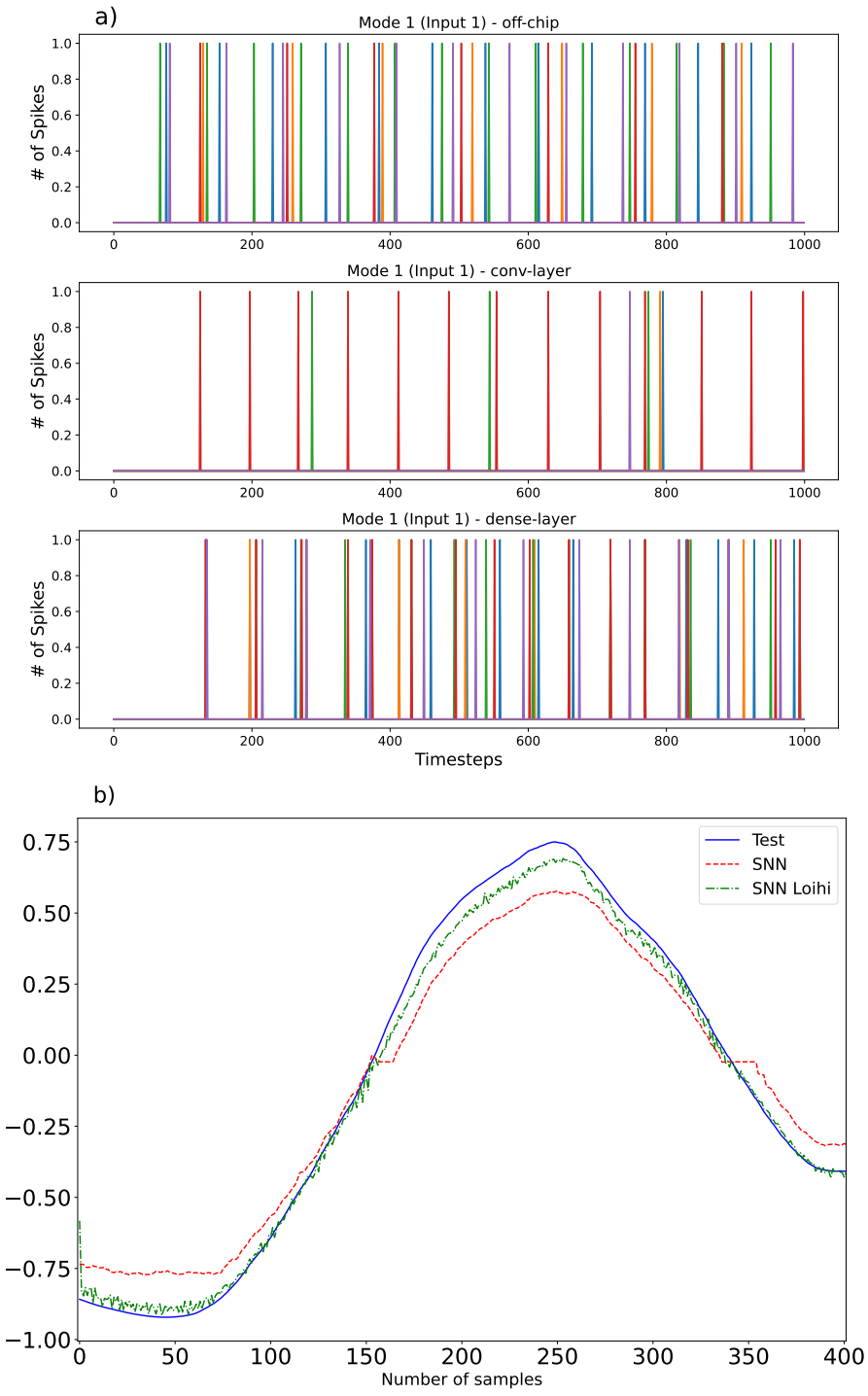

Figure 6.9: Results for mode 1: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

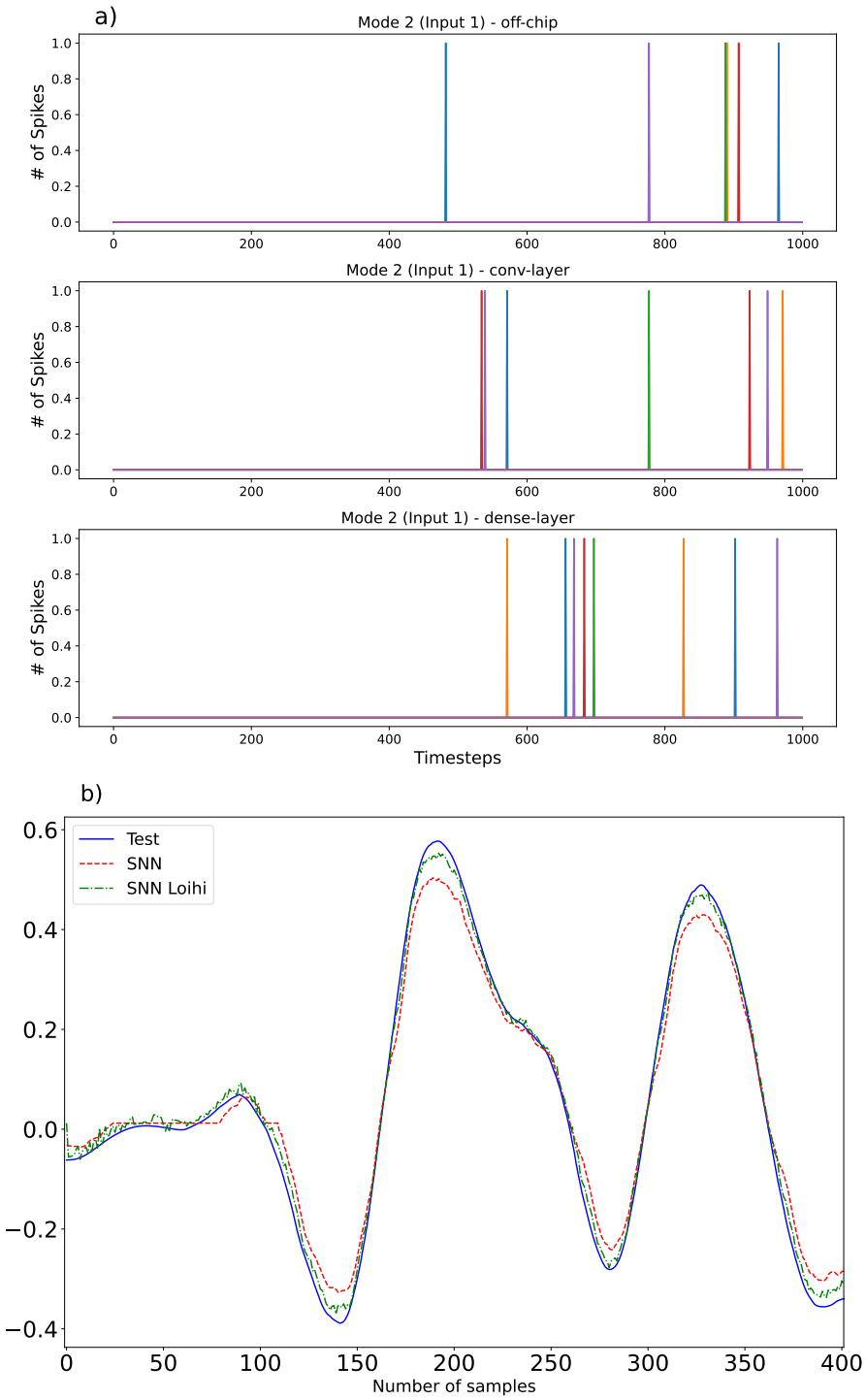

Figure 6.10: Results for mode 2: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

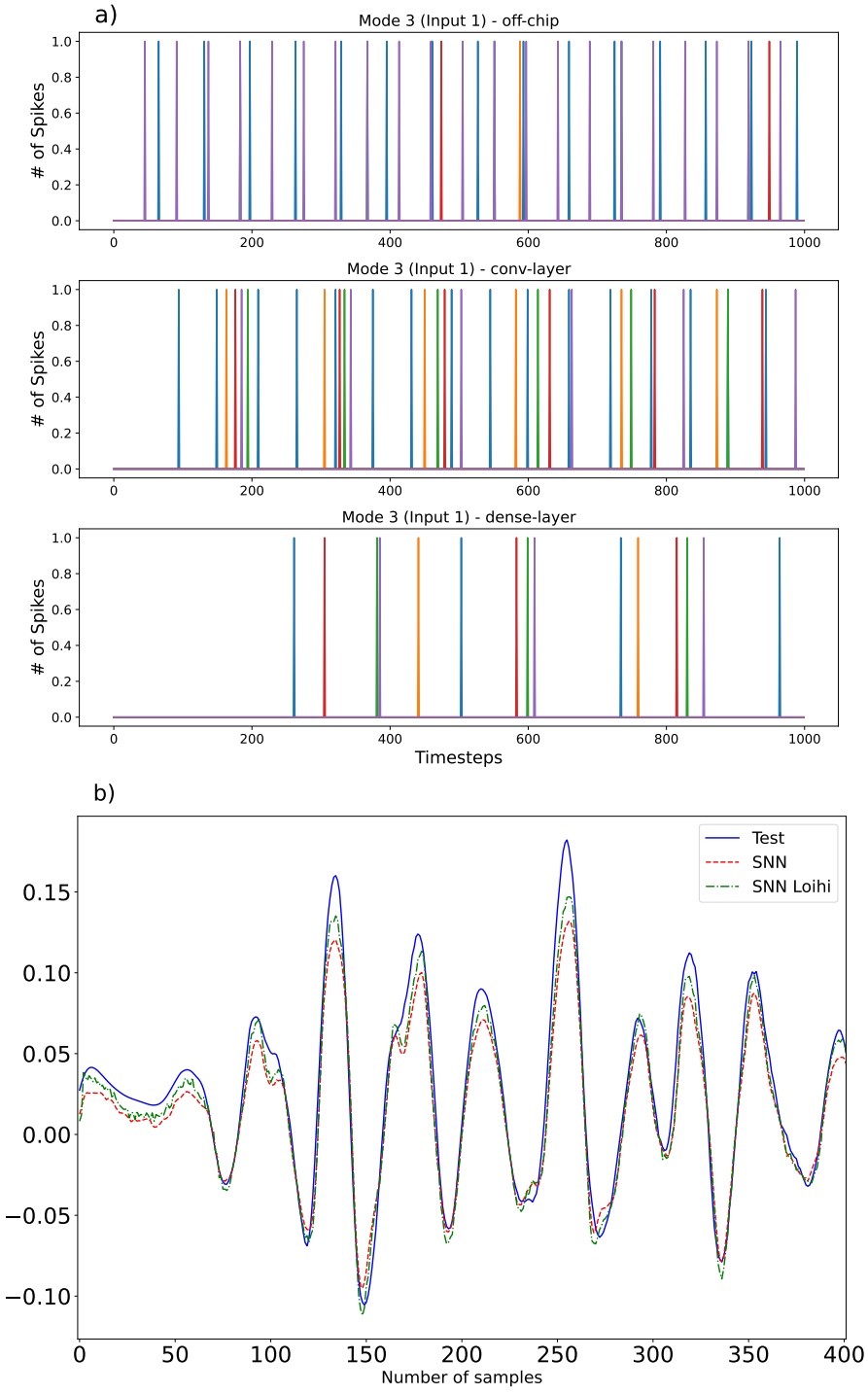

Figure 6.11: Results for mode 3: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

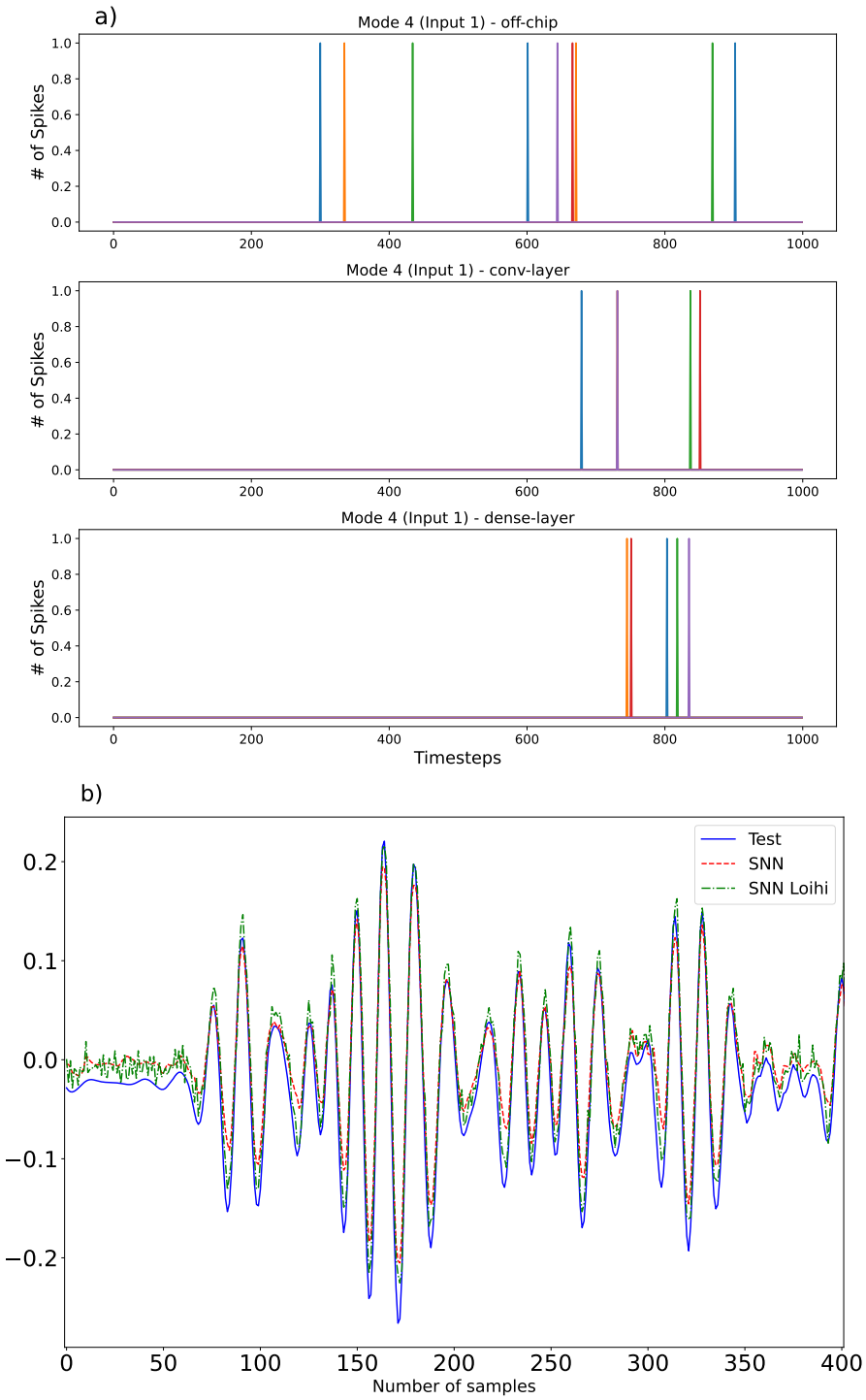

Figure 6.12: Results for mode 4: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

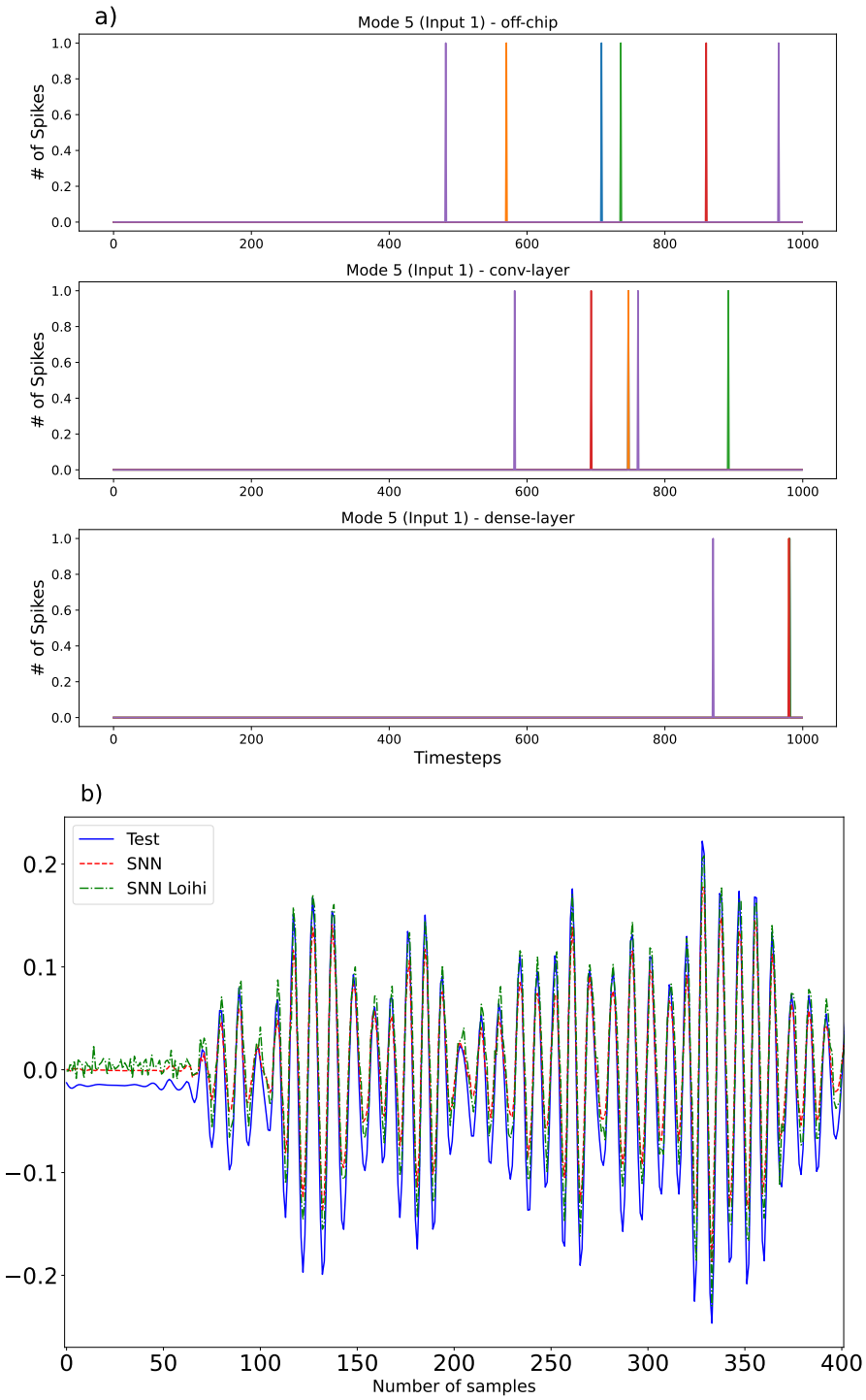

Figure 6.13: Results for mode 5: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

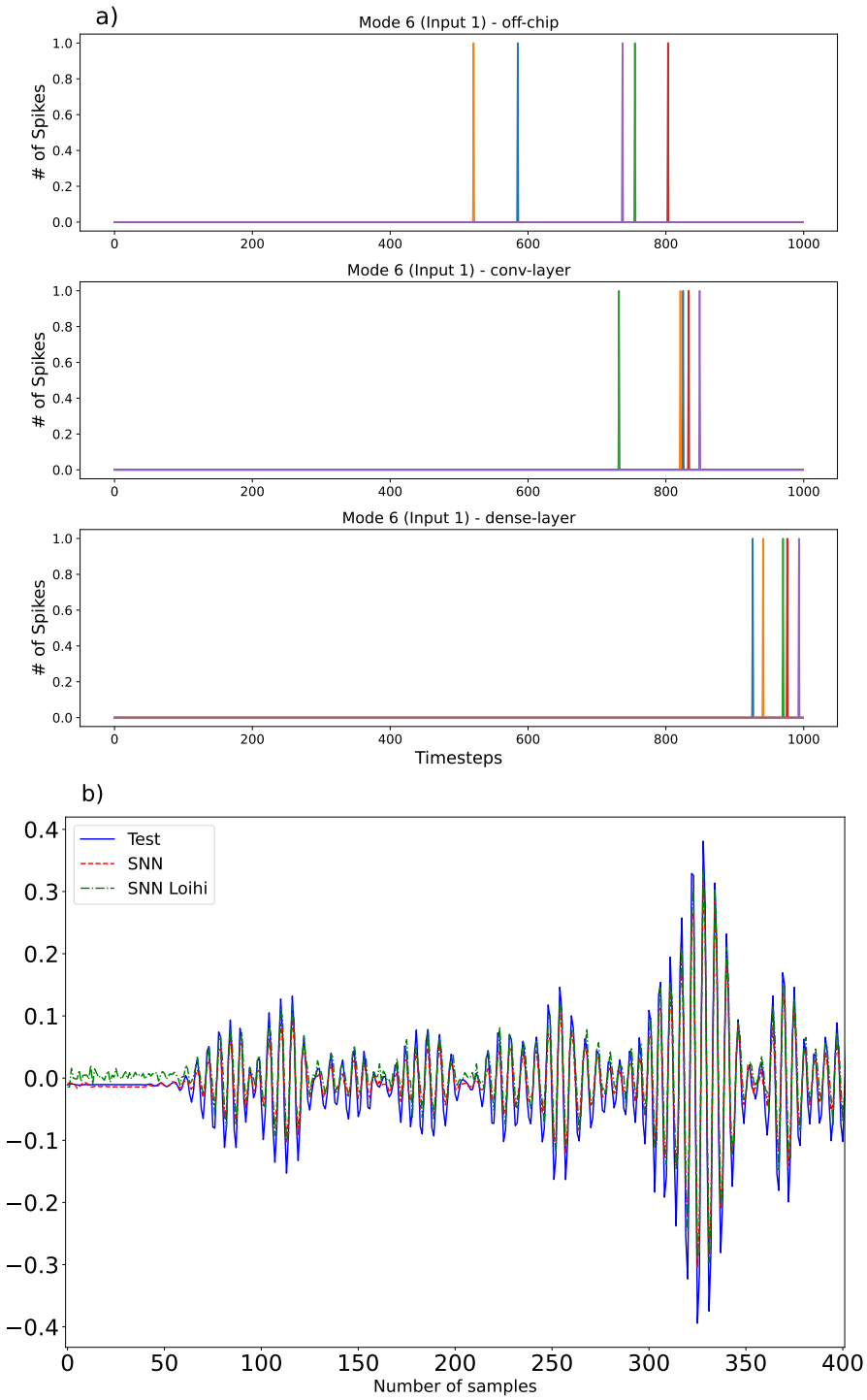

Figure 6.14: Results for mode 6: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

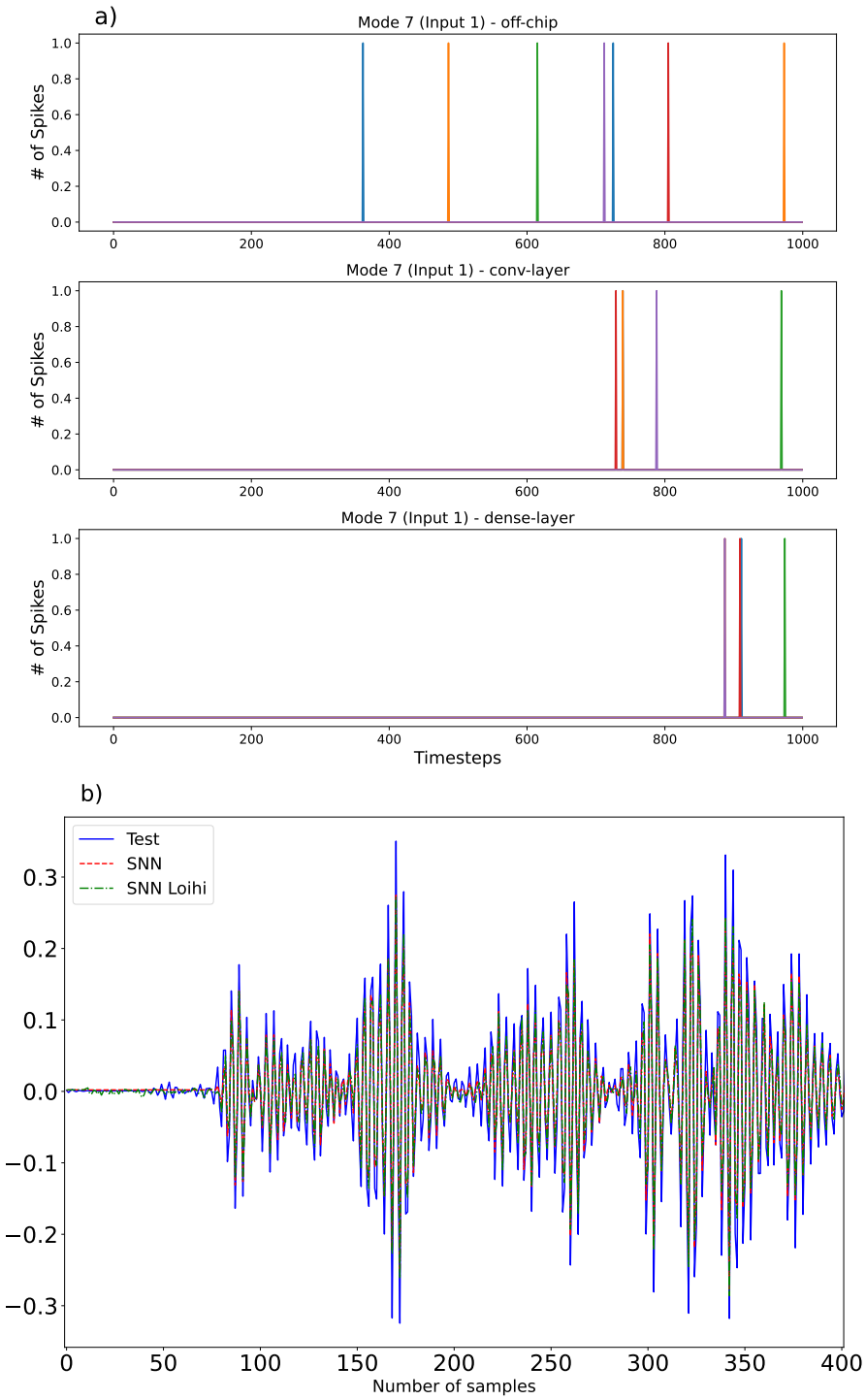

Figure 6.15: Results for mode 7: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

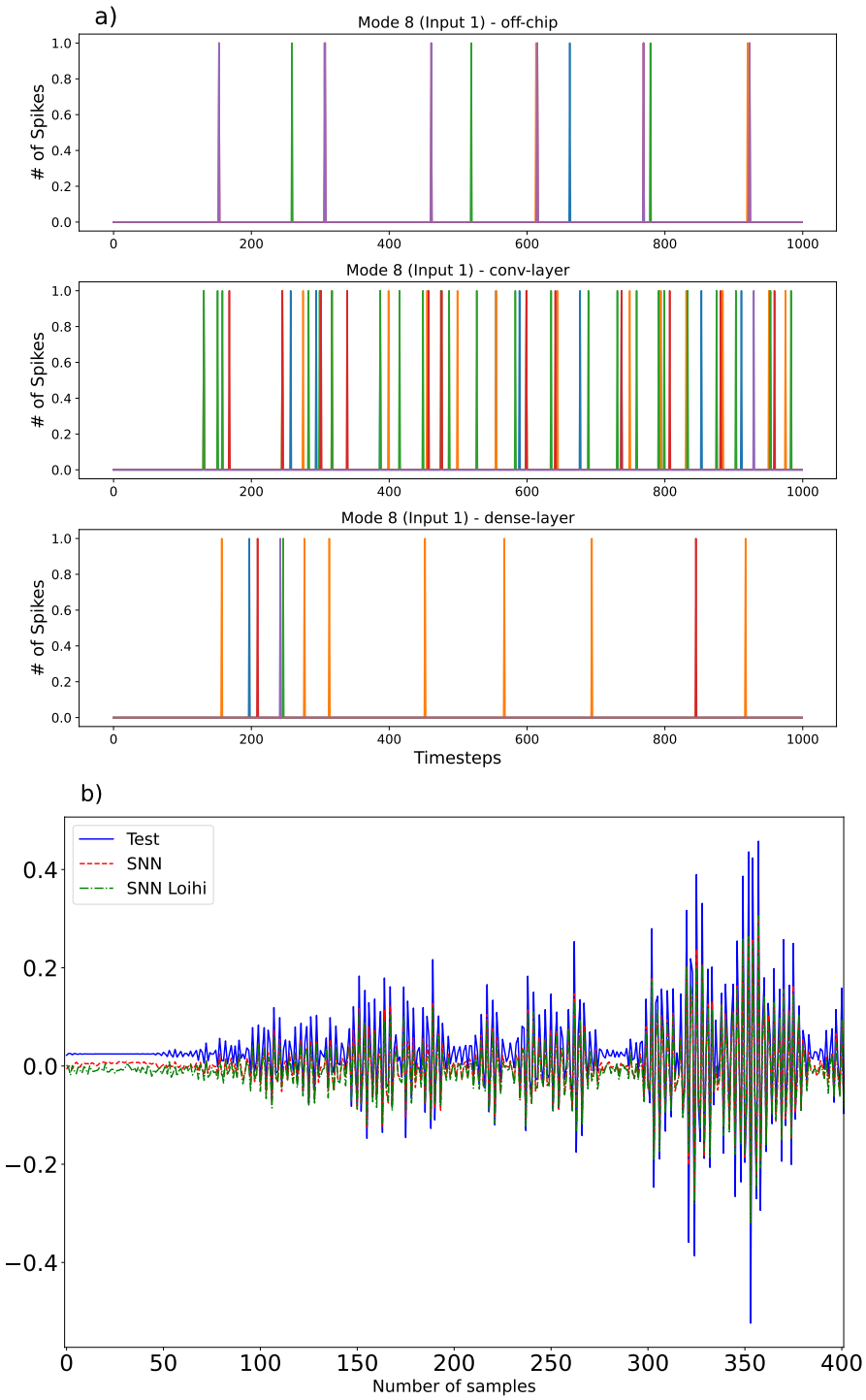

Figure 6.16: Results for mode 8: a) Neural activities of 5 neurons of each layer. One input vector is shown over 1000 timesteps. b) Predictions over the testing set with the SNN architecture (dashed red line) and running the SNN on the Loihi emulator (dash-dot green line).

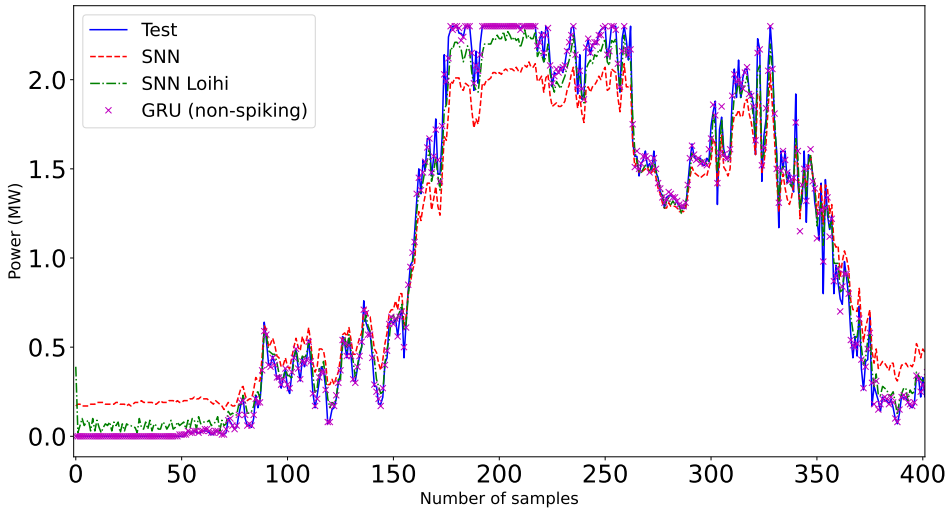

Figure 6.17: One-step ahead WPFs with the SNN architecture (dashed red line), running the SNN on the Loihi emulator (dash-dot green line) and a non-spiking VMD-GRU model (purple crosses) over the testing set.

At this stage, the performance of our models can finally be examined (Step 6). The model is designed to provide 1-step ahead point forecasts (Figure 6.17). The dashed-red lines show the forecasts obtained while tuning the model using the NengoDL framework in Step 4. While this preliminary model is able to forecast increasing/decreasing trends of power generation, it is not as accurate for high or low power generation scenarios. Nonetheless, this initial assessment allows us to prepare our model for Loihi (dash-dot green line), which demonstrates the same skill to detect increasing/decreasing trends of power generation as the preliminary model, while showing a better capability of forecasting high/low power generation values. This difference of performance also arises from the model architecture itself. When the model is initially evaluated outside the Loihi framework, it cannot discern that the first layer is only set to start to generate spikes. Such nuance is captured when the model is configured to be implemented on Loihi. Additionally, we observe that the forecasts are not as accurate compared to the VMD-GRU model presented in chapter 4, and the outputs are generally noisier. However, this is an expected outcome due to the current limitations of neuromorphic hardware.

In conclusion, we have successfully transformed a non-spiking neural model into a spiking one with a reasonably good performance, having achieved a 2.84% NMAE for 1-step ahead forecasts with the model adapted to neuromorphic hardware.

6.4 Conclusions

Neuromorphic computing provides a new paradigm to build energy-efficient low-latency algorithms in contrast to the current state-of-the-art ML/DL strategies. In particular, SNNs aim to learn in a more biologically plausible manner (Tan et al., 2020) by mimicking more closely the spike-based transmission of information which occurs in the brain (Kasabov, 2019). Currently, the two major challenges for the use and implementation of SNNs are 1) the training of such models, as the well-established training strategies based on the backpropagation algorithm applied to ML/DL cannot be used directly as spikes are not differentiable, and 2) the implementation of SNNs on neuromorphic hardware, as SNNs must be tailored to cater to the specific requirements of the hardware. The first challenge has been addressed so far with different approaches, such as ANN-to-SNN conversion, and using variations of error backpropagation to directly train SNNs. The second challenge is hardware-dependent, and should be addressed according to the requisites of the hardware used to implement the SNN.

In this chapter, we adopt an ANN-to-SNN conversion approach to forecast wind power, and obtain these WPFs emulating or running the spiking model using the neuromorphic hardware Loihi (Davies et al., 2018). SNNs are designed using the framework provided by the software Nengo (Bekolay et al., 2014; Rasmussen, 2019). First, we build and train the non-spiking neural network. After training, we map the parameters and replace the activation functions for their spiking counterparts, which will be used during the prediction stage. Then, without considering hardware specific constraints, some preliminary results are evaluated to tune some spike-related parameters such as the firing rate or the amplitude of the spikes. Finally, the SNN is further adjusted to be run on the hardware emulator (or actually running the model on Loihi if available) to obtain the WPFs. Following all these steps, we have managed to reach our goal of achieving a good level of performance with the proposed spiking architecture, obtaining a NMAE of 2.84% for one-step ahead forecasts when the model is emulated on Loihi.

As the current training schemes are still in their infancy, and the availability of neuromorphic devices is still limited, there are plenty of directions to follow to con-

tinue the development of neuromorphic-based WPF algorithms. Regarding the first point, the described ANN-to-SNN conversion approach can be further tuned, by for instance adjusting the firing rates of each layer individually, or analyzing the effect of synaptic filters to smooth the output. In addition to that, extending the modelling of spiking networks to train them directly would lead to a more efficient use of neuromorphic hardware, as well as using online approximations of the backpropagation algorithm to learn online from the available data (Davies et al., 2021), leveraging the ability of SNNs to implement real-time algorithms. Furthermore, neuromorphic hardware keeps getting developed, such as Loihi 2 (Orchard et al., 2021), which will contribute to strengthen the implementation of more complex neuromorphic algorithms.

Chapter 7

Conclusions

The main research objective of this thesis was to sustainably improve short-term wind power forecasts without introducing new modelling algorithms but identifying ways of improving the existing ones. This has been achieved by better understanding and quantifying forecast errors, developing and proposing guidelines to benchmark novel statistical wind power forecasting models, and leveraging the current data and cutting-edge technology to provide such forecasts in the most efficient manner possible.

This chapter serves as an epilogue, summarizing the main contributions of this thesis, assessing critically the research performed as well as indicating possible directions for future research.

7.1 Main contributions

The central contributions of this thesis can be summarized into three categories: 1) decomposition and quantification of sources of forecast error, 2) standardization of model benchmarking for statistical wind power forecasting models, and 3) efficient processing of these forecasting models by exploiting the potential of data and technology.

Chapter 2 presents a simulation-based statistical framework to identify the main sources of error, by decomposing the forecast error and quantifying the contribution of each source to the total forecast error. Even if this type of insight is well known in the context of statistical error modelling, such a decomposition has barely been

addressed previously despite the ample literature on time series model misspecification. Thus, two novel indices are defined to quantify the inflation of the MSE with respect to a given true model, which can be an ARIMA or SARIMA model of any order. The proposed framework has been applied to wind speed forecasting (which can be later converted into forecasts of wind power using a predefined power curve, or used as an exogenous variable of a more advanced WPF model), where wind speed data have been generated with ARIMA$(3, 1, 1)$ processes (assuming this is the true model) to quantify the indices which measure the influence of misspecifying the model and its parameters. In relation to our research objectives, this quantification will allow practitioners to better select models while avoiding overfitting by finding the adequate order of AR and MA terms.

Chapter 3 presents an overview of the most common metrics applied for evaluating statistical WPF models in the recent literature for both point and probabilistic estimates. Regarding the latter, we have paid particular attention to the evaluation of prediction intervals, the most popular representation of probabilistic estimates. Furthermore, this chapter illustrates the capability of these metrics to properly evaluate the performance of WPF models over different datasets, time resolution and other model specific attributes. These aspects are often disregarded to determine the validity of a forecasting model over an out-of-sample set, as the values of these metrics could be influenced by the intrinsic characteristics of the dataset and they could fluctuate considerably for different periods of the same testing set. A numerical study is presented using wind power data from Ireland with two different time resolutions (10 minutes and 1 hour) and decomposition-based hybrid models to be assessed by the performance evaluation metrics. Furthermore, we have stressed the importance of leveraging these performance evaluation metrics to additionally assess the robustness of WPF models against different power production scenarios in order to have a more comprehensive vision on model performance. Therefore, the research carried out in this chapter proves to be useful in the model development stage of statistical WPF models.

In fact, the insights drawn from chapters 2 and 3, together with the main guidelines outlined by the IEA Wind Task 36 (Giebel et al., 2020), have been the foun-

dations of the work conducted in chapter 4, where a benchmarking framework to evaluate short-term WPFs of statistical models has been developed. The lack of this type of benchmarks is one of the main research gaps found in the literature review, reducing the real capability of new wind power forecasting methodologies implemented for industry applications (Zack et al., 2019). Therefore, this benchmarking framework is designed to evaluate statistical WPF models over varied wind power generation scenarios with standardized performance evaluation metrics (NMAE for point estimates, and PICP and PINAW for prediction intervals). An illustrative example of this benchmarking framework is applied to a set of 21 decomposition-based hybrid models (Qian et al., 2019), which have been implemented considering three state-of-the-art decomposition algorithms (VMD, EMD and EEMD) and a set of neural network based forecasting models. Any possible bias to initialization in these neural network based models is also addressed in the proposed framework. Furthermore, as the benchmarking process considers diverse realistic operational conditions of the wind farms, not only the accuracy but the robustness and reliability of the WPF models are examined in this framework. All things considered, the implementation of such a benchmark in the research community would be helpful to overcome the current state of WPF modelling, where hundreds of publications claim to provide the best model performance with respect to other state-of-the-art models, although under different evaluation criteria. The insights drawn from this benchmark are valuable from an industrial point of view as well, as this benchmark would allow practitioners to choose objectively the best existing model for any particular application, which depends directly on the prediction horizon (Table 4.1). For instance, wind farm operators use the information provided by WPFs to bid in the electricity market. The higher the deviation of the forecast is with respect to the actual value, the higher the penalisation will be in economical terms. Furthermore, having a more representative and realistic vision of WPFs in terms of model performance will contribute to unleash the potential of WPFs as a tool to reach the energy transition scenarios projected by the IEA (International Energy Agency, 2021) in which renewable energy sources will have a higher share in the electricity mix. This point is particularly important for Electricity Transmission System Operators (e.g. EirGrid

in Ireland), as improved WPFs will facilitate the schedule of electricity generated from wind farms. Otherwise, inaccurate WPFs will result in additional costs as the wind farms will be run either with reduced efficiency if the WPF estimate is lower than the actual generation or demand must be met with another sources of electricity (such as gas or coal plants) if the WPF estimate is higher than expected. This line of thought can be extended to other renewable energy sources characterized by their intermittency such as the case of solar energy (Yang et al., 2021b). In any case, this framework does not only allow to select a certain forecasting model in terms of improved WPFs, but it extends such selection to probabilistic forecasts represented as prediction intervals, allowing to use more advanced decision-making strategies for industrial applications (Pinson et al., 2007a; Wang et al., 2016c) and consequently increasing the competitiveness of both wind farm operators and the Transmission System Operator.

Another thought we have kept in mind throughout this thesis is that the ultimate goal of reducing the O&M cost of onshore wind energy is to make it more accessible to transition to a greener future (de Simón-Martín et al., 2019). Therefore, it makes sense that our wind power forecasting methodologies are also *green*, in the sense that the computational cost should not be too high or should at least be optimized as much as possible, in line with the computational efficiency pursued in this thesis as one of the research objectives. Thus, the last main contribution is based on this idea. The work presented in Chapter 5 aims to leverage high-resolution data collected at a turbine-level in order to provide improved WPFs while not increasing excessively the computational cost required. This is achieved by grouping turbines using the DBSCAN clustering algorithm and building forecasting models for each identified cluster. A case study has been performed in this chapter using the turbine-level data extracted from the farm labelled as WF-II. Two clusters have been identified in this wind farm, and have resulted in improvements in terms of NMAE scores around 5-10% with respect to an individual farm-level forecasting model for very short-term forecasts.

On the other hand, another take on this *green* concept is developed in Chapter 6, where a perspective from a technological point of view is provided. There is a trend

to apply ML/DL models to forecast wind power, as clearly observed in the literature review (Table 1.3). Even if these methodologies are able to estimate WPFs with high accuracy, they can become very expensive computationally during the training stage. Therefore, this chapter introduces the idea of using neuromorphic devices to build energy-efficient algorithms to reduce the computational cost associated with the current status of machine and deep learning models. In particular, an ANN-to-SNN conversion approach has been applied. As a starting point, the architecture of a conventional neural network model is carefully designed so it can be successfully converted into an equivalent spiking network which is used during the prediction stage. The architecture is also tailored for the deployment of the SNN on neuromorphic hardware, specifically Loihi, a neuromorphic chip developed by Intel (Davies et al., 2021). Emulating this device, we have achieved a NMAE score of 2.84% for one-step ahead WPFs.

In a nutshell, two main directions have been followed to develop this last contribution. From the data point of view, clustering wind turbines reduces indeed the computational cost while improving forecasting accuracy compared to a single farm-level model for short-term horizons. On the other side, the new computational paradigm provided by neuromorphic computing open up a lot of possibilities, particularly in the implementation of energy-efficient wind power forecasting models which can be leveraged for online forecasting and their implementation in the Internet of Things space.

7.2 Limitations

In this section, we indicate the existing limitations to provide the readers with a complete vision of the research carried out.

The statistical framework to decompose forecast errors presented in chapter 2 can only be applied to ARIMA or SARIMA type of models at the current stage, and the data must be generated synthetically from this type of processes to estimate such decomposition. Furthermore, the comparison with other misspecified models is restricted to the same order of integration. That is, the framework allows us to compare a given ARIMA(p_1, d_1, q_1) model with another ARIMA(p_2, d_2, q_2) model

if and only if $d_1 = d_2$.

A further standardization of the evaluation of WPF models is limited by the availability and the representativeness of the data. In this thesis, data are representative in terms of sample size, as two years and a half of data allow to draw meaningful conclusions (as indicated by Yan et al. (2022), a one-year long dataset is large enough as all seasons are covered) and the proposed framework does reduce the impact of data to a minimum extent (by dividing the data into different training/testing subsets representing varied operational conditions). However, the operational conditions might not be necessarily representative for other wind regions. This limitation could be addressed by analyzing the representativeness of the operational conditions of the Irish wind farm data for other regions.

The clustering-based approach described in chapter 5 is limited in terms of scalability, so it should be tailored for every specific case to be studied. Furthermore, the same type of data may not necessarily be available for another wind farms, so the clustering variable selection has to be performed for other wind farms to find the best configuration. On the other hand, the deployment of SNN algorithms for neuromorphic devices is tied to the development of such devices, which is unfortunately not very mature yet, so it will take a few years to consider the benefits of neuromorphic devices for more than a proof of concept.

7.3 Directions for future research

There are several lines of potential research. The statistical framework introduced in chapter 2 has plenty of potential for improvement: it could be extended to consider not only time series analysis models but other widespread methodologies such as artificial neural networks. For that, the definitions and concepts defined within the theoretical basis should be adjusted to fit other type of statistical methodologies. Furthermore, introducing real data into the equation could also allow to establish some type of index addressing the error coming from the data themselves. Synthetic data are still valuable using ARIMA type of models, but further research is necessary to carefully choose the coefficients of the AR and MA parameters.

With respect to the findings of chapters 3 and 4, the benchmark can be further

improved if data are to be standardized in some manner, for instance estimating the most optimal sample size and statistics for such standardization (Hahn et al., 2017), and ensuring data are collected following international standards (Alizadeh and Ozansoy, 2016). Furthermore, the results of the benchmark could be extended with open source wind data to facilitate the reproducibility of results (Effenberger and Ludwig, 2022).

It would also be positive to investigate particular aspects of every type of forecasting model and take them into consideration for the concluding comparison provided by the benchmark. For instance, we have identified how sensitive decomposition-based hybrid models are to some initialization parameters, so several configurations of these parameters might result in a distinct better performance depending on the prediction horizon we are aiming for our model. Thus, this kind of considerations should be addressed to find the best possible configuration of any given wind power forecasting model and be presented in the benchmark for the sake of completeness. Furthermore, investigation with respect to the validity of the model over time should be considered. That is, forecasting models should be retrained adding new measurements after some time to maintain a similar level of performance. However, tools to quantify rigorously how and when to take this decision have barely been investigated in the literature.

Other potential lines of research can be pursued using the conclusions drawn from the computational optimization of wind power forecasting models. In terms of leveraging high resolution data, the turbine clustering approach can be extended using other clustering algorithms and adding other variables (such as meteorological data and the operational state of the turbines) to determine such clusters. In addition, this clustering is subject to changes if for instance the wind direction changes, so a mechanism to update the clusters in real time would be beneficial to keep yielding improved wind power forecasts. On the other hand, the application of neuromorphic computing for wind power forecasting is the line of investigation which has the greatest potential, as this technology is taking its first steps, meaning that neuromorphic devices still have little computational power, and the algorithms to train spiking neural networks are still not well defined as the backpropagation algorithm cannot

be applied directly to spikes. This way, future research on neuromorphic algorithms for WPF should be focused on: 1) adapting to the current computational capabilities of neuromorphic devices such as Loihi and Loihi2, and 2) implementing algorithms which can train spikes directly instead of ANN-to-SNN conversion schemes to maximize the energy-efficient features of neuromorphic devices, and 3) implementing online approaches to allow updating the parameters of the spiking neural network in real time with the arrival of new measurements.

Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., et al. (2021). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*.

Abdoos, A. A. (2016). A new intelligent method based on combination of VMD and ELM for short term wind power forecasting. *Neurocomputing*, 203:111–120.

Abedinia, O., Bagheri, M., Naderi, M. S., and Ghadimi, N. (2020a). A new combinatory approach for wind power forecasting. *IEEE Systems Journal*, 14(3):4614–4625.

Abedinia, O., Lotfi, M., Bagheri, M., Sobhani, B., Shafie-Khah, M., and Catalão, J. P. (2020b). Improved EMD-based complex prediction model for wind power forecasting. *IEEE Transactions on Sustainable Energy*, 11(4):2790–2802.

Acikgoz, H., Yildiz, C., and Sekkeli, M. (2020). An extreme learning machine based very short-term wind power forecasting method for complex terrain. *Energy*

Sources, Part A: Recovery, Utilization, and Environmental Effects, 42(22):2715–2730.

Afshari-Igder, M., Niknam, T., and Khooban, M.-H. (2018). Probabilistic wind power forecasting using a novel hybrid intelligent method. *Neural Computing and Applications*, 30(2):473–485.

Aghajani, A., Kazemzadeh, R., and Ebrahimi, A. (2016). A novel hybrid approach for predicting wind farm power production based on wavelet transform, hybrid neural networks and imperialist competitive algorithm. *Energy conversion and management*, 121:232–240.

Ahmad, T. and Zhang, D. (2022). A data-driven deep sequence-to-sequence long-short memory method along with a gated recurrent neural network for wind power forecasting. *Energy*, 239:122109.

Ahmadpour, A. and Farkoush, S. G. (2020). Gaussian models for probabilistic and deterministic Wind Power Prediction: Wind farm and regional. *International Journal of Hydrogen Energy*, 45(51):27779–27791.

Akusok, A., Björk, K.-M., Miche, Y., and Lendasse, A. (2015). High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access*, 3:1011–1025.

Al-Yahyai, S., Charabi, Y., and Gastli, A. (2010). Review of the use of numerical weather prediction (NWP) models for wind energy assessment. *Renewable and Sustainable Energy Reviews*, 14(9):3192–3198.

Alessandrini, S., Delle Monache, L., Sperati, S., and Nissen, J. (2015). A novel application of an analog ensemble for short-term wind power forecasting. *Renewable Energy*, 76:768–781.

Alizadeh, S. and Ozansoy, C. (2016). The role of communications and standardization in wind power applications–A review. *Renewable and Sustainable Energy Reviews*, 54:944–958.

Alkhayat, G. and Mehmood, R. (2021). A review and taxonomy of wind and solar energy forecasting methods based on deep learning. *Energy and AI*, page 100060.

Alshelahi, A., Wang, J., Yu, M., Byon, E., and Saigal, R. (2021). Integrative density forecast and uncertainty quantification of wind power generation. *IEEE Transactions on Sustainable Energy*.

Aly, H. H. (2020). A novel deep learning intelligent clustered hybrid models for wind speed and power forecasting. *Energy*, 213:118773.

Amjady, N., Keynia, F., and Zareipour, H. (2011a). Short-term wind power forecasting using ridgelet neural network. *Electric Power Systems Research*, 81(12):2099–2107.

Amjady, N., Keynia, F., and Zareipour, H. (2011b). Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization. *IEEE transactions on sustainable energy*, 2(3):265–276.

An, X., Jiang, D., Zhao, M., and Liu, C. (2012). Short-term prediction of wind power using EMD and chaotic theory. *Communications in Nonlinear Science and Numerical Simulation*, 17(2):1036–1042.

Applied Brain Research (2021). Converting a Keras model to an SNN on Loihi. https://www.nengo.ai/nengo-loihi/v1.0.0/examples/keras-to-loihi.html. [Online; last accessed 27-December-2021].

Azimi, R., Ghofrani, M., and Ghayekhloo, M. (2016). A hybrid wind power forecasting model based on data mining and wavelets analysis. *Energy conversion and management*, 127:208–225.

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Baringo, L. and Conejo, A. J. (2015). Offering strategy of wind-power producer: A multi-stage risk-constrained approach. *IEEE Transactions on Power Systems*, 31(2):1420–1429.

Baxevani, A. and Lenzi, A. (2018). Very short-term spatio-temporal wind power prediction using a censored Gaussian field. *Stochastic environmental research and risk assessment*, 32(4):931–948.

Bazionis, I. K. and Georgilakis, P. S. (2021). Review of Deterministic and Probabilistic Wind Power Forecasting: Models, Methods, and Future Research. *Electricity*, 2(1):13–47.

Bekolay, T., Bergstra, J., Hunsberger, E., DeWolf, T., Stewart, T. C., Rasmussen, D., Choo, X., Voelker, A., and Eliasmith, C. (2014). Nengo: a Python tool for building large-scale functional brain models. *Frontiers in neuroinformatics*, 7:48.

Bellec, G., Scherr, F., Subramoney, A., Hajek, E., Salaj, D., Legenstein, R., and Maass, W. (2020). A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature communications*, 11(1):1–15.

Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.

Berlinet, A. and Thomas-Agnan, C. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bernardoni, F., Ciri, U., Rotea, M., and Leonardi, S. (2020). Real-time identification of clusters of turbines. In *Journal of Physics: Conference Series*, volume 1618, page 022032. IOP Publishing.

Bessa, R., Moreira, C., Silva, B., and Matos, M. (2014). Handling renewable energy variability and uncertainty in power systems operation. *Wiley Interdisciplinary Reviews: Energy and Environment*, 3(2):156–178.

Bessa, R. J., Matos, M. A., Costa, I. C., Bremermann, L., Franchin, I. G., Pestana, R., Machado, N., Waldl, H.-P., and Wichmann, C. (2012a). Reserve setting and steady-state security assessment using wind power uncertainty forecast: a case study. *IEEE Transactions on Sustainable Energy*, 3(4):827–836.

Bessa, R. J., Miranda, V., Botterud, A., Wang, J., and Constantinescu, E. M. (2012b). Time adaptive conditional kernel density estimation for wind power forecasting. *IEEE Transactions on Sustainable Energy*, 3(4):660–669.

Bessa, R. J., Miranda, V., Botterud, A., Zhou, Z., and Wang, J. (2012c). Time-adaptive quantile-copula for wind power probabilistic forecasting. *Renewable Energy*, 40(1):29–39.

Bessa, R. J., Miranda, V., and Gama, J. (2009). Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. *IEEE Transactions on Power Systems*, 24(4):1657–1666.

Bhaskar, K. and Singh, S. N. (2012). AWNN-assisted wind power forecasting using feed-forward neural network. *IEEE transactions on sustainable energy*, 3(2):306–315.

Blonbou, R. (2011). Very short-term wind power forecasting with neural networks and adaptive Bayesian learning. *Renewable Energy*, 36(3):1118–1124.

Bokde, N., Feijóo, A., Villanueva, D., and Kulat, K. (2019). A review on hybrid empirical mode decomposition models for wind speed and wind power prediction. *Energies*, 12(2):254.

Boshnakov, G. and Halliday, J. (2020). Package 'sarima'. https://cran.r-project.org/web/packages/sarima/sarima.pdf. [Online; last accessed 29-November-2021].

Box, G. and Jenkins, G. M. (1973). Some comments on a paper by Chatfield and Prothero and on a review by Kendall. *Journal of the Royal Statistical Society, Series A*, 136:337–352.

Box, G. E. (1976). Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799.

Box, G. E. P. and Jenkins, G. M. (1970). *Time Series Analysis, Forecasting and Control*. San Francisco: Holden-Day.

Boyd, S., Parikh, N., and Chu, E. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc.

Bremnes, J. B. (2004). Probabilistic wind power forecasts using local quantile regression. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 7(1):47–54.

Brockwell, P. J. and Davis, R. A. (2002). *Introduction to Time Series and Forecasting, 2nd. ed.* Springer-Verlang.

Buhan, S. and Çadırcı, I. (2015). Multistage wind-electric power forecast by using a combination of advanced statistical methods. *IEEE Transactions on Industrial Informatics*, 11(5):1231–1242.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Cadenas, E., Rivera, W., Campos-Amezcua, R., and Heard, C. (2016). Wind speed prediction using a univariate ARIMA model and a multivariate NARX model. *Energies*, 9(2):109.

Cannon, A. J. (2011). Quantile regression neural networks: Implementation in R and application to precipitation downscaling. *Computers & geosciences*, 37(9):1277–1284.

Cao, Y., Chen, Y., and Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1):54–66.

Carpinone, A., Giorgio, M., Langella, R., and Testa, A. (2015). Markov chain modeling for very-short-term wind power forecasting. *Electric power systems research*, 122:152–158.

Carrillo, C., Montaño, A. O., Cidrás, J., and Díaz-Dorado, E. (2013). Review of power curve modelling for wind turbines. *Renewable and Sustainable Energy Reviews*, 21:572–581.

Carvalho, V. R., Moraes, M. F., Braga, A. P., and Mendes, E. M. (2020). Evaluating five different adaptive decomposition methods for EEG signal seizure detection and classification. *Biomedical Signal Processing and Control*, 62:102073.

Catalão, J., Pousinho, H., and Mendes, V. (2010). Hybrid wavelet-PSO-ANFIS approach for short-term wind power forecasting in Portugal. *IEEE Transactions on sustainable energy*, 2(1):50–59.

Catalão, J. P. d. S., Pousinho, H. M. I., and Mendes, V. M. F. (2011). Short-term wind power forecasting in Portugal by neural networks and wavelet transform. *Renewable energy*, 36(4):1245–1251.

Cavalcante, L., Bessa, R. J., Reis, M., and Browell, J. (2017). LASSO vector autoregression structures for very short-term wind power forecasting. *Wind Energy*, 20(4):657–675.

Çevik, H. H., Çunkaş, M., and Polat, K. (2019). A new multistage short-term wind power forecast model using decomposition and artificial intelligence methods. *Physica A: Statistical Mechanics and its Applications*, 534:122177.

Chai, T. and Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific model development*, 7(3):1247–1250.

Chang, G., Lu, H., Chang, Y., and Lee, Y. (2017). An improved neural network-based approach for short-term wind speed and power forecast. *Renewable energy*, 105:301–311.

Chatfield, C. (1996). Model uncertainty and forecast accuracy. *Journal of Forecasting*, 15:495–508.

Chauvin, Y. and Rumelhart, D. E. (2013). *Backpropagation: theory, architectures, and applications*. Psychology press.

Chen, C. (1997). Robustness properties of some forecasting methods for seasonal time series: a Monte Carlo study. *International Journal of Forecasting*, 13(2):269–280.

Chen, C. and Liu, H. (2020). Medium-term wind power forecasting based on multi-resolution multi-learner ensemble and adaptive model selection. *Energy Conversion and Management*, 206:112492.

Chen, H., Zhang, J., Tao, Y., and Tan, F. (2019a). Asymmetric GARCH type models for asymmetric volatility characteristics analysis and wind power forecasting. *Protection and Control of Modern Power Systems*, 4(1):1–11.

Chen, N., Qian, Z., Nabney, I. T., and Meng, X. (2013). Wind power forecasts using Gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems*, 29(2):656–665.

Chen, Y., Zhang, S., Zhang, W., Peng, J., and Cai, Y. (2019b). Multifactor spatio-temporal correlation model based on a combination of convolutional neural network and long short-term memory neural network for wind speed forecasting. *Energy Conversion and Management*, 185:783–799.

Chitsaz, H., Amjady, N., and Zareipour, H. (2015). Wind power forecast using wavelet neural network trained by improved clonal selection algorithm. *Energy conversion and Management*, 89:588–598.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 1724–1734.

Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras. [Online; last accessed 25-March-2022].

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Colak, I., Sagiroglu, S., and Yesilbudak, M. (2012). Data mining and wind power prediction: A literature review. *Renewable Energy*, 46:241–247.

Conejo, A. J., Carrión, M., Morales, J. M., et al. (2010). *Decision making under uncertainty in electricity markets*, volume 1. Springer.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Costa, A., Crespo, A., Navarro, J., Lizcano, G., Madsen, H., and Feitosa, E. (2008). A review on the young history of the wind power short-term prediction. *Renewable and Sustainable Energy Reviews*, 12:1725–1744.

Cox, M. G. and Harris, P. (2004). Statistical error modelling.

Cui, J., Yu, R., Zhao, D., Yang, J., Ge, W., and Zhou, X. (2019). Intelligent load pattern modeling and denoising using improved variational mode decomposition for various calendar periods. *Applied Energy*, 247:480–491.

Daubechies, I. and Bates, B. J. (1993). Ten lectures on wavelets.

Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., Dimou, G., Joshi, P., Imam, N., Jain, S., et al. (2018). Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99.

Davies, M., Wild, A., Orchard, G., Sandamirskaya, Y., Guerra, G. A. F., Joshi, P., Plank, P., and Risbud, S. R. (2021). Advancing neuromorphic computing with Loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934.

Davies, N. and Newbold, P. (1980). Forecasting with misspecified models. *Journal of the Royal Statistical Society, Series C*, 29:87–92.

De Caro, F., De Stefani, J., Vaccaro, A., and Bontempi, G. (2021). DAFT-E: Feature-based Multivariate and Multi-step-ahead Wind Power Forecasting. *IEEE Transactions on Sustainable Energy*.

De Gooijer, J. G. and Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473.

de Simón-Martín, M., de la Puente-Gil, Á., Borge-Diez, D., Ciria-Garcés, T., and González-Martínez, A. (2019). Wind energy planning for a sustainable transition

to a decarbonized generation scenario based on the opportunity cost of the wind energy: Spanish Iberian Peninsula as case study. *Energy Procedia*, 157:1144–1163.

Delgado-Bonal, A. and Marshak, A. (2019). Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy*, 21(6):541.

Demolli, H., Dokuz, A. S., Ecemis, A., and Gokcek, M. (2019). Wind power forecasting based on daily wind speed data using machine learning algorithms. *Energy Conversion and Management*, 198:111823.

Der Kiureghian, A. and Ditlevsen, O. (2009). Aleatory or epistemic? Does it matter? *Structural safety*, 31(2):105–112.

Devi, A. S., Maragatham, G., Boopathi, K., and Rangaraj, A. (2020). Hourly day-ahead wind power forecasting with the EEMD-CSO-LSTM-EFG deep learning technique. *Soft Computing*, 24(16):12391–12411.

DeWolf, T., Jaworski, P., and Eliasmith, C. (2020). Nengo and low-power AI hardware for robust, embedded neurorobotics. *Frontiers in Neurorobotics*, 14.

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., and Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–8. ieee.

Ding, M., Zhou, H., Xie, H., Wu, M., Nakanishi, Y., and Yokoyama, R. (2019). A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting. *Neurocomputing*, 365:54–61.

Dong, J., Dai, W., Tang, L., and Yu, L. (2019). Why do EMD-based methods improve prediction? A multiscale complexity perspective. *Journal of Forecasting*, 38(7):714–731.

Dong, L., Wang, L., Khahro, S. F., Gao, S., and Liao, X. (2016). Wind power day-ahead prediction with cluster analysis of NWP. *Renewable and Sustainable Energy Reviews*, 60:1206–1212.

Dong, Q., Sun, Y., and Li, P. (2017). A novel forecasting model based on a hybrid processing strategy and an optimized local linear fuzzy neural network to make wind power forecasting: A case study of wind farms in China. *Renewable Energy*, 102:241–257.

Dong, W., Sun, H., Tan, J., Li, Z., Zhang, J., and Yang, H. (2022). Regional wind power probabilistic forecasting based on an improved kernel density estimation, regular vine copulas, and ensemble learning. *Energy*, 238:122045.

Dong, Y., Zhang, H., Wang, C., and Zhou, X. (2021a). A novel hybrid model based on Bernstein polynomial with mixture of Gaussians for wind power forecasting. *Applied Energy*, 286:116545.

Dong, Y., Zhang, H., Wang, C., and Zhou, X. (2021b). Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. *Neurocomputing*, 462:169–184.

Dowell, J. and Pinson, P. (2015). Very-short-term probabilistic wind power forecasts by sparse vector autoregression. *IEEE Transactions on Smart Grid*, 7(2):763–770.

Dragomiretskiy, K. and Zosso, D. (2013). Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544.

Draxl, C., Clifton, A., Hodge, B.-M., and McCaa, J. (2015). The wind integration national dataset (WIND) toolkit. *Applied Energy*, 151:355–366.

Du, P., Wang, J., Yang, W., and Niu, T. (2019). A novel hybrid model for short-term wind power forecasting. *Applied Soft Computing*, 80:93–106.

Duan, J., Wang, P., Ma, W., Fang, S., and Hou, Z. (2022). A novel hybrid model based on nonlinear weighted combination for short-term wind power forecasting. *International Journal of Electrical Power & Energy Systems*, 134:107452.

Duan, J., Wang, P., Ma, W., Tian, X., Fang, S., Cheng, Y., Chang, Y., and Liu, H. (2021). Short-term wind power forecasting using the hybrid model of improved variational mode decomposition and Correntropy Long Short-term memory neural network. *Energy*, 214:118980.

Effenberger, N. and Ludwig, N. (2022). A collection and categorization of open-source wind and wind power datasets. *Wind Energy*.

EirGrid (2019). All-Island Ten-Year Transmission Forecast Statement. http://www.eirgridgroup.com/site-files/library/EirGrid/All-Island-Ten-Year-Transmission-Forecast-Statement-2019.pdf. [Online; last accessed 29-November-2021].

EirGrid (2022). EirGrid: System and Renewable Reports. https://www.eirgridgroup.com/how-the-grid-works/renewables/. [Online; last accessed 16-February-2022].

ELIA (2022). ELIA: Wind power generation. https://www.elia.be/en/grid-data/power-generation/wind-power-generation. [Online; last accessed 22-Feb-2022].

Eliasmith, C. and Anderson, C. H. (2003). *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press.

Elsner, J. B. and Tsonis, A. A. (1996). *Singular spectrum analysis: a new tool in time series analysis*. Springer Science & Business Media.

ENTSO-E (2021). Hourly load demand data. https://www.entsoe.eu/data/power-stats/. [Online; last accessed 12-January-2022].

Erdem, E. and Shi, J. (2011). ARMA based approaches for forecasting the tuple of wind speed and direction. *Applied Energy*, 88(4):1405–1414.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Ezzat, A. A. (2020). Turbine-specific short-term wind speed forecasting considering within-farm wind field dependencies and fluctuations. *Applied Energy*, 269:115034.

Faber, M. H. (2005). On the treatment of uncertainties and probabilities in engineering decision analysis.

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P.-A. (2019). Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963.

Feng, C., Cui, M., Hodge, B.-M., and Zhang, J. (2017). A data-driven multi-model methodology with deep feature selection for short-term wind forecasting. *Applied Energy*, 190:1245–1257.

Fine, T. L. (2006). *Feedforward neural network methodology*. Springer Science & Business Media.

Foley, A. M., Leahy, P. G., Marvuglia, A., and McKeogh, E. J. (2012). Current methods and advances in forecasting of wind power generation. *Renewable Energy*, 37:1–8.

Frank, H. P. and Landberg, L. (1997). Modelling the wind climate of Ireland. *Boundary-Layer Meteorology*, 85(3):359–377.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Fuller, W. A. (1996). *Introduction to Statistical Time Series Analysis*. San Francisco:John Wiley & Sons.

Gaffney, F., Deane, J. P., and Gallachóir, B. Ó. (2019). Reconciling high renewable electricity ambitions with market economics and system operation: Lessons from Ireland's power system. *Energy Strategy Reviews*, 26:100381.

Gallego-Castillo, C., Bessa, R., Cavalcante, L., and Lopez-Garcia, O. (2016). On-line quantile regression in the RKHS (Reproducing Kernel Hilbert Space) for operational probabilistic forecasting of wind power. *Energy*, 113:355–365.

Gallego-Castillo, C., Cuerva-Tejero, A., and Lopez-Garcia, O. (2015). A review on the recent history of wind power ramp forecasting. *Renewable and Sustainable Energy Reviews*, 52:1148–1157.

Gan, Z., Li, C., Zhou, J., and Tang, G. (2021). Temporal convolutional networks interval prediction model for wind speed forecasting. *Electric Power Systems Research*, 191:106865.

Gendeel, M., Yuxian, Z., and Aoqi, H. (2018). Performance comparison of ANNs model with VMD for short-term wind speed forecasting. *IET Renewable Power Generation*, 12(12):1424–1430.

Gendeel, M., Zhang, Y., Qian, X., and Xing, Z. (2021). Deterministic and probabilistic interval prediction for wind farm based on VMD and weighted LS-SVM. *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, 43(7):800–814.

Ghosh, B., Basu, B., and O'Mahony, M. (2009). Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE transactions on intelligent transportation systems*, 10(2):246–254.

Ghoushchi, S. J., Manjili, S., Mardani, A., and Saraji, M. K. (2021). An extended new approach for forecasting short-term wind power using modified fuzzy wavelet neural network: A case study in wind power plant. *Energy*, 223:120052.

Giebel, G., Cline, J., Frank, H., Shaw, W., Pinson, P., Hodge, B.-M., Kariniotakis, G., Madsen, J., and Möhrlen, C. (2016). Wind power forecasting: IEA Wind Task 36 & future research issues. In *Journal of Physics: Conference Series*, volume 753, page 032042. IOP Publishing.

Giebel, G., Draxl, C., Brownsword, R., Kariniotakis, G., and Denhard, M. (2011). The state-of-the-art in short-term prediction of wind power. A literature overview.

Giebel, G. and Kariniotakis, G. (2017). Wind power forecasting—a review of the state of the art. *Renewable energy forecasting*, pages 59–109.

Giebel, G., Shaw, W., Frank, H., Draxl, C., Zack, J., Pinson, P., Möhrlen, C., Kariniotakis, G., and Bessa, R. (2020). IEA Wind Task 36-An Overview. In *19th Wind Integration Workshop 2020*.

Gilbert, C., Browell, J., and McMillan, D. (2019). Leveraging turbine-level data for

improved probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy*, 11(3):1152–1160.

Gilbert, C., Messner, J. W., Pinson, P., Trombe, P.-J., Verzijlbergh, R., van Dorp, P., and Jonker, H. (2020). Statistical post-processing of turbulence-resolving weather forecasts for offshore wind power forecasting. *Wind Energy*, 23(4):884–897.

Gneiting, T. and Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151.

Gonzalez, E., Stephen, B., Infield, D., and Melero, J. J. (2019). Using high-frequency SCADA data for wind turbine performance monitoring: A sensitivity study. *Renewable energy*, 131:841–853.

González-Aparicio, I. and Zucker, A. (2015). Impact of wind power uncertainty forecasting on the market integration of wind energy in Spain. *Applied energy*, 159:334–349.

González Sopeña, J., Pakrashi, V., and Ghosh, B. (2021a). An overview of performance evaluation metrics for short-term statistical wind power forecasting. *Renewable and Sustainable Energy Reviews*, 138:110515.

González Sopeña, J. M., Pakrashi, V., and Ghosh, B. (2020). Multi-step ahead wind power forecasting for Ireland using an ensemble of VMD-ELM models. In *2020 31st Irish Signals and Systems Conference (ISSC)*, pages 1–5. IEEE.

González Sopeña, J. M., Pakrashi, V., and Ghosh, B. (2021b). Can we improve short-term wind power forecasts using turbine-level data? A case study in Ireland. In *2021 IEEE Madrid PowerTech*, pages 1–6. IEEE.

González Sopeña, J. M., Pakrashi, V., and Ghosh, B. (2021c). Decomposition-based hybrid models for very short-term wind power forecasting. In *Engineering Proceedings*, volume 5, page 39. Multidisciplinary Digital Publishing Institute.

González Sopeña, J., Maury, C., Pakrashi, V., and Ghosh, B. (2022a). Turbine-level clustering for improved short-term wind power forecasting. In *Journal of Physics: Conference Series*, volume 2265, page 022052. IOP Publishing.

González Sopeña, J. M., Dasgupta, T., Pakrashi, V., Xiao, H., and Ghosh, B. (2022b). How poorly does my forecast perform under a wrong ARIMA model? A decomposition-based simulation algorithm for assessment. https://www.stat.rutgers.edu/home/hxiao/arima.pdf. [Preprint].

González Sopeña, J. M., Pakrashi, V., and Ghosh, B. (2022c). A benchmarking framework for performance evaluation of statistical wind power forecasting models. http://dx.doi.org/10.2139/ssrn.4129779. [Preprint].

González Sopeña, J. M., Pakrashi, V., and Ghosh, B. (2022d). A spiking neural network based wind power forecasting model for neuromorphic devices. *Energies*. In press.

Gonçalves, C., Pinson, P., and Bessa, R. J. (2020). Towards data markets in renewable energy forecasting. *IEEE Transactions on Sustainable Energy*, 12(1):533–542.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Grigonytė, E. and Butkevičiūtė, E. (2016). Short-term wind speed forecasting using ARIMA model. *Energetika*, 62(1-2).

Gu, B., Zhang, T., Meng, H., and Zhang, J. (2021). Short-term forecasting and uncertainty analysis of wind power based on long short-term memory, cloud model and non-parametric kernel density estimation. *Renewable Energy*, 164:687–708.

Guo, H. and Gelfand, S. B. (1990). Analysis of gradient descent learning algorithms for multilayer feedforward neural networks. In *29th IEEE Conference on Decision and Control*, pages 1751–1756. IEEE.

Hahn, B., Welte, T., Faulstich, S., Bangalore, P., Boussion, C., Harrison, K., Miguelanez-Martin, E., O'Connor, F., Pettersson, L., Soraghan, C., et al. (2017). Recommended practices for wind farm data collection and reliability assessment for O&M optimization. *Energy Procedia*, 137:358–365.

Han, L., Jing, H., Zhang, R., and Gao, Z. (2019a). Wind power forecast based on improved Long Short Term Memory network. *Energy*, 189:116300.

Han, L., Romero, C. E., and Yao, Z. (2015). Wind power forecasting based on principle component phase space reconstruction. *Renewable Energy*, 81:737–744.

Han, L., Zhang, R., Wang, X., Bao, A., and Jing, H. (2019b). Multi-step wind power forecast based on VMD-LSTM. *IET Renewable Power Generation*, 13(10):1690–1700.

Hanifi, S., Liu, X., Lin, Z., and Lotfian, S. (2020). A critical review of wind power forecasting methods—past, present and future. *Energies*, 13(15):3764.

Hao, Y., Dong, L., Liao, X., Liang, J., Wang, L., and Wang, B. (2019). A novel clustering algorithm based on mathematical morphology for wind power generation prediction. *Renewable energy*, 136:572–585.

Hao, Y. and Tian, C. (2019). A novel two-stage forecasting model based on error factor and ensemble method for multi-step wind power forecasting. *Applied energy*, 238:368–383.

Haque, A. U., Nehrir, M. H., and Mandal, P. (2014). A hybrid intelligent model for deterministic and quantile regression approach for probabilistic wind power forecasting. *IEEE Transactions on power systems*, 29(4):1663–1672.

He, Y. and Li, H. (2018). Probability density forecasting of wind power using quantile regression neural network and kernel density estimation. *Energy conversion and management*, 164:374–384.

He, Y., Li, H., Wang, S., and Yao, X. (2021). Uncertainty analysis of wind power probability density forecasting based on cubic spline interpolation and support vector quantile regression. *Neurocomputing*, 430:121–137.

Heinermann, J. and Kramer, O. (2016). Machine learning ensembles for wind power prediction. *Renewable Energy*, 89:671–679.

Hestenes, M. R. (1969). Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hong, D., Ji, T., Li, M., and Wu, Q. (2019). Ultra-short-term forecast of wind speed and wind power based on morphological high frequency filter and double similarity search algorithm. *International Journal of Electrical Power & Energy Systems*, 104:868–879.

Hong, K., Wang, L., and Xu, S. (2018). A variational mode decomposition approach for degradation assessment of power transformer windings. *IEEE Transactions on Instrumentation and Measurement*, 68(4):1221–1229.

Hong, T. and Fan, S. (2016). Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914–938.

Hong, T., Pinson, P., and Fan, S. (2014). Global energy forecasting competition 2012. *International Journal of Forecasting*, 30(2):357–363.

Hong, T., Pinson, P., Fan, S., Zareipour, H., Troccoli, A., and Hyndman, R. J. (2016). Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *International Journal of forecasting*, 32(3):896–913.

Hong, Y.-Y., Chang, H.-L., and Chiu, C.-S. (2010). Hour-ahead wind power and speed forecasting using simultaneous perturbation stochastic approximation (SPSA) algorithm and neural network with fuzzy inputs. *Energy*, 35(9):3870–3876.

Hong, Y.-Y. and Rioflorido, C. L. P. P. (2019). A hybrid deep learning-based neural network for 24-h ahead wind power forecasting. *Applied Energy*, 250:530–539.

Hossain, M. A., Chakrabortty, R. K., Elsawah, S., and Ryan, M. J. (2021). Very short-term forecasting of wind power generation using hybrid deep learning model. *Journal of Cleaner Production*, 296:126564.

Hu, J., Lin, Y., Tang, J., and Zhao, J. (2020a). A new wind power interval prediction approach based on reservoir computing and a quality-driven loss function. *Applied Soft Computing*, 92:106327.

Hu, J., Luo, Q., Tang, J., Heng, J., and Deng, Y. (2022). Conformalized temporal convolutional quantile regression networks for wind power interval forecasting. *Energy*, page 123497.

Hu, J., Tang, J., and Lin, Y. (2020b). A novel wind power probabilistic forecasting approach based on joint quantile regression and multi-objective optimization. *Renewable Energy*, 149:141–164.

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., and Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995.

Huang, Y., Li, J., Hou, W., Zhang, B., Zhang, Y., Li, Y., and Sun, L. (2020). Improved clustering and deep learning based short-term wind energy forecasting in large-scale wind farms. *Journal of Renewable and Sustainable Energy*, 12(6):066101.

Hüllermeier, E. and Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506.

Hunsberger, E. and Eliasmith, C. (2015). Spiking deep networks with LIF neurons. *arXiv preprint arXiv:1510.08829*.

Hunsberger, E. and Eliasmith, C. (2016). Training spiking deep networks for neuromorphic hardware. *arXiv preprint arXiv:1611.05141*.

Hyndman, R. J. (2020). A brief history of forecasting competitions. *International Journal of Forecasting*, 36(1):7–14.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.

Hyndman, R. J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., and Razbash, S. (2020). Package 'forecast'. *[Online, last accessed 17-Jan-2021] https://cran.r-project.org/web/packages/forecast/forecast.pdf*.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688.

Intel's Neuromorphic Computing Lab (2021). Lava: A Software Framework for Neuromorphic Computing . https://github.com/lava-nc/lava. [Online; last accessed 25-March-2022].

International Energy Agency (2021). World Energy Outlook 2021. https://www.iea.org/reports/world-energy-outlook-2021. [Online; last accessed 14-March-2022].

Iversen, E. B., Morales, J. M., Møller, J. K., Trombe, P.-J., and Madsen, H. (2017). Leveraging stochastic differential equations for probabilistic forecasting of wind power using a dynamic power curve. *Wind Energy*, 20(1):33–44.

Jaeger, H. (2007). Echo state network. *scholarpedia*, 2(9):2330.

Jiang, Y., Xingying, C., Kun, Y., and Yingchen, L. (2017). Short-term wind power forecasting using hybrid method based on enhanced boosting algorithm. *Journal of Modern Power Systems and Clean Energy*, 5(1):126–133.

Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., and Rehman, M. U. (2019). A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting. *Ieee Access*, 7:28309–28318.

Jung, J. and Broadwater, R. P. (2014). Current status and future advances for wind speed and power forecasting. *Renewable and Sustainable Energy Reviews*, 31:762–777.

Jursa, R. and Rohrig, K. (2008). Short-term wind power forecasting using evolutionary algorithms for the automated specification of artificial intelligence models. *International Journal of Forecasting*, 24(4):694–709.

Karakuş, O., Kuruoğlu, E. E., and Altınkaya, M. A. (2017). One-day ahead wind speed/power prediction based on polynomial autoregressive model. *IET Renewable Power Generation*, 11(11):1430–1439.

Kasabov, N., Scott, N. M., Tu, E., Marks, S., Sengupta, N., Capecci, E., Othman, M., Doborjeh, M. G., Murli, N., Hartono, R., et al. (2016). Evolving spatio-temporal data machines based on the NeuCube neuromorphic framework: Design methodology and selected applications. *Neural Networks*, 78:1–14.

Kasabov, N. K. (2019). *Time-space, spiking neural networks and brain-inspired artificial intelligence*. Springer.

Kavousi-Fard, A., Khosravi, A., and Nahavandi, S. (2015). A new fuzzy-based combined prediction interval for wind power forecasting. *IEEE Transactions on Power Systems*, 31(1):18–26.

Kendall, M. G. (1971). Review of Box and Jenkins (1970). *Journal of the Royal Statistical Society, Series A*, 134:450–453.

Khan, K., Rehman, S. U., Aziz, K., Fong, S., and Sarasvady, S. (2014). DBSCAN: Past, present and future. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)*, pages 232–238. IEEE.

Khazaei, S., Ehsan, M., Soleymani, S., and Mohammadnezhad-Shourkaei, H. (2022). A high-accuracy hybrid method for short-term wind power forecasting. *Energy*, 238:122020.

Khorramdel, B., Chung, C., Safari, N., and Price, G. (2018). A fuzzy adaptive probabilistic wind power prediction framework using diffusion kernel density estimators. *IEEE Transactions on Power Systems*, 33(6):7109–7121.

Khosravi, A. and Nahavandi, S. (2013). Combined nonparametric prediction intervals for wind power generation. *IEEE Transactions on Sustainable Energy*, 4(4):849–856.

Khosravi, A. and Nahavandi, S. (2014). Closure to the discussion of "prediction intervals for short-term wind farm generation forecasts" and "combined nonparametric prediction intervals for wind power generation" and the discussion of "combined nonparametric prediction intervals for wind power generation". *IEEE Transactions on Sustainable Energy*, 5(3):1022–1023.

Khosravi, A., Nahavandi, S., and Creighton, D. (2013). Prediction intervals for short-term wind farm power generation forecasts. *IEEE Transactions on sustainable energy*, 4(3):602–610.

Khosravi, A., Nahavandi, S., Creighton, D., and Atiya, A. F. (2010). Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE transactions on neural networks*, 22(3):337–346.

Kim, S. and Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32(3):669–679.

Koenker, R. and Bassett Jr, G. (1978). Regression quantiles. *Econometrica*, 46(1):33–50.

Koenker, R. and Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, 15(4):143–156.

Korprasertsak, N. and Leephakpreeda, T. (2019). Robust short-term prediction of wind power generation under uncertainty via statistical interpretation of multiple forecasting models. *Energy*, 180:387–397.

Kou, P., Gao, F., and Guan, X. (2013). Sparse online warped Gaussian process for wind power probabilistic forecasting. *Applied energy*, 108:410–428.

Kusiak, A. and Zhang, Z. (2010). Short-horizon prediction of wind power: A data-driven approach. *IEEE Transactions on Energy Conversion*, 25(4):1112–1122.

Lago, J., Marcjasz, G., De Schutter, B., and Weron, R. (2021). Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983.

Lahouar, A. and Slama, J. B. H. (2017). Hour-ahead wind power forecast based on random forests. *Renewable energy*, 109:529–541.

Landberg, L., Giebel, G., Nielsen, H. A., Nielsen, T., and Madsen, H. (2003). Short-term prediction—an overview. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 6(3):273–280.

Landry, M., Erlinger, T. P., Patschke, D., and Varrichio, C. (2016). Probabilistic gradient boosting machines for GEFCom2014 wind forecasting. *International Journal of Forecasting*, 32(3):1061–1066.

Laszuk, D. (2017). Python implementation of Empirical Mode Decomposition algorithm. https://github.com/laszukdawid/PyEMD. [Online; last accessed 25-March-2022].

Lee, D. and Baldick, R. (2013). Short-term wind power ensemble prediction based on Gaussian processes and neural networks. *IEEE Transactions on Smart Grid*, 5(1):501–510.

Lei, M., Shiyan, L., Chuanwen, J., Hongling, L., and Yan, Z. (2009). A review on the forecasting of wind speed and generated power. *Renewable and sustainable energy reviews*, 13(4):915–920.

Leñero-Bardallo, J. A., Carmona-Galán, R., and Rodríguez-Vázquez, A. (2018). Applications of event-based image sensors—Review and analysis. *International Journal of Circuit Theory and Applications*, 46(9):1620–1630.

Leng, H., Li, X., Zhu, J., Tang, H., Zhang, Z., and Ghadimi, N. (2018). A new wind power prediction method based on ridgelet transforms, hybrid feature selection and closed-loop forecasting. *Advanced Engineering Informatics*, 36:20–30.

Lenzi, A., Steinsland, I., and Pinson, P. (2018). Benefits of spatiotemporal modeling for short-term wind power forecasting at both individual and aggregated levels. *Environmetrics*, 29(3):e2493.

Li, C., Tang, G., Xue, X., Chen, X., Wang, R., and Zhang, C. (2020a). The short-term interval prediction of wind power using the deep learning model with gradient descend optimization. *Renewable Energy*, 155:197–211.

Li, L., Li, Y., Zhou, B., Wu, Q., Shen, X., Liu, H., and Gong, Z. (2020b). An adaptive time-resolution method for ultra-short-term wind power prediction. *International Journal of Electrical Power & Energy Systems*, 118:105814.

Li, L., Yin, X.-L., Jia, X.-C., and Sobhani, B. (2020c). Day ahead powerful probabilistic wind power forecast using combined intelligent structure and fuzzy clustering algorithm. *Energy*, 192:116498.

Li, L.-L., Chang, Y.-B., Tseng, M.-L., Liu, J.-Q., and Lim, M. K. (2020d). Wind power prediction using a novel model on wavelet decomposition-support vector machines-improved atomic search algorithm. *Journal of Cleaner Production*, 270:121817.

Li, L.-L., Zhao, X., Tseng, M.-L., and Tan, R. R. (2020e). Short-term wind power forecasting based on support vector machine with improved dragonfly algorithm. *Journal of Cleaner Production*, 242:118447.

Li, S., Wang, P., and Goel, L. (2015). Wind power forecasting using neural network ensembles with feature selection. *IEEE Transactions on sustainable energy*, 6(4):1447–1456.

Li, W., Yang, T., Delicato, F. C., Pires, P. F., Tari, Z., Khan, S. U., and Zomaya, A. Y. (2018). On enabling sustainable edge computing with renewable energy resources. *IEEE Communications Magazine*, 56(5):94–101.

Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, Z., Ye, L., Zhao, Y., Song, X., Teng, J., and Jin, J. (2016). Short-term wind power prediction based on extreme learning machine with error correction. *Protection and Control of Modern Power Systems*, 1(1):1–8.

Lim, B. and Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.

Lin, C.-K., Wild, A., Chinya, G. N., Cao, Y., Davies, M., Lavery, D. M., and Wang, H. (2018a). Programming spiking neural networks on Intel's Loihi. *Computer*, 51(3):52–61.

Lin, Y., Yang, M., Wan, C., Wang, J., and Song, Y. (2018b). A multi-model combination approach for probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy*, 10(1):226–237.

Lin, Z. and Liu, X. (2020). Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network. *Energy*, 201:117693.

Lin, Z., Liu, X., and Collu, M. (2020). Wind power prediction based on high-frequency SCADA data along with isolation forest and deep learning neural networks. *International Journal of Electrical Power & Energy Systems*, 118:105835.

Lipu, M. H., Miah, M. S., Hannan, M., Hussain, A., Sarker, M. R., Ayob, A., Saad, M. H. M., and Mahmud, M. S. (2021). Artificial Intelligence Based Hybrid Forecasting Approaches for Wind Power Generation: Progress, Challenges and Prospects. *IEEE Access*, 9:102460–102489.

Liu, H. and Chen, C. (2019). Data processing strategies in wind energy forecasting models and applications: A comprehensive review. *Applied Energy*, 249:392–408.

Liu, H., Chen, C., Lv, X., Wu, X., and Liu, M. (2019a). Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods. *Energy Conversion and Management*, 195:328–345.

Liu, H. and Duan, Z. (2020). Corrected multi-resolution ensemble model for wind

power forecasting with real-time decomposition and Bivariate Kernel density estimation. *Energy Conversion and Management*, 203:112265.

Liu, H., Li, Y., Duan, Z., and Chen, C. (2020). A review on multi-objective optimization framework in wind energy forecasting techniques and applications. *Energy Conversion and Management*, 224:113324.

Liu, H., Tian, H.-Q., Chen, C., and Li, Y.-f. (2010). A hybrid statistical method to predict wind speed and wind power. *Renewable energy*, 35(8):1857–1861.

Liu, J., Wang, X., and Lu, Y. (2017). A novel hybrid methodology for short-term wind power forecasting based on adaptive neuro-fuzzy inference system. *Renewable energy*, 103:620–629.

Liu, M.-D., Ding, L., and Bai, Y.-L. (2021a). Application of hybrid model based on empirical mode decomposition, novel recurrent neural networks and the ARIMA to wind speed prediction. *Energy Conversion and Management*, 233:113917.

Liu, X., Lin, Z., and Feng, Z. (2021b). Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM. *Energy*, 227:120492.

Liu, X., Yang, L., and Zhang, Z. (2021c). Short-term multi-step ahead wind power predictions based on a novel deep convolutional recurrent network method. *IEEE Transactions on Sustainable Energy*.

Liu, Y., Wang, H., Han, S., Yan, J., Li, L., and Chen, Z. (2019b). Quantitative method for evaluating detailed volatility of wind power at multiple temporal-spatial scales. *Global Energy Interconnection*, 2(4):318–327.

Lobo, J. L., Del Ser, J., Bifet, A., and Kasabov, N. (2020). Spiking neural networks and online learning: An overview and perspectives. *Neural Networks*, 121:88–100.

Lu, H., Ma, X., Huang, K., and Azimi, M. (2020a). Prediction of offshore wind farm power using a novel two-stage model combining kernel-based nonlinear extension of the Arps decline model with a multi-objective grey wolf optimizer. *Renewable and Sustainable Energy Reviews*, 127:109856.

Lu, P., Ye, L., Zhao, Y., Dai, B., Pei, M., and Tang, Y. (2021). Review of meta-heuristic algorithms for wind power prediction: Methodologies, applications and challenges. *Applied Energy*, 301:117446.

Lu, P., Ye, L., Zhong, W., Qu, Y., Zhai, B., Tang, Y., and Zhao, Y. (2020b). A novel spatio-temporal wind power forecasting framework based on multi-output support vector machine and optimization strategy. *Journal of Cleaner Production*, 254:119993.

Lv, J., Zheng, X., Pawlak, M., Mo, W., and Miśkowicz, M. (2021). Very short-term probabilistic wind power prediction using sparse machine learning and nonparametric density estimation algorithms. *Renewable Energy*.

Ma, Q., Zheng, J., Li, S., and Cottrell, G. W. (2019). Learning representations for time series clustering. *Advances in neural information processing systems*, 32:3781–3791.

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671.

Mac Nally, R., Duncan, R. P., Thomson, J. R., and Yen, J. D. (2018). Model selection using information criteria, but is the "best" model any good? *Journal of Applied Ecology*, 55(3):1441–1444.

Madsen, H., Pinson, P., Kariniotakis, G., Nielsen, H. A., and Nielsen, T. S. (2005). Standardizing the performance evaluation of short-term wind power prediction models. *Wind engineering*, 29(6):475–489.

Mahmoud, T., Dong, Z., and Ma, J. (2018). An advanced approach for optimal wind power generation prediction intervals by using self-adaptive evolutionary extreme learning machine. *Renewable energy*, 126:254–269.

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889.

Maldonado-Correa, J., Solano, J., and Rojas-Moncayo, M. (2021). Wind power forecasting: A systematic literature review. *Wind Engineering*, 45(2):413–426.

Mangan, N. M., Kutz, J. N., Brunton, S. L., and Proctor, J. L. (2017). Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009.

Marugán, A. P., Márquez, F. P. G., Perez, J. M. P., and Ruiz-Hernández, D. (2018). A survey of artificial neural network in wind energy systems. *Applied energy*, 228:1822–1836.

Mazzeu, J. H. G., Ruiz, E., and Veiga, H. (2018). Uncertainty and density forecasts of arma models: comparison of asymptotic, bayesian, and bootstrap procedures. *Journal of Economic Surveys*, 32:388–419.

Mazzi, N., Lorenzoni, A., Rech, S., and Lazzaretto, A. (2015). Electricity auctions: a European view on markets and practices. In *2015 12th International Conference on the European Energy Market (EEM)*, pages 1–5. IEEE.

Mc Garrigle, E. V. and Leahy, P. G. (2015). Quantifying the value of improved wind energy forecasts in a pool-based electricity market. *Renewable Energy*, 80:517–524.

Mehrjoo, M., Jozani, M. J., Pawlak, M., and Bagen, B. (2021). A multilevel modeling approach towards wind farm aggregated power curve. *IEEE Transactions on Sustainable Energy*.

Meka, R., Alaeddini, A., and Bhaganagar, K. (2021). A robust deep learning framework for short-term wind power forecast of a full-scale wind farm using atmospheric variables. *Energy*, 221:119759.

Meng, A., Chen, S., Ou, Z., Ding, W., Zhou, H., Fan, J., and Yin, H. (2022). A hybrid deep learning architecture for wind power prediction based on bi-attention mechanism and crisscross optimization. *Energy*, 238:121795.

Messner, J. W. and Pinson, P. (2019). Online adaptive lasso estimation in vector

autoregressive models for high dimensional wind power forecasting. *International Journal of Forecasting*, 35(4):1485–1498.

Meyer, Y. (1992). *Wavelets and Operators: Volume 1*. Number 37. Cambridge University Press.

Mishra, S. and Dash, P. (2019). Short-term prediction of wind power using a hybrid pseudo-inverse Legendre neural network and adaptive firefly algorithm. *Neural Computing and Applications*, 31(7):2243–2268.

Mitchell, T. M., Keller, R. M., and Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. *Machine learning*, 1(1):47–80.

Möhrlen, C., Lerner, J., Messner, J. W., Browell, J., Tuohy, A., Zack, J., Collier, C., and Giebel, G. (2018). IEA wind recommended practices for the implementation of wind power forecasting solutions part 2 and 3: designing and executing forecasting benchmarks and evaluation of forecast solutions. In *17th Wind Integration Workshop*.

Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. (2018). DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access*, 7:1991–2005.

Nagy, G. I., Barta, G., Kazi, S., Borbély, G., and Simon, G. (2016). GEFCom2014: Probabilistic solar and wind power forecasting using a generalized additive tree ensemble approach. *International Journal of Forecasting*, 32(3):1087–1093.

Naik, J., Bisoi, R., and Dash, P. (2018a). Prediction interval forecasting of wind speed and wind power using modes decomposition based low rank multi-kernel ridge regression. *Renewable energy*, 129:357–383.

Naik, J., Dash, S., Dash, P., and Bisoi, R. (2018b). Short term wind power forecasting using hybrid variational mode decomposition and multi-kernel regularized pseudo inverse neural network. *Renewable Energy*, 118:180–212.

Naik, J., Satapathy, P., and Dash, P. (2018c). Short-term wind speed and wind power

prediction using hybrid empirical mode decomposition and kernel ridge regression. *Applied Soft Computing*, 70:1167–1188.

Nazaré, G., Castro, R., and Gabriel Filho, L. R. (2020). Wind power forecast using neural networks: Tuning with optimization techniques and error analysis. *Wind Energy*, 23(3):810–824.

Nazir, M. S., Alturise, F., Alshmrany, S., Nazir, H., Bilal, M., Abdalla, A. N., Sanjeevikumar, P., M Ali, Z., et al. (2020). Wind generation forecasting methods and proliferation of artificial neural network: a review of five years research trend. *Sustainability*, 12(9):3778.

Nielsen, H. A., Madsen, H., and Nielsen, T. S. (2006). Using quantile regression to extend an existing wind power forecasting system with probabilistic forecasts. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 9(1-2):95–108.

Nielson, J., Bhaganagar, K., Meka, R., and Alaeddini, A. (2020). Using atmospheric inputs for artificial neural networks to improve wind turbine power prediction. *Energy*, 190:116273.

Niu, Z., Yu, Z., Tang, W., Wu, Q., and Reformat, M. (2020). Wind power forecasting using attention-based gated recurrent unit network. *Energy*, 196:117081.

Okumus, I. and Dinler, A. (2016). Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Conversion and Management*, 123:362–371.

Orchard, G., Frady, E. P., Rubin, D. B. D., Sanborn, S., Shrestha, S. B., Sommer, F. T., and Davies, M. (2021). Efficient Neuromorphic Signal Processing with Loihi 2. In *2021 IEEE Workshop on Signal Processing Systems (SiPS)*, pages 254–259. IEEE.

Oreshkin, B. N., Carpov, D., Chapados, N., and Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Orlov, A., Sillmann, J., and Vigo, I. (2020). Better seasonal forecasts for the renewable energy industry. *Nature Energy*, 5(2):108–110.

Osório, G., Matias, J., and Catalão, J. (2015). Short-term wind power forecasting using adaptive neuro-fuzzy inference system combined with evolutionary particle swarm optimization, wavelet transform and mutual information. *Renewable Energy*, 75:301–307.

Ouyang, T., Huang, H., He, Y., and Tang, Z. (2020). Chaotic wind power time series prediction via switching data-driven modes. *Renewable Energy*, 145:270–281.

Ozkan, M. B. and Karagoz, P. (2015). A novel wind power forecast model: Statistical hybrid wind power forecast technique (SHWIP). *IEEE Transactions on industrial informatics*, 11(2):375–387.

Pakrashi, V., Bhowmik, B., González Sopeña, J. M., Mucchielli, P., and Ghosh, B. (2020). Wind power prediction and early downtime detection for Ireland. In *Proceedings of the Civil Engineering Research Ireland 2020 Conference*.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR.

Patel, K., Hunsberger, E., Batir, S., and Eliasmith, C. (2021). A spiking neural network for image segmentation. *arXiv preprint arXiv:2106.08921*.

Peinke, J., Barth, S., Böttcher, F., Heinemann, D., and Lange, B. (2004). Turbulence, a challenging problem for wind energy. *Physica A: Statistical Mechanics and its Applications*, 338(1-2):187–193.

Peng, H., Liu, F., and Yang, X. (2013). A hybrid strategy of short term wind power prediction. *Renewable Energy*, 50:590–595.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., et al. (2022). Forecasting: theory and practice. *International Journal of Forecasting*.

Petropoulos, F., Hyndman, R. J., and Bergmeir, C. (2018). Exploring the sources of uncertainty: Why does bagging for time series forecasting work? *European Journal of Operational Research*, 268(2):545–554.

Pichault, M., Vincent, C., Skidmore, G., and Monty, J. (2022). LiDAR-based detection of wind gusts: An experimental study of gust propagation speed and impact on wind power ramps. *Journal of Wind Engineering and Industrial Aerodynamics*, 220:104864.

Pincus, S. M. (1991). Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301.

Pinson, P. (2012). Very-short-term probabilistic forecasting of wind power with generalized logit–normal distributions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 61(4):555–576.

Pinson, P., Chevallier, C., and Kariniotakis, G. N. (2007a). Trading wind generation from short-term probabilistic forecasts of wind power. *IEEE Transactions on Power Systems*, 22(3):1148–1156.

Pinson, P. and Kariniotakis, G. (2004). On-line assessment of prediction risk for wind power production forecasts. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 7(2):119–132.

Pinson, P. and Kariniotakis, G. (2010). Conditional prediction intervals of wind power generation. *IEEE Transactions on Power Systems*, 25(4):1845–1856.

Pinson, P. and Madsen, H. (2012). Adaptive modelling and forecasting of offshore wind power fluctuations with Markov-switching autoregressive models. *Journal of forecasting*, 31(4):281–313.

Pinson, P., Madsen, H., Nielsen, H. A., Papaefthymiou, G., and Klöckl, B. (2009). From probabilistic forecasts to statistical scenarios of short-term wind power production. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(1):51–62.

Pinson, P., Nielsen, H. A., Møller, J. K., Madsen, H., and Kariniotakis, G. N. (2007b). Non-parametric probabilistic forecasts of wind power: required properties and evaluation. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 10(6):497–516.

Pinson, P. and Tastu, J. (2014). Discussion of "Prediction intervals for short-term wind farm generation forecasts" and "Combined nonparametric prediction intervals for wind power generation". *IEEE Transactions on Sustainable Energy*, 5(3):1019–1020.

Pollack, J. B. (1990). Backpropagation is sensitive to initial conditions. *Complex systems*, 4:269–280.

Pousinho, H. M. I., Mendes, V. M. F., and Catalão, J. P. d. S. (2011). A hybrid PSO–ANFIS approach for short-term wind power prediction in Portugal. *Energy Conversion and Management*, 52(1):397–402.

Prechelt, L. (1998). Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer.

Prósper, M. A., Otero-Casal, C., Fernández, F. C., and Miguez-Macho, G. (2019). Wind power forecasting for a real onshore wind farm on complex terrain using WRF high resolution simulations. *Renewable energy*, 135:674–686.

Pukkila, T., Koreisha, S., and Kallinen, A. (1990). The Identification of ARMA Models. *Biometrika*, 77:537–548.

Putz, D., Gumhalter, M., and Auer, H. (2021). A novel approach to multi-horizon wind power forecasting based on deep neural architecture. *Renewable Energy*, 178:494–505.

Qian, W. and Wang, J. (2020). An improved seasonal GM (1, 1) model based on the HP filter for forecasting wind power generation in China. *Energy*, 209:118499.

Qian, Z., Pei, Y., Zareipour, H., and Chen, N. (2019). A review and discussion of decomposition-based hybrid models for wind energy forecasting applications. *Applied energy*, 235:939–953.

Quan, H., Khosravi, A., Yang, D., and Srinivasan, D. (2019). A survey of computational intelligence techniques for wind power uncertainty quantification in smart grids. *IEEE transactions on neural networks and learning systems*, 31(11):4582–4599.

Quan, H., Srinivasan, D., and Khosravi, A. (2013). Short-term load and wind power forecasting using neural network-based prediction intervals. *IEEE transactions on neural networks and learning systems*, 25(2):303–315.

Qureshi, A. S. and Khan, A. (2019). Adaptive transfer learning in deep neural networks: Wind power prediction using knowledge transfer from region to region and between different task domains. *Computational Intelligence*, 35(4):1088–1112.

Qureshi, A. S., Khan, A., Zameer, A., and Usman, A. (2017). Wind power prediction using deep neural network based meta regression and transfer learning. *Applied Soft Computing*, 58:742–755.

Radziukynas, V. and Klementavicius, A. (2014). Short-term wind speed forecasting with ARIMA model. In *2014 55th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pages 145–149. IEEE.

Ramirez-Rosado, I. J., Fernandez-Jimenez, L. A., Monteiro, C., Sousa, J., and Bessa, R. (2009). Comparison of two new short-term wind-power forecasting systems. *Renewable Energy*, 34(7):1848–1854.

Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer.

Rasmussen, D. (2019). NengoDL: Combining deep learning and neuromorphic modelling methods. *Neuroinformatics*, 17(4):611–628.

Rayi, V. K., Mishra, S., Naik, J., and Dash, P. (2021). Adaptive VMD based optimized deep learning mixed kernel ELM autoencoder for single and multistep wind power forecasting. *Energy*, page 122585.

Remy, P. (2020). Temporal Convolutional Networks for Keras. https://github.com/philipperemy/keras-tcn. [Online; last accessed 3-March-2022].

Ren, Y., Suganthan, P., and Srikanth, N. (2015). Ensemble methods for wind and solar power forecasting—a state-of-the-art review. *Renewable and Sustainable Energy Reviews*, 50:82–91.

Renani, E. T., Elias, M. F. M., and Rahim, N. A. (2016). Using data-driven approach for wind power prediction: A comparative study. *Energy Conversion and Management*, 118:193–203.

Roungkvist, J. S. and Enevoldsen, P. (2020). Timescale classification in wind forecasting: A review of the state-of-the-art. *Journal of Forecasting*, 39(5):757–768.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., and Liu, S.-C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Sadaei, H. J., e Silva, P. C. d. L., Guimarães, F. G., and Lee, M. H. (2019). Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, 175:365–377.

Safari, N., Chung, C., and Price, G. (2017). Novel multi-step short-term wind power prediction framework based on chaotic time series analysis and singular spectrum analysis. *IEEE Transactions on Power Systems*, 33(1):590–601.

Saleh, A. E., Moustafa, M. S., Abo-Al-Ez, K. M., and Abdullah, A. A. (2016). A hybrid neuro-fuzzy power prediction system for wind energy generation. *International Journal of Electrical Power & Energy Systems*, 74:384–395.

Santhosh, M., Venkaiah, C., and Vinod Kumar, D. (2020). Current advances and approaches in wind speed and wind power forecasting for improved renewable energy integration: A review. *Engineering Reports*, 2(6):e12178.

Sasser, C., Yu, M., and Delgado, R. (2022). Improvement of wind power prediction from meteorological characterization with machine learning models. *Renewable Energy*, 183:491–501.

Schliebs, S. and Kasabov, N. (2013). Evolving spiking neural network—a survey. *Evolving Systems*, 4(2):87–98.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61:85–117.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4):1875–1897.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

Shahid, F., Khan, A., Zameer, A., Arshad, J., and Safdar, K. (2020a). Wind power prediction using a three stage genetic ensemble and auxiliary predictor. *Applied Soft Computing*, 90:106151.

Shahid, F., Zameer, A., Mehmood, A., and Raja, M. A. Z. (2020b). A novel wavenets long short term memory paradigm for wind power prediction. *Applied Energy*, 269:115098.

Shahid, F., Zameer, A., and Muneeb, M. (2021). A novel genetic LSTM model for wind power forecast. *Energy*, 223:120069.

Sharifian, A., Ghadi, M. J., Ghavidel, S., Li, L., and Zhang, J. (2018). A new method based on Type-2 fuzzy neural network for accurate wind power forecasting under uncertain data. *Renewable energy*, 120:220–230.

Shi, J., Ding, Z., Lee, W.-J., Yang, Y., Liu, Y., and Zhang, M. (2013). Hybrid forecasting model for very-short term wind power forecasting based on grey relational

analysis and wind speed distribution features. *IEEE Transactions on Smart Grid*, 5(1):521–526.

Shi, J., Guo, J., and Zheng, S. (2012). Evaluation of hybrid forecasting approaches for wind speed and power generation time series. *Renewable and Sustainable Energy Reviews*, 16(5):3471–3480.

Shi, Z., Liang, H., and Dinavahi, V. (2017). Direct interval forecast of uncertain wind power based on recurrent neural networks. *IEEE Transactions on Sustainable Energy*, 9(3):1177–1187.

Shrestha, S. B. and Orchard, G. (2018). Slayer: Spike layer error reassignment in time. In *Advances in Neural Information Processing Systems*, pages 1412–1421.

Sideratos, G. and Hatziargyriou, N. D. (2007). An advanced statistical method for wind power forecasting. *IEEE Transactions on power systems*, 22(1):258–265.

Sideratos, G. and Hatziargyriou, N. D. (2012). Probabilistic wind power forecasting using radial basis function neural networks. *IEEE Transactions on Power Systems*, 27(4):1788–1796.

Skajaa, A., Edlund, K., and Morales, J. M. (2015). Intraday trading of wind energy. *IEEE Transactions on power systems*, 30(6):3181–3189.

Soman, S. S., Zareipour, H., Malik, O., and Mandal, P. (2010). A review of wind power and wind speed forecasting methods with different time horizons. In *North American Power Symposium 2010*, pages 1–8. IEEE.

Sommer, B., Pinson, P., Messner, J. W., and Obst, D. (2021). Online distributed learning in wind power forecasting. *International Journal of Forecasting*, 37(1):205–223.

Song, H., Zheng, T., Liu, H., and Zhang, H. (2014). Modeling MW-dependent ramp rate in the electricity market. In *2014 IEEE PES General Meeting| Conference & Exposition*, pages 1–5. IEEE.

Sorensen, P., Cutululis, N. A., Vigueras-Rodríguez, A., Jensen, L. E., Hjerrild, J., Donovan, M. H., and Madsen, H. (2007). Power fluctuations from large wind farms. *IEEE Transactions on Power Systems*, 22(3):958–965.

Sorjamaa, A. and Lendasse, A. (2006). Time series prediction using DirRec strategy. In *Esann*, volume 6, pages 143–148. Citeseer.

Soroudi, A. and Amraee, T. (2013). Decision making under uncertainty in energy systems: State of the art. *Renewable and Sustainable Energy Reviews*, 28:376–384.

Sotavento Wind Farm (2022). Sotavento wind farm data. https://www.sotaventogalicia.com/area-tecnica/datos-tiempo-real/datos-tiempo-real-historicos/. [Online; last accessed 30-March-2022].

Stathopoulos, C., Kaperoni, A., Galanis, G., and Kallos, G. (2013). Wind power prediction based on numerical and statistical models. *Journal of Wind Engineering and Industrial Aerodynamics*, 112:25–38.

Stewart, K., Orchard, G., Shrestha, S. B., and Neftci, E. (2020). On-chip few-shot learning with surrogate gradient descent on a neuromorphic processor. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 223–227. IEEE.

Su, Z., Wang, J., Lu, H., and Zhao, G. (2014). A new hybrid model optimized by an intelligent optimization algorithm for wind speed forecasting. *Energy Conversion and Management*, 85:443–452.

Sun, G., Jiang, C., Cheng, P., Liu, Y., Wang, X., Fu, Y., and He, Y. (2018). Short-term wind power forecasts by a synthetical similar time series data mining method. *Renewable energy*, 115:575–584.

Sun, M., Feng, C., and Zhang, J. (2019). Conditional aggregated probabilistic wind power forecasting based on spatio-temporal correlation. *Applied Energy*, 256:113842.

Sun, M., Feng, C., and Zhang, J. (2020a). Multi-distribution ensemble probabilistic wind power forecasting. *Renewable Energy*, 148:135–149.

Sun, Y., Wang, P., Zhai, S., Hou, D., Wang, S., and Zhou, Y. (2020b). Ultra short-term probability prediction of wind power based on LSTM network and condition normal distribution. *Wind Energy*, 23(1):63–76.

Sweeney, C., Bessa, R. J., Browell, J., and Pinson, P. (2020). The future of forecasting for renewable energy. *Wiley Interdisciplinary Reviews: Energy and Environment*, 9(2):e365.

Taherkhani, A., Belatreche, A., Li, Y., and Maguire, L. P. (2015). DL-ReSuMe: A delay learning-based remote supervised method for spiking neurons. *IEEE transactions on neural networks and learning systems*, 26(12):3137–3149.

Taieb, S. B., Bontempi, G., Atiya, A. F., and Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the nn5 forecasting competition. *Expert systems with applications*, 39(8):7067–7083.

Taieb, S. B., Bontempi, G., Sorjamaa, A., and Lendasse, A. (2009). Long-term prediction of time series by combining direct and MIMO strategies. In *2009 International Joint Conference on Neural Networks*, pages 3054–3061. IEEE.

Taieb, S. B., Sorjamaa, A., and Bontempi, G. (2010). Multiple-output modeling for multi-step-ahead time series forecasting. *Neurocomputing*, 73(10-12):1950–1957.

Tan, C., Šarlija, M., and Kasabov, N. (2020). Spiking neural networks: Background, recent development and the NeuCube architecture. *Neural Processing Letters*, 52(2):1675–1701.

Tang, J., Deng, C., and Huang, G.-B. (2015). Extreme learning machine for multilayer perceptron. *IEEE transactions on neural networks and learning systems*, 27(4):809–821.

Tascikaraoglu, A. and Uzunoglu, M. (2014). A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews*, 34:243–254.

Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., and Maida, A. (2019). Deep learning in spiking neural networks. *Neural networks*, 111:47–63.

Tavanaei, A. and Maida, A. (2019). BP-STDP: Approximating backpropagation using spike timing dependent plasticity. *Neurocomputing*, 330:39–47.

Tawn, R. and Browell, J. (2022). A review of very short-term wind and solar power forecasting. *Renewable and Sustainable Energy Reviews*, 153:111758.

Team, R. C. and Worldwide, C. (2002). The R stats package. *R Foundation for Statistical Computing, Vienna, Austria: Available from: http://www.R-project.org.*

Tian, Z. (2021). A state-of-the-art review on wind power deterministic prediction. *Wind Engineering*, 45(5):1374–1392.

Torres, J. L., Garcia, A., De Blas, M., and De Francisco, A. (2005). Forecast of hourly average wind speed with ARMA models in Navarre (Spain). *Solar energy*, 79(1):65–77.

Torres, M. E., Colominas, M. A., Schlotthauer, G., and Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4144–4147. IEEE.

Torres-Carrión, P., González-González, C., Bernal-Bravo, C., and Infante-Moro, A. (2018). Gesture-based children computer interaction for inclusive education: A systematic literature review. In *International Conference on Technology Trends*, pages 133–147. Springer.

Troen, I. and Lundtang Petersen, E. (1989). European wind atlas.

Vargas, S. A., Esteves, G. R. T., Maçaira, P. M., Bastos, B. Q., Oliveira, F. L. C., and Souza, R. C. (2019). Wind power generation: A review and a research agenda. *Journal of Cleaner Production*, 218:850–870.

Voelker, A. R. and Eliasmith, C. (2020). Programming neuromorphics using the Neural Engineering Framework. *Handbook of Neuroengineering*, pages 1–43.

Von Krannichfeldt, L., Wang, Y., Zufferey, T., and Hug, G. (2021). Online ensemble approach for probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy.*

Wan, C., Lin, J., Wang, J., Song, Y., and Dong, Z. Y. (2016). Direct quantile regression for nonparametric probabilistic forecasting of wind power generation. *IEEE Transactions on Power Systems*, 32(4):2767–2778.

Wan, C., Wang, J., Lin, J., Song, Y., and Dong, Z. Y. (2017). Nonparametric prediction intervals of wind power via linear programming. *IEEE Transactions on Power Systems*, 33(1):1074–1076.

Wan, C., Xu, Z., Pinson, P., Dong, Z. Y., and Wong, K. P. (2013a). Optimal prediction intervals of wind power generation. *IEEE Transactions on Power Systems*, 29(3):1166–1174.

Wan, C., Xu, Z., Pinson, P., Dong, Z. Y., and Wong, K. P. (2013b). Probabilistic forecasting of wind power generation using extreme learning machine. *IEEE Transactions on Power Systems*, 29(3):1033–1044.

Wang, C., Zhang, H., and Ma, P. (2020a). Wind power forecasting based on singular spectrum analysis and a new hybrid Laguerre neural network. *Applied Energy*, 259:114139.

Wang, G., Jia, R., Liu, J., and Zhang, H. (2020b). A hybrid wind power forecasting approach based on Bayesian model averaging and ensemble learning. *Renewable Energy*, 145:2426–2434.

Wang, H., Lei, Z., Liu, Y., Peng, J., and Liu, J. (2019a). Echo state network based ensemble approach for wind power forecasting. *Energy Conversion and Management*, 201:112188.

Wang, H., Xue, W., Liu, Y., Peng, J., and Jiang, H. (2020c). Probabilistic wind power forecasting based on spiking neural network. *Energy*, 196:117072.

Wang, H.-z., Li, G.-q., Wang, G.-b., Peng, J.-c., Jiang, H., and Liu, Y.-t. (2017a). Deep learning based ensemble approach for probabilistic wind power forecasting. *Applied energy*, 188:56–70.

Wang, J. and Hu, J. (2015). A robust combination approach for short-term wind speed forecasting and analysis–Combination of the ARIMA (Autoregressive Inte-

grated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model. *Energy*, 93:41–56.

Wang, J., Niu, T., Lu, H., Yang, W., and Du, P. (2019b). A novel framework of reservoir computing for deterministic and probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy*, 11(1):337–349.

Wang, J., Song, Y., Liu, F., and Hou, R. (2016a). Analysis and application of forecasting models in wind power integration: A review of multi-step-ahead wind speed forecasting models. *Renewable and Sustainable Energy Reviews*, 60:960–981.

Wang, K., Qi, X., Liu, H., and Song, J. (2018). Deep belief network based k-means cluster approach for short-term wind power forecasting. *Energy*, 165:840–852.

Wang, L., Tao, R., Hu, H., and Zeng, Y.-R. (2021a). Effective wind power prediction using novel deep learning network: Stacked independently recurrent autoencoder. *Renewable Energy*, 164:642–655.

Wang, Q., Martinez-Anido, C. B., Wu, H., Florita, A. R., and Hodge, B.-M. (2016b). Quantifying the economic and grid reliability impacts of improved wind power forecasting. *IEEE Transactions on sustainable energy*, 7(4):1525–1537.

Wang, Q., Wu, H., Florita, A. R., Martinez-Anido, C. B., and Hodge, B.-M. (2016c). The value of improved wind power forecasting: Grid flexibility quantification, ramp capability analysis, and impacts of electricity market operation timescales. *Applied Energy*, 184:696–713.

Wang, R., Li, C., Fu, W., and Tang, G. (2019c). Deep learning method based on gated recurrent unit and variational mode decomposition for short-term wind power interval prediction. *IEEE transactions on neural networks and learning systems*, 31(10):3814–3827.

Wang, S.-X., Li, M., Zhao, L., and Jin, C. (2019d). Short-term wind power prediction based on improved small-world neural network. *Neural Computing and Applications*, 31(7):3173–3185.

Wang, X., Han, Y., Leung, V. C., Niyato, D., Yan, X., and Chen, X. (2020d). Convergence of edge computing and deep learning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(2):869–904.

Wang, Y., Hu, Q., Meng, D., and Zhu, P. (2017b). Deterministic and probabilistic wind power forecasting using a variational Bayesian-based adaptive robust multi-kernel regression model. *Applied energy*, 208:1097–1112.

Wang, Y. and Markert, R. (2016). Filter bank property of variational mode decomposition and its applications. *Signal Processing*, 120:509–521.

Wang, Y. and Wu, L. (2016). On practical challenges of decomposition-based hybrid forecasting algorithms for wind speed and solar irradiation. *Energy*, 112:208–220.

Wang, Y., Yu, Y., Cao, S., Zhang, X., and Gao, S. (2020e). A review of applications of artificial intelligent algorithms in wind farms. *Artificial Intelligence Review*, 53(5):3447–3500.

Wang, Y., Zou, R., Liu, F., Zhang, L., and Liu, Q. (2021b). A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304:117766.

Wen, H., Ma, J., Gu, J., Yuan, L., and Jin, Z. (2022). Sparse variational gaussian process based day-ahead probabilistic wind power forecasting. *IEEE Transactions on Sustainable Energy*.

Wilczak, J. M., Olson, J. B., Djalalova, I., Bianco, L., Berg, L. K., Shaw, W. J., Coulter, R. L., Eckman, R. M., Freedman, J., Finley, C., et al. (2019). Data assimilation impact of in situ and remote sensing meteorological observations on wind power forecasts during the first W ind F orecast I mprovement P roject (WFIP). *Wind Energy*, 22(7):932–944.

Willmott, C. J. (1981). On the validation of models. *Physical geography*, 2(2):184–194.

Willmott, C. J., Robeson, S. M., and Matsuura, K. (2012). A refined index of model performance. *International Journal of climatology*, 32(13):2088–2094.

Winkler, R. L. (1972). A decision-theoretic approach to interval estimation. *Journal of the American Statistical Association*, 67(337):187–191.

Wit, E., Heuvel, E. v. d., and Romeijn, J.-W. (2012). 'All models are wrong...': an introduction to model uncertainty. *Statistica Neerlandica*, 66(3):217–236.

Wu, Q., Guan, F., Lv, C., and Huang, Y. (2021). Ultra-short-term multi-step wind power forecasting based on CNN-LSTM. *IET Renewable Power Generation*, 15(5):1019–1029.

Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018a). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331.

Wu, Y.-K., Su, P.-E., Wu, T.-Y., Hong, J.-S., and Hassan, M. Y. (2018b). Probabilistic wind-power forecasting using weather ensemble models. *IEEE Transactions on Industry Applications*, 54(6):5609–5620.

Wu, Z. and Huang, N. E. (2009). Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in adaptive data analysis*, 1(01):1–41.

Wu, Z., Xia, X., Xiao, L., and Liu, Y. (2020). Combined model with secondary decomposition-model selection and sample selection for multi-step wind power forecasting. *Applied Energy*, 261:114345.

Würth, I., Valldecabres, L., Simon, E., Möhrlen, C., Uzunoğlu, B., Gilbert, C., Giebel, G., Schlipf, D., and Kaifel, A. (2019). Minute-scale forecasting of wind power—results from the collaborative workshop of iea wind task 32 and 36. *Energies*, 12(4):712.

Xiang, L., Liu, J., Yang, X., Hu, A., and Su, H. (2021). Ultra-short term wind power prediction applying a novel model named SATCN-LSTM. *Energy Conversion and Management*, page 115036.

Xie, W., Zhang, P., Chen, R., and Zhou, Z. (2018). A nonparametric bayesian framework for short-term wind power probabilistic forecast. *IEEE Transactions on Power Systems*, 34(1):371–379.

Xiong, B., Lou, L., Meng, X., Wang, X., Ma, H., and Wang, Z. (2022). Short-term wind power forecasting based on attention mechanism and deep learning. *Electric Power Systems Research*, 206:107776.

Xu, Q., He, D., Zhang, N., Kang, C., Xia, Q., Bai, J., and Huang, J. (2015). A short-term wind power forecasting approach with adjustment of numerical weather prediction input by data mining. *IEEE Transactions on sustainable energy*, 6(4):1283–1291.

Yan, J., Li, K., Bai, E., Yang, Z., and Foley, A. (2016). Time series wind power forecasting based on variant Gaussian Process and TLBO. *Neurocomputing*, 189:135–144.

Yan, J., Li, K., Bai, E.-W., Deng, J., and Foley, A. M. (2015a). Hybrid probabilistic wind power forecasting using temporally local Gaussian process. *IEEE Transactions on Sustainable Energy*, 7(1):87–95.

Yan, J., Liu, Y., Han, S., and Qiu, M. (2013). Wind power grouping forecasts and its uncertainty analysis using optimized relevance vector machine. *Renewable and sustainable energy reviews*, 27:613–621.

Yan, J., Liu, Y., Han, S., Wang, Y., and Feng, S. (2015b). Reviews on uncertainty analysis of wind power forecasting. *Renewable and Sustainable Energy Reviews*, 52:1322–1330.

Yan, J., Möhrlen, C., Göçmen, T., Kelly, M., Wessel, A., and Giebel, G. (2022). Uncovering wind power forecasting uncertainty sources and their propagation through the whole modelling chain. *Renewable and Sustainable Energy Reviews*, 165:112519.

Yan, J. and Ouyang, T. (2019). Advanced wind power prediction based on data-driven error correction. *Energy conversion and management*, 180:302–311.

Yan, J., Zhang, H., Liu, Y., Han, S., and Li, L. (2019). Uncertainty estimation for wind energy conversion by probabilistic wind turbine power curve modelling. *Applied energy*, 239:1356–1370.

Yang, B., Zhong, L., Wang, J., Shu, H., Zhang, X., Yu, T., and Sun, L. (2021a). State-of-the-art one-stop handbook on wind forecasting technologies: An overview of classifications, methodologies, and analysis. *Journal of Cleaner Production*, 283:124628.

Yang, D., Li, W., Yagli, G. M., and Srinivasan, D. (2021b). Operational solar forecasting for grid integration: Standards, challenges, and outlook. *Solar Energy*, 224:930–937.

Yang, L., He, M., Zhang, J., and Vittal, V. (2015). Support-vector-machine-enhanced Markov model for short-term wind power forecast. *IEEE Transactions on Sustainable Energy*, 6(3):791–799.

Yang, M., Shi, C., and Liu, H. (2021c). Day-ahead wind power forecasting based on the clustering of equivalent power curves. *Energy*, 218:119515.

Yang, Y., Deng, J., and Wu, C. (2009). Analysis of mode mixing phenomenon in the empirical mode decomposition method. In *2009 Second International Symposium on Information Science and Engineering*, pages 553–556. IEEE.

Ye, L., Zhao, Y., Zeng, C., and Zhang, C. (2017). Short-term wind power prediction based on spatial model. *Renewable Energy*, 101:1067–1074.

Yeh, J.-R., Shieh, J.-S., and Huang, N. E. (2010). Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Advances in adaptive data analysis*, 2(02):135–156.

Yesilbudak, M., Sagiroglu, S., and Colak, I. (2017). A novel implementation of kNN classifier based on multi-tupled meteorological input data for wind power prediction. *Energy Conversion and Management*, 135:434–444.

Yildiz, C., Acikgoz, H., Korkmaz, D., and Budak, U. (2021). An improved residual-based convolutional neural network for very short-term wind power forecasting. *Energy Conversion and Management*, 228:113731.

Yin, H., Ou, Z., Huang, S., and Meng, A. (2019). A cascaded deep learning wind

power prediction approach based on a two-layer of mode decomposition. *Energy*, 189:116316.

Yin, H., Ou, Z., Zhu, Z., Xu, X., Fan, J., and Meng, A. (2021). A novel asexual-reproduction evolutionary neural network for wind power prediction based on generative adversarial networks. *Energy Conversion and Management*, 247:114714.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing.

Yousuf, M. U., Al-Bahadly, I., and Avci, E. (2019). Current perspective on the accuracy of deterministic wind speed and power forecasting. *IEEE Access*, 7:159547–159564.

Yu, R., Gao, J., Yu, M., Lu, W., Xu, T., Zhao, M., Zhang, J., Zhang, R., and Zhang, Z. (2019a). LSTM-EFG for wind power forecasting based on sequential correlation features. *Future Generation Computer Systems*, 93:33–42.

Yu, R., Liu, Z., Li, X., Lu, W., Ma, D., Yu, M., Wang, J., and Li, B. (2019b). Scene learning: Deep convolutional networks for wind power prediction by embedding turbines into grid space. *Applied energy*, 238:249–257.

Yu, Y., Han, X., Yang, M., and Yang, J. (2020). Probabilistic prediction of regional wind power based on spatiotemporal quantile regression. *IEEE Transactions on Industry Applications*, 56(6):6117–6127.

Yuan, R., Wang, B., Mao, Z., and Watada, J. (2021). Multi-objective wind power scenario forecasting based on PG-GAN. *Energy*, 226:120379.

Yuan, X., Chen, C., Jiang, M., and Yuan, Y. (2019). Prediction interval of wind power using parameter optimized Beta distribution based LSTM model. *Applied Soft Computing*, 82:105550.

Yuan, X., Chen, C., Yuan, Y., Huang, Y., and Tan, Q. (2015). Short-term wind power prediction based on LSSVM–GSA model. *Energy Conversion and Management*, 101:393–401.

Yuan, X., Tan, Q., Lei, X., Yuan, Y., and Wu, X. (2017). Wind power prediction using hybrid autoregressive fractionally integrated moving average and least square support vector machine. *Energy*, 129:122–137.

Yunus, K., Thiringer, T., and Chen, P. (2015). ARIMA-based frequency-decomposed modeling of wind speed time series. *IEEE Transactions on Power Systems*, 31(4):2546–2556.

Zack, J., Möhrlen, C., and Draxl, C. (2019). IEA Wind Task 36 Session Topic 4: Request for Feedback on Version 1 of the Recommended Practices for Forecast Solution Selection. In *Proceedings of the 18th International Workshop on Large-Scale Integration of Wind Power into Power Systems (Dublin, Ireland). Energynautics GmbH, Darmstadt Germany.*

Zameer, A., Arshad, J., Khan, A., and Raja, M. A. Z. (2017). Intelligent and robust prediction of short term wind power using genetic programming based ensemble of neural networks. *Energy conversion and management*, 134:361–372.

Zendehboudi, A., Baseer, M. A., and Saidur, R. (2018). Application of support vector machine models for forecasting solar and wind energy resources: A review. *Journal of cleaner production*, 199:272–285.

Zhang, F., Li, P.-C., Gao, L., Liu, Y.-Q., and Ren, X.-Y. (2021a). Application of autoregressive dynamic adaptive (ARDA) model in real-time wind power forecasting. *Renewable Energy*, 169:129–143.

Zhang, G., Liu, H., Zhang, J., Yan, Y., Zhang, L., Wu, C., Hua, X., and Wang, Y. (2019a). Wind power prediction based on variational mode decomposition multi-frequency combinations. *Journal of Modern Power Systems and Clean Energy*, 7(2):281–288.

Zhang, G., Wu, Y., Wong, K. P., Xu, Z., Dong, Z. Y., and Iu, H. H.-C. (2014a). An advanced approach for construction of optimal wind power prediction intervals. *IEEE transactions on power systems*, 30(5):2706–2715.

Zhang, H., Liu, Y., Yan, J., Han, S., Li, L., and Long, Q. (2020a). Improved deep

mixture density network for regional wind power probabilistic forecasting. *IEEE Transactions on Power Systems*, 35(4):2549–2560.

Zhang, H., Yan, J., Liu, Y., Gao, Y., Han, S., and Li, L. (2021b). Multi-source and temporal attention network for probabilistic wind power prediction. *IEEE Transactions on Sustainable Energy*.

Zhang, J., Liu, D., Li, Z., Han, X., Liu, H., Dong, C., Wang, J., Liu, C., and Xia, Y. (2021c). Power prediction of a wind farm cluster based on spatiotemporal correlations. *Applied Energy*, 302:117568.

Zhang, J., Meng, H., Gu, B., and Li, P. (2020b). Research on short-term wind power combined forecasting and its Gaussian cloud uncertainty to support the integration of renewables and EVs. *Renewable Energy*, 153:884–899.

Zhang, J., Yan, J., Infield, D., Liu, Y., and Lien, F.-s. (2019b). Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and Gaussian mixture model. *Applied Energy*, 241:229–244.

Zhang, W., Lin, Z., and Liu, X. (2021d). Short-term offshore wind power forecasting-A hybrid model based on Discrete Wavelet Transform (DWT), Seasonal Autoregressive Integrated Moving Average (SARIMA), and deep-learning-based Long Short-Term Memory (LSTM). *Renewable Energy*.

Zhang, Y., Le, J., Liao, X., Zheng, F., and Li, Y. (2019c). A novel combination forecasting model for wind power integrating least square support vector machine, deep belief network, singular spectrum analysis and locality-sensitive hashing. *Energy*, 168:558–572.

Zhang, Y., Li, Y., and Zhang, G. (2020c). Short-term wind power forecasting approach based on Seq2Seq model using NWP data. *Energy*, page 118371.

Zhang, Y., Liu, K., Qin, L., and An, X. (2016). Deterministic and probabilistic interval prediction for short-term wind power generation based on variational mode decomposition and machine learning methods. *Energy Conversion and Management*, 112:208–219.

Zhang, Y. and Wang, J. (2016). K-nearest neighbors and a kernel density estimator for GEFCom2014 probabilistic wind power forecasting. *International Journal of forecasting*, 32(3):1074–1080.

Zhang, Y., Wang, J., and Wang, X. (2014b). Review on probabilistic forecasting of wind power generation. *Renewable and Sustainable Energy Reviews*, 32:255–270.

Zhao, C., Wan, C., and Song, Y. (2019). An adaptive bilevel programming model for nonparametric prediction intervals of wind power generation. *IEEE Transactions on Power Systems*, 35(1):424–439.

Zhao, C., Wan, C., and Song, Y. (2021). Operating reserve quantification using prediction intervals of wind power: an integrated probabilistic forecasting and decision methodology. *IEEE Transactions on Power Systems*.

Zhao, Y., Ye, L., Li, Z., Song, X., Lang, Y., and Su, J. (2016). A novel bidirectional mechanism based on time series model for wind power forecasting. *Applied Energy*, 177:793–803.

Zhao, Y., Ye, L., Pinson, P., Tang, Y., and Lu, P. (2018). Correlation-constrained and sparsity-controlled vector autoregressive model for spatio-temporal wind power forecasting. *IEEE Transactions on Power Systems*, 33(5):5029–5040.

Zhou, B., Duan, H., Wu, Q., Wang, H., Or, S. W., Chan, K. W., and Meng, Y. (2021). Short-term prediction of wind power and its ramp events based on semi-supervised generative adversarial network. *International Journal of Electrical Power & Energy Systems*, 125:106411.

Zhou, B., Ma, X., Luo, Y., and Yang, D. (2019). Wind power prediction based on LSTM networks and nonparametric kernel density estimation. *IEEE Access*, 7:165279–165292.

Ziel, F., Croonenbroeck, C., and Ambach, D. (2016). Forecasting wind power–modeling periodic and non-linear effects under conditional heteroscedasticity. *Applied Energy*, 177:285–297.

Zjavka, L. and Mišák, S. (2018). Direct wind power forecasting using a polynomial decomposition of the general differential equation. *IEEE Transactions on Sustainable Energy*, 9(4):1529–1539.

Zou, W., Li, C., and Chen, P. (2019). An inter type-2 FCR algorithm based T–S fuzzy model for short-term wind power interval prediction. *IEEE Transactions on Industrial Informatics*, 15(9):4934–4943.

Appendices

Appendix A

Literature review

This appendix contains tables with all reviewed papers ordered chronologically. They contain information about how the forecasts have been modelled, the type of output (whether they provide point, probabilistic estimates, or both), the region/country where the data are from, and the resolution of these data.

Table A.1: Reviewed WPF modelling publications published before 2016.

Reference	Modelling	Output	Data	Resolution
Nielsen et al. (2006)	QR	Prob	Denmark	15-min
Sideratos and Hatziargyriou (2007)	ANN	Point	Denmark	1-h
Jursa and Rohrig (2008)	ANN, Nearest Neighbors search	Point	Germany	1-h
Bessa et al. (2009)	ANN	Point	Iberian Peninsula	10-min
Ramirez-Rosado et al. (2009)	ANN, AR	Point	Portugal	1-h
Catalão et al. (2010)	ANFIS	Point	Portugal	15-min
Hong et al. (2010)	ANN	Point	N/A	30-min, 1-h
Kusiak and Zhang (2010)	ANN, SVM, BT, RF, kNN	Point	N/A	10-s
Liu et al. (2010)	ARIMA	Point	China	1-h
Pinson and Kariniotakis (2010)	Adapted resampling, fuzzy inference	Prob	Denmark	1-h
Amjady et al. (2011a)	ANN	Point	Ireland Sotavento	1-h
Amjady et al. (2011b)	ANN	Point	Canada USA	1-h
An et al. (2012)	Grey model, Lyapunov exponent	Point	China	10-min
Blonbou (2011)	ANN	Point	Guadeloupe	5-, 10-min
Catalão et al. (2011)	ANN	Point	Portugal	15-min
Pousinho et al. (2011)	ANFIS	Point	Portugal	15-min
Bessa et al. (2012b)	KDE	Prob	NREL	1-h
Bessa et al. (2012c)	KDE	Prob	NREL	1-h
Bhaskar and Singh (2012)	ANN, WNN	Point	NREL	1-h
Pinson and Madsen (2012)	AR, Markov model	Both	Denmark	10-min
Sideratos and Hatziargyriou (2012)	ANN	Prob	Denmark Greece	1-h
Chen et al. (2013)	Gaussian Process, WP curve	Point	China	1-h
Khosravi and Nahavandi (2013)	ANN, LUBE	Prob	Australia	15-min
Khosravi et al. (2013)	ANN, LUBE	Prob	Australia	5-min
Kou et al. (2013)	Gaussian Process	Both	China	15-min
Lee and Baldick (2013)	ANN, Gaussian Process	Point	GEFCom2012	1-h
Peng et al. (2013)	ANN, physical-statistical method	Point	China	10-min
Quan et al. (2013)	ANN, LUBE	Prob	Australia	1-h
Shi et al. (2013)	ANN, LS-SVM	Point	China	15-min
Stathopoulos et al. (2013)	Numerical+statistical models	Both	Greece	1-h
Wan et al. (2013a)	ELM	Prob	Australia	1-h
Wan et al. (2013b)	ELM	Prob	Australia	1-h
Haque et al. (2014)	ANN, QR	Both	USA	10-min
Zhang et al. (2014a)	ELM	Prob	Australia NREL	5-min 1-h
Alessandrini et al. (2015)	Analog Ensemble	Prob	Italy	1-h
Buhan and Çadırcı (2015)	SVM, ANN	Point	Turkey	1-h
Carpinone et al. (2015)	Markov chain model	Both	USA	10-min
Chitsaz et al. (2015)	WNN	Point	Canada	1-h
Dowell and Pinson (2015)	VAR	Both	Australia	5-min
Han et al. (2015)	ANN	Point	NREL	1-h
Kavousi-Fard et al. (2015)	ANN, LUBE	Prob	Australia	15-min
Li et al. (2015)	ANN	Point	NREL	1-h
Osório et al. (2015)	ANFIS	Point	Portugal	15-min
Ozkan and Karagoz (2015)	Dynamic clustering, linear regression	Point	Turkey	1-h
Xu et al. (2015)	ANN	Point	GEFCom2012	1-h
Yan et al. (2015a)	Gaussian Process	Both	Ireland USA	1-h
Yang et al. (2015)	SVM, Markov model	Both	N/A	10-min
Yuan et al. (2015)	LS-SVM	Point	N/A	1-h

Table A.2: Reviewed WPF modelling publications published in 2016-2017.

Reference	Modelling	Type	Output	Resolution
Abdoos (2016)	ELM	Point	NREL Sotavento	10-min
Aghajani et al. (2016)	ANN	Point	Canada	1-h
Azimi et al. (2016)	ANN	Point	NREL	1-h
Dong et al. (2016)	ANN	Point	China	15-min
Gallego-Castillo et al. (2016)	QR, RKHS	Prob	GEFCom2012	1-h
Heinermann and Kramer (2016)	SVM, DT	Point	NREL	10-min
Landry et al. (2016)	GBM	Prob	GEFCom2014	1-h
Li et al. (2016)	ELM	Point	China	15-min
Nagy et al. (2016)	GBM, QRF	Prob	GEFCom2014	1-h
Renani et al. (2016)	ANFIS, kNN, RF, SVM, ANN, M5Rules	Point	Iran	5-min
Saleh et al. (2016)	Neuro-fuzzy model	Point	USA	1-h
Wan et al. (2016)	ELM, QR	Prob	Denmark	10-min
Yan et al. (2016)	Gaussian Process	Point	Ireland	15-min
Zhang et al. (2016)	LS-SVM, ESN, ELM, QR	Both	China	10-min
Zhang and Wang (2016)	kNN, KDE	Both	GEFCom2014	1-h
Zhao et al. (2016)	ELM	Point	NREL	1-h
Ziel et al. (2016)	ARMA, GARCH	Both	Germany	10-min
Cavalcante et al. (2017)	VAR	Point	N/A	1-h
Chang et al. (2017)	ANN	Point	Taiwan	10-min
Dong et al. (2017)	Neuro-fuzzy system	Point	China	15-min
Iversen et al. (2017)	Stochastic DE	Both	Denmark	1-h
Jiang et al. (2017)	ARMA	Point	China	1-h
Karakuş et al. (2017)	AR	Point	GEFCom2012	1-h
Lahouar and Slama (2017)	RF	Both	Tunisia	10-min
Liu et al. (2017)	ANFIS, ANN, LS-SVM	Point	China	15-min
Qureshi et al. (2017)	DNN	Point	Europe	1-h
Safari et al. (2017)	LS-SVM	Point	Canada Sotavento	10-min
Shi et al. (2017)	RNN, LUBE	Prob	Canada	1-h
Wang et al. (2017a)	Deep CNN	Prob	China	5-min
Wang et al. (2017b)	Kernel regression	Both	China	10-min
Ye et al. (2017)	CFD, Power curve	Point	N/A	10-min
Yesilbudak et al. (2017)	kNN	Point	Denmark	10-min
Yuan et al. (2017)	ARFIMA, LS-SVM	Point	N/A	10-min
Zameer et al. (2017)	ANN	Point	Europe	1-h

Table A.3: Reviewed WPF modelling publications published in 2018-2019.

Reference	Modelling	Output	Data	Resolution
Afshari-Igder et al. (2018)	ELM	Prob	Canada	1-h
Baxevani and Lenzi (2018)	Gaussian Process	Both	Denmark	15-min
He and Li (2018)	ANN, QR, KDE	Both	Canada	1-h
Khorramdel et al. (2018)	ELM, KDE	Prob	Canada Spain	30-, 60-min
Leng et al. (2018)	ANN	Point	Canada NREL	1-h
Lin et al. (2018b)	KDE, Bayesian learning, Beta distribution	Both	GEFCom2014	1-h
Mahmoud et al. (2018)	ELM	Prob	Australia	30-min
Naik et al. (2018a)	Kernel regression	Prob	USA	10-, 30-min, 1-h
Naik et al. (2018b)	ANN	Point	USA Sotavento	10-, 30-min, 1-h, 3-h
Naik et al. (2018c)	Kernel regression	Point	USA	10-, 30-min, 1-h, 3-h
Sharifian et al. (2018)	Neuro-fuzzy system	Point	Canada	1-h
Sun et al. (2018)	WNN	Point	China	1-h
Wang et al. (2018)	DBN	Point	Sotavento	10-min
Wu et al. (2018b)	ANN, LUBE	Prob	Taiwan	1-h
Xie et al. (2018)	Markov model, AR	Prob	USA	1-h
Zhao et al. (2018)	VAR	Point	Denmark	15-min
Zjavka and Mišák (2018)	Differential polynomial NN	Point	Czechia	30-min
Chen et al. (2019a)	GARCH	Point	China	5-min
Çevik et al. (2019)	ANFIS, ANN, SVM	Point	Sotavento	1-h
Demolli et al. (2019)	LASSO, kNN, xGBoost, RF, SVM	Point	Turkey	10-min
Ding et al. (2019)	GRU	Point	China	15-min
Du et al. (2019)	WNN	Point	Sotavento	10-min
Han et al. (2019a)	LSTM	Point	ELIA (Belgium)	15-min
Han et al. (2019b)	LSTM	Point	ELIA (Belgium) NREL	15-min 15-min
Hao et al. (2019)	ANN	Point	China	15-min
Hao and Tian (2019)	ELM	Point	Canada Sotavento	1-h 10-min
Hong and Rioflorido (2019)	Deep CNN	Point	Taiwan	1-h
Hong et al. (2019)	LS-SVM	Point	ELIA (Belgium) Canada Australia China	15-min 10-min 10-min 1-h
Ju et al. (2019)	CNN, GBM	Point	China	5-min
Korprasertsak and Leephakpreeda (2019)	ARMA, ANN, Grey model	Both	Sotavento	1-h
Messner and Pinson (2019)	VAR	Point	France Denmark	10-min 15-min
Mishra and Dash (2019)	Legendre NN	Point	Sotavento USA	10-, 30-, 60-min
Prósper et al. (2019)	WRF	Point	Spain	1-h
Qureshi and Khan (2019)	DNN	Point	Europe	1-h
Sun et al. (2019)	ANN, RF, SVM, GBM, GMM	Point	USA	1-h
Wang et al. (2019a)	ESN	Point	ELIA (Belgium) China	15-min 15-min
Wang et al. (2019b)	ESN	Both	Sotavento	10-min
Wang et al. (2019c)	GRU	Prob	NREL	10-min
Wang et al. (2019d)	ANN	Point	China	15-min
Yan and Ouyang (2019)	ANN, SVM, BT, RF	Point	China	15-min
Yin et al. (2019)	CNN, LSTM	Point	Sotavento	1-h
Yu et al. (2019a)	LSTM	Point	NREL	10-min
Yu et al. (2019b)	Deep CNN	Point	NREL	10-min
Yuan et al. (2019)	LSTM	Prob	N/A	10-min
Zhang et al. (2019a)	ARMA, ANN, LS-SVM	Point	China	10-min
Zhang et al. (2019b)	LSTM, GMM	Point	China	15-min
Zhang et al. (2019c)	LS-SVM, DBN	Point	China	10-min
Zhao et al. (2019)	ELM, QR	Prob	NREL	15-min
Zou et al. (2019)	LUBE, IT-2 FCR	Prob	NREL	10-min

Table A.4: Reviewed WPF modelling publications published in 2020.

Reference	Modelling	Output	Data	Resolution
Abedinia et al. (2020a)	CNN	Point	Canada	1-h
			Sotavento	1-h
Abedinia et al. (2020b)	ANN	Point	Canada	1-h
			Sotavento	1-h
Acikgoz et al. (2020)	ELM	Point	Turkey	1-h
Ahmadpour and Farkoush (2020)	Gaussian Process	Both	GEFCom2012	1-h
Aly (2020)	ANN, WNN	Point	Canada	1-h
Chen and Liu (2020)	ANN, ELM, RNN, LSTM	Point	Canada	1-h
Devi et al. (2020)	LSTM	Point	India	15-min
Gilbert et al. (2020)	Large eddy simulation	Point	Denmark	10-min
Hu et al. (2020a)	ESN	Both	China	15-min
Hu et al. (2020b)	QR, RKHS	Both	China	15-min
			Sotavento	1-h
Li et al. (2020a)	LSTM, LUBE	Prob	NREL	10-min
Li et al. (2020b)	ANN	Point	China	15-min
Li et al. (2020c)	WRF, power curve	Point	Tunisia	10-min
Li et al. (2020d)	SVM	Point	France	1-h
Li et al. (2020e)	SVM	Point	France	1-h
Lin and Liu (2020)	DNN	Point	UK	1-s
Lin et al. (2020)	DNN	Point	UK	1-s
Liu and Duan (2020)	ELM, KDE	Both	China	10-, 30-, 60-min
Lu et al. (2020a)	Grey box model	Point	ELIA (Belgium)	3-h
Lu et al. (2020b)	SVM	Point	N/A	15-min
Nazaré et al. (2020)	ANN	Point	Portugal	15-min
Nielson et al. (2020)	ANN	Point	USA	1-h
Niu et al. (2020)	GRU	Point	NREL	1-h
Ouyang et al. (2020)	ANN, SVM, RF	Point	USA	15-min
Qian and Wang (2020)	Grey model	Point	China	N/A
Shahid et al. (2020a)	Ensemble ANN	Point	Europe	1-h
Shahid et al. (2020b)	LSTM	Point	Europe	1-h
Sun et al. (2020a)	ANN, RF, SVM, GBM	Both	USA	1-h
Sun et al. (2020b)	LSTM	Both	China	15-min
Wang et al. (2020a)	Laguerre NN	Point	China	1-h
Wang et al. (2020b)	ANN, SVM	Point	China	15-min
Wang et al. (2020c)	SNN, LUBE	Prob	Belgium	15-min
			China	15-min
Wu et al. (2020)	ANN-based models	Point	ELIA (Belgium)	15-min
Yu et al. (2020)	CNN, LSTM, QR	Prob	China	15-min
Zhang et al. (2020a)	ANN	Both	GEFCom2012	1-h
Zhang et al. (2020c)	Encoder-decoder, LSTM	Point	Sotavento	10-min

Table A.5: Reviewed WPF modelling publications published in 2021.

Reference	Modelling	Output	Data	Resolution
Alshelahi et al. (2021)	Stochastic DE, power curve	Both	N/A	10-min
De Caro et al. (2021)	RF, Lazy learning, persistence	Point	Australia Italy	1-h
Dong et al. (2021a)	Bernstein polynomial, mixture of Gaussians	Point	China	15-min
Dong et al. (2021b)	Bernstein polynomial, ANN	Point	China	15-min
Duan et al. (2021)	LSTM	Point	China	5-, 10-min
Gendeel et al. (2021)	LS-SVM, LUBE	Both	China	15-min
Ghoushchi et al. (2021)	WNN	Point	Iran	10-min
Gu et al. (2021)	LSTM, KDE	Both	China	15-min
He et al. (2021)	SVM, QR, KDE	Both	Canada China	1-h 10-min
Hossain et al. (2021)	CNN, GRU	Point	Australia	5-min
Liu et al. (2021c)	CNN, GRU	Point	N/A	10-min
Lv et al. (2021)	Sparse ML models, KDE	Both	NREL	1-h
Meka et al. (2021)	TCN	Point	N/A	10-min
Putz et al. (2021)	DNN	Point	Europe	15-, 30-, 60-min
Rayi et al. (2021)	ELM	Point	USA Sotavento	10-, 60-min
Shahid et al. (2021)	LSTM	Point	Europe	1-h
Sommer et al. (2021)	AR	Point	Denmark	15-min
Von Krannichfeldt et al. (2021)	SVM, RF, GBRT, ERT	Prob	ELIA (Belgium)	15-min
Wang et al. (2021a)	Autoencoder	Point	ELIA (Belgium)	15-min
Wu et al. (2021)	CNN, LSTM	Point	China	10-min
Xiang et al. (2021)	TCN, LSTM	Point	NREL	10-min
Yang et al. (2021c)	Fuzzy C-means, power curve	Point	China	15-min
Yildiz et al. (2021)	CNN	Point	Turkey	1-h
Yin et al. (2021)	GAN	Point	Sotavento	10-min
Yuan et al. (2021)	GAN	Prob	NREL	5-min
Zhang et al. (2021a)	AR	Point	China	1-s
Zhang et al. (2021b)	LSTM	Both	China	1-h
Zhang et al. (2021c)	CNN, LSTM	Point	China	10-min
Zhang et al. (2021d)	SARIMA, LSTM	Point	UK	1-h
Zhou et al. (2021)	GAN	Point	Belgium China	15-min 60-min

Table A.6: Reviewed WPF modelling publications published in 2022.

Reference	Modelling	Output	Data	Resolution
Ahmad and Zhang (2022)	LSTM	Point	ELIA (Belgium)	15-min
Dong et al. (2022)	KDE	Prob	China	1-h
Duan et al. (2022)	LSTM, DBN	Point	China	10-min
Hu et al. (2022)	TCN, QR	Prob	China Sotavento	15-min 1-h
Khazaei et al. (2022)	ANN	Point	Sotavento	10-min
Meng et al. (2022)	GRU	Point	Spain	1-h
Sasser et al. (2022)	DT	Point	USA	10-min
Wen et al. (2022)	Gaussian Process	Prob	GEFCom2014	1-h
Xiong et al. (2022)	CNN, LSTM	Point	China	3-min

Appendix B

Packages and simulations

In Chapter 2, all the simulations have been run using R. The ARMA(1,1) series are obtained using the `arima.sim()` function available in the package stats (Team and Worldwide, 2002). As this function does not support seasonality, the seasonal ARIMA processes are modelled by the function `sim_sarima()` from the package sarima (Boshnakov and Halliday, 2020). The function `Arima()` from the package forecast (Hyndman et al., 2020) is used to fit both true and misspecified models. The predictions are finally estimated with the function `predict()`, also available in the package stats.

In the rest of Chapters, all the simulations have been run using Python. The VMD algorithm has been implemented using the `vmdpy` library (Carvalho et al., 2020), EMD and EEMD with the `PyEMD` library (Laszuk, 2017), `keras-tcn` (Remy, 2020) for TCNs, and Keras with Tensorflow backend for the rest of models (Chollet et al., 2015; Abadi et al., 2015). ELMs (used in section 5.2) have been implemented with the library `HP-ELM` (Akusok et al., 2015). The DBSCAN algorithm has been implemented with the library `scikit-learn` (Buitinck et al., 2013).

In Chapter 6, the library Nengo (Bekolay et al., 2014) is used to simulate neuromorphic algorithms together with the extensions NengoDL (Rasmussen, 2019) for DL and NengoLoihi to emulate the behavior of Loihi hardware.

Appendix C

Equivalence of Definition 1 and the minimizer of the KL distance

The precise meaning of this equivalence is elaborated as follows. Let $f_{\theta_M,n}(y_1,\ldots,y_n)$ and $f_n(y_1,\ldots,y_n)$ be the joint densities of $\{Y_1,\ldots,Y_n\}$ under the model M with parameter θ_M and the true model $\mathcal{M}$ respectively. We shall consider the limit of the scaled KL-divergence

$$\frac{1}{n}\mathrm{KL}(f_{\theta_M,n}\|f_n) = \frac{1}{n}\int \log\left(\frac{f_n(y_1,\ldots,y_n)}{f_{\theta_M,n}(y_1,\ldots,y_n)}\right) f_n(y_1,\ldots,y_n)\,dy_1\ldots dy_n.$$

Let us use $\boldsymbol{y}_n := (y_1,\ldots,y_n)'$ to simplify the notation. Also let $\hat{y}_k$ be the best linear prediction of y_k using $y_1,\ldots,y_{k-1}$ under the model M with parameter θ_M, and denote by ν_j the corresponding prediction error variance. Under normality of M, it holds that

$$-\frac{1}{n}\log f_{\theta_M,n}(\boldsymbol{y}_n) = \frac{1}{2n}\sum_{k=1}^{n}\left[\log(2\pi\nu_k) + \frac{(y_k-\hat{y}_k)^2}{\nu_k}\right].$$

As $k\to\infty$, it holds that $\hat{y}_k \to \hat{y}_k^{\theta_M}$, and

$$\lim_{k\to\infty}\nu_k = \nu_\infty := E_{\theta_M}\left(y_k - \hat{y}_k^{\theta_M}\right)^2.$$

In the limit of the KL divergence,

$$\lim_{n\to\infty}\frac{1}{n}\mathrm{KL}(f_{\theta_M,n}\|f_n) = \lim_{n\to\infty}\frac{1}{n}\int \log\left[f_n(\boldsymbol{y}_n)\right] f_n(\boldsymbol{y}_n)\, d\boldsymbol{y}_n - \\ - \lim_{n\to\infty}\frac{1}{n}\int \log\left[f_{\theta_M,n}(\boldsymbol{y}_n)\right] f_n(\boldsymbol{y}_n)\, d\boldsymbol{y}_n,$$

since the first term does not involve θ_M, we aim to minimize the second term, which in the limit becomes

$$-\lim_{n\to\infty}\frac{1}{n}\int \log\left[f_{\theta_M,n}(\boldsymbol{y}_n)\right] f_n(\boldsymbol{y}_n)\, d\boldsymbol{y}_n = \frac{1}{2}\log(2\pi\nu_\infty) + \frac{1}{2\nu_\infty}E_{\mathcal{M}}\left[\hat{Y}_{n+1}^{\theta_M} - Y_{n+1}\right]^2.$$

This leads to our definition of the best misspecified model.

Appendix D

Model misspecification with SARIMA models

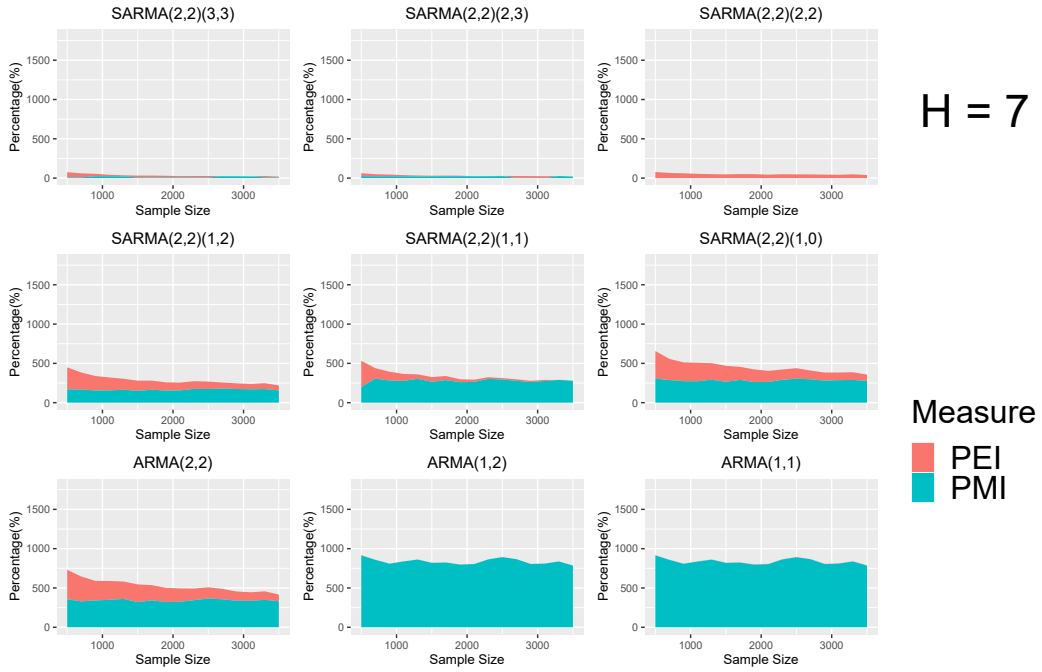

Figure D.1: Decomposition of percentage inflation in forecast errors for a true SARMA$(2,2)(2,2)_7$ process for 7-step ahead predictions. 5000 simulations have been run for every scenario.

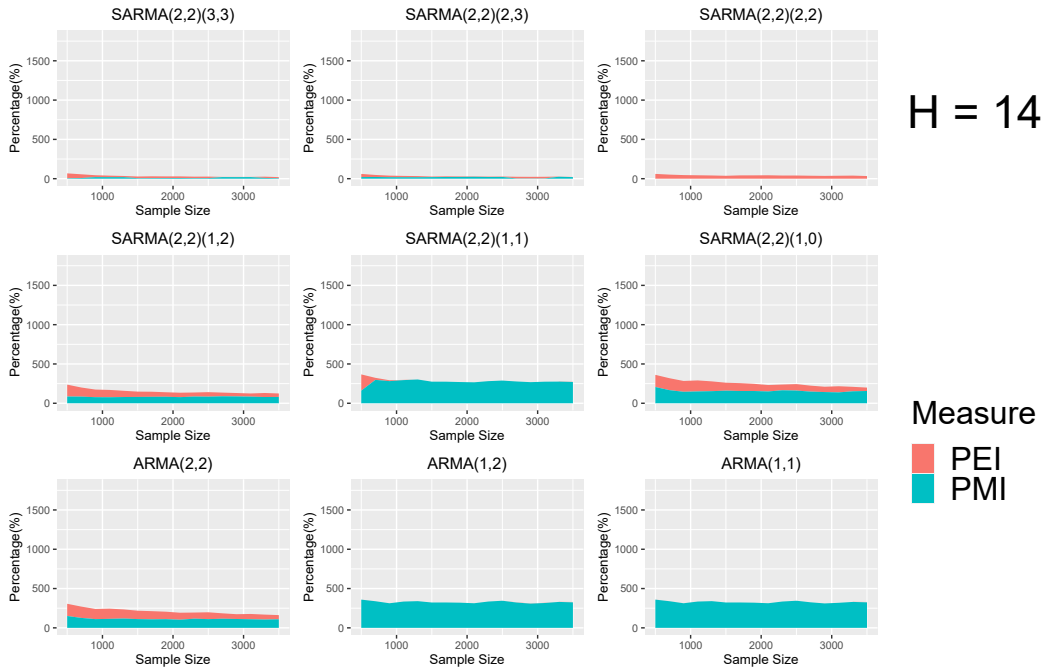

Figure D.2: Decomposition of percentage inflation in forecast errors for a true SARMA(2,2)(2,2)$_7$ process for 14-step ahead predictions. 5000 simulations have been run for every scenario.

Appendix E

List of publications

Journal publications

- González-Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2021). An overview of performance evaluation metrics for short-term statistical wind power forecasting. *Renewable and Sustainable Energy Reviews*, 138, 110515. DOI: https://doi.org/10.1016/j.rser.2020.110515

- González-Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2022). A spiking neural network based wind power forecasting model for neuromorphic devices. *Energies, in press.*

Conference publications

- González Sopeña, J. M., Maury, C., Pakrashi, V., & Ghosh, B. (2022). Turbine-level clustering for improved short-term wind power forecasting. In Journal of Physics: Conference Series (Vol. 2265, No. 2, p. 022052). IOP Publishing. DOI: https://doi.org/10.1088/1742-6596/2265/2/022052

- González Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2021). Can we improve short-term wind power forecasts using turbine-level data? A case study in Ireland. In *2021 IEEE Madrid PowerTech* (pp. 1-6). IEEE. DOI: https://doi.org/10.1109/PowerTech46648.2021.9494805

- González Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2021). Decomposition-

Based hybrid models for very short-term wind power forecasting. *Engineering Proceedings*, 5(1), 39. DOI: https://doi.org/10.3390/engproc2021005039

- González Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2020). Multi-step ahead wind power forecasting for Ireland using an ensemble of VMD-ELM models. In *2020 31st Irish Signals and Systems Conference (ISSC)* (pp. 1-5). IEEE. DOI: https://doi.org/10.1109/ISSC49989.2020.9180155

- González Sopeña, J. M., Pakrashi, V., & Ghosh, B. (2022). Ensemble empirical mode decomposition based deep learning model for short-term wind power forecasting. Floating Offshore Energy Devices: GREENER, 20, 58. DOI: https://doi.org/10.21741/9781644901731-8

- Pakrashi, V., Bhowmik B., González Sopeña J. M., Mucchielli P., & Ghosh B. (2020). Wind power prediction and early downtime detection for Ireland. In *Proceedings of the Civil Engineering Research Ireland 2020 Conference* (pp. 337-340). Available at https://sword.cit.ie/ceri/2020/12/2

Preprints

- González Sopeña, J. M., Pakrashi, V., & Ghosh, B. A Benchmarking Framework for Performance Evaluation of Statistical Wind Power Forecasting Models (2022). Available at http://dx.doi.org/10.2139/ssrn.4129779.

- González Sopeña, J. M., Dasgupta, T., Pakrashi, V., Xiao, H., & Ghosh, B. How poorly does my forecast perform under a wrong ARIMA model? A decomposition-based simulation algorithm for assessment (2022). Available at https://www.stat.rutgers.edu/home/hxiao/arima.pdf