TÉCNICO LISBOA

Electrical load, wind speed and power forecasting with feed-forward neural networks

Eduardo Praun Machado

Thesis to obtain the Master of Science Degree in

Energy Engineering and Management

Supervisors: Prof. Hugo Gabriel Valente Morais
Prof. Tiago Manuel Campelos Ferreira Pinto

Examination Committee

Chairperson: Prof. Duarte de Mesquita e Sousa
Supervisor: Prof. Hugo Gabriel Valente Morais
Member of the Committee: Prof. Sérgio Filipe Carvalho Ramos

October 2021

I dedicate this work to my family, who has always given me the necessary support and encouragement.

Acknowledgments

First, I would like to thank my supervisors, Professors Hugo Morais and Tiago Pinto, for proposing such an interesting and important research topic, the given encouragement, the availability, the expertise and the comments, which were certainly crucial for the development of this thesis. It was an extremely challenging task that has broaden my knowledge in an area that I have always been interested in learning and working on.

In addition to my supervisors, I also would like to acknowledge researchers Vanessa Guedes, Ricardo Dutra and all the others colleagues on the wind energy team of the Department of Materials, Energy Efficiency and Complementary Generation (DME) at the Electric Energy Research Center (Centro de Pesquisas de Energia Elétrica - Cepel), for providing a unique experience in the wind energy sector in a reference place for research in the energy field, for the wind farm data availability and the assistance with the data analysis. Also, I express my gratitude to Cepel for the granted master's scholarship during that time.

Finally, I would like to thank my friends, colleagues and professors at the Instituto Superior Técnico (IST) for their contribution during the many hours of study and for the enriching academic experience throughout the master's.

Resumo

A transição energética para fontes renováveis de energia e a tendência a eletrificação da sociedade vêm impondo um enorme desafio à operação do sistema elétrico devido ao aumento da complexidade, da variabilidade e das incertezas associadas a essas transformações. Os avanços recentes das tecnologias computacionais permitem o desenvolvimento de algoritmos sofisticados, eficientes e que conseguem processar uma grande quantidade de dados rapidamente. Nesse sentido, o uso de modelos de aprendizagem automática vem ganhando cada vez mais atenção do setor elétrico, uma vez que podem fornecer previsões precisas sobre o comportamento do sistema e auxiliar todas as partes interessadas a optimizar as suas atividades. O objetivo deste trabalho é desenvolver uma metodologia para melhorar as previsões de consumo de energia elétrica e da velocidade do vento geradas por um modelo de aprendizagem automática, nomeadamente, uma rede neuronal feed-forward (FFNN), incorporando uma etapa de correção dos erros de previsão, a qual envolve a estimação dos erros iniciais por meio de outra FFNN. Os resultados demonstraram que a metodologia proposta foi capaz de melhorar significativamente as previsões do consumo de energia elétrica em relação aos modelos de referência, enquanto, para as previsões da velocidade do vento, nenhuma diferença de desempenho notável foi observada. Por fim, estimativas da potência eólica gerada foram feitas a partir das previsões de vento e de uma curva de potência modelada com auxílio de um método de agregação. Os resultados obtidos pela metodologia proposta foram mais precisos do que os que utilizaram as previsões do modelo de referência.

Palavras-chave: Aprendizagem automática, Consumo de energia elétrica, Produção eólica, Previsões, Rede Neuronais, Velocidade do vento.

Abstract

The energy transition to renewable energy sources and the trend towards electrification of society have been imposing a tremendous challenge to the operation of the electrical system due to the increase in complexity, variability and uncertainties associated with these changes. The recent advances in computational technologies allow the development of sophisticated and efficient algorithms that can process a large amount of data in a very fast pace. In this sense, the use of machine learning (ML) models has been gaining increased attention from the electricity sector, as they can provide accurate predictions of the system behavior and help all the stakeholders to optimize their activities. The aim of this work is to develop a methodology to enhance the load demand and the wind speed forecasts generated by a ML model, namely, a feed-forward neural network (FFNN), by incorporating a forecasting error correction step which involves the prediction of the initial errors by another FFNN. The results showed that the proposed methodology was able to significantly improve the load demand forecasts in comparison with the reference models while, for the wind speed forecasts, any notable performance difference was verified. In addition to that, estimates of the generated wind power were made using the wind speed forecasts and a power curve modeled with the aid of a clustering method. The results obtained with the proposed methodology were more accurate than those that used the predictions of the reference model.

Keywords: Feed-forward neural networks, Forecasting, Load demand, Machine Learning, Wind power, Wind speed.

Contents

List of Tables

List of Figures

Nomenclature

ACF Autocorrelation Function

ANN Artificial Neural Network

ANWP Adjusted Numerical Weather Prediction

AR2 Adaboost

ARIMA Autoregressive Integrated Moving Average

BIC Bayesian Inference Criterion

CDF Cumulative Distribution Function

CFD Computational Fluid Dynamics

DBSCAN Density Based Spatial Clustering of Applications with Noise

DMD Dynamic Mode Decomposition

ECMWF European Centre for Medium Range Weather Forecasts

EMD Empirical Mode Decomposition

EU European Union

FFNN Feed-forward Neural Network

GA Genetic Algorithm

GHG Greenhouse Gas

GP Gaussian Process

GRUNN Gated Recurrent Unit Neral Network

GTA Grid Traverse Algorithm

HHT Hilbert-Huang Transform

IPCC International Panel on Climate Change

KF Kalman Filter

LR Learning Rate

LSTM Long Sort Term Memory

MAE Mean Average Error

MAPE Mean Average Percentage Error

MBE Mean Error Bias

MI Mutual Information

ML Machine Learning

MSE Mean Squared Error

NMAE Normalized Mean Average Error

NRMSE Normalized Root Mean Squared Error

NWP Numerical Weather Prediction

PDF Probability Density Function

PSO Particle Swarm Optimization

ReLU Rectified Linear Unit Function

RES Renewable Energy Source

RF Random Forest

RMSE Root Mean Squared Error

RNN Recurrent Neural Network

SCADA Supervisory, Control and Data Aquisition

SVM Support Vector Machine

SVR Support Vector Regression

tanh Hyperbolic Tangent Function

Chapter 1

Introduction

1.1 Motivation

According to the last International Panel on Climate Change (IPCC) report [1], there are strong evidences of human influence on the global climate change, which is characterized by the global average temperature increase, higher ocean levels, occurrence of catastrophic events, among others. The raise of greenhouse gases (GHG) concentration in the atmosphere is most likely the main driver of the changes on the Earth's climate. The energy sector accounts for a quarter of the world GHG emissions [1], being the one with the largest share. Therefore, it is not possible to discuss climate change without considering the energy sector.

The change of climate conditions can affect physical, biological and human systems in global and regional scales and require adaptation and mitigation measures. In order to reduce GHG emissions, several national, regional and global agreements have been signed, such as the European Union (EU) 2009/28/EC directive [2], in which it was agreed that the EU countries must fulfill at least 32% of its total energy needs with renewable energy by the year of 2030. This type of measure is in accordance with the findings presented in [3], which states that large-scale changes in the energy system is a must to achieve this goal and mitigate the global climate crisis. However, the energy transition towards cleaner sources demands profound and challenging changes in the sector infrastructure, policies, regulations, market design and operation.

The aging of conventional power plants, technological advances and cost reductions are allowing cleaner sources, mainly solar and wind-based systems, to boost its share in the electricity mix at the expense of fossil fuels [4]. Simultaneously, the electrification of significant sectors, such as transport and heating, are increasing the load demand while the system decentralization is altering the load patterns and the energy flow, as consumers are changing their roles to become prosumers, i.e. someone who both produces and consumes energy [4]. Moreover, the growth of digital and storage technologies also increases system complexity and can expose it to cyber threats [5]. All this transformation requires various adaptation measures which involves technical, economical and political issues [5] that must be applied to ensure a reliable, affordable, safe and high-quality electricity.

The challenges and efforts to integrate renewable energy sources (RES) in the electricity grid will depend on their degree of penetration on each country/region. Usually, an early stage modification is the necessity to expand the transmission grid due to the fact that the RES power plants are commonly located in remote areas, where there is a higher availability of sun and wind resources. After this, the changes can go from simpler ones, such as minor operational adjustments for small share of RES, to more complex ones, such as long-term energy storage for electricity mix based on RES [4]. Furthermore, the larger share of RES in the electricity mix increases the system unpredictability and variability due to the uncertain availability of natural resources.

This uncertainty of energy supply and demand can cause grid instability issues, such as overvoltage and frequency deviations. To overcome this situation, the system must be flexible and resilient enough in order to cope with rapid generation and load changes and balance them at every moment. In this regard, recent technologies and approaches such as small-scale energy storage systems (e.g. batteries) and demand response programs have been gaining increased attention [5]. However, they are not always feasible as they can be extremely expensive and still need more improvement to be applied in a large scale. On the other hand, one common way for system operators to deal with this variability is by defining a certain amount of energy reserve that can be used to adjust the system frequency. In liberalized markets, such as the Iberian Electricity Market (MIBEL[1]), the necessity and use of this reserve can configure an extra cost to the system operation and lead to an increase of the electricity price.

The above-mentioned solutions have in common the fact that they try to deal with the system variability by giving immediate responses to instantaneous deviations, by discharging the batteries, turning off electrical appliances or increasing the power of a generator for example. Nonetheless, they all typically have their behavior planned based on power generation and load demand predictions. Therefore, accurate forecasts are essential for these tools to optimize their performances and, consequently, the whole system operation.

The interest for energy demand and supply forecasts has significantly increased since the oil crisis in the 1970's [6]. At that time, most of the employed models were statistical based such as linear regression, ordinary least squares, stepwise regression, among others. However, the growing complexity of electrical systems and the unknown relationships between multiple variables imposes great difficulties that these simpler models can not handle.

On the other hand, a boost in the application of computational methods have been observed in recent years. The technological developments in computing allow the creation of sophisticated and efficient computational methods which typically use advanced statistical concepts. They can be combined with statistical models, used to estimate its parameters or to make forecasts with reduced computational time and improved performance [6]. With the ever-growing data availability from the distribution/transmission system operators, power plants supervisory, control and data acquisition (SCADA) systems, electricity market, weather forecasts and these models proficiency to handle complex data relationships, the forecast accuracy can be significantly enhanced. More specifically, ML models have shown great capability to generate accurate predictions of the system behavior from energy generation to consumption

[1] MIBEL webpage: https://www.mibel.com

and can benefit the system operators to better handle the sector issues, policy-makers to plan future actions, market agents to optimize energy tariffs and, also, producers to manage their power plants [7]. Therefore, the employment of these models can help to achieve climate change mitigation and energy transition goals efficiently and should be of great interest for the sector.

1.2 Objectives and Deliverables

The main goal of this work is to investigate and propose a ML method to generate forecasts of electrical load demand and wind speed in different time scales by incorporating an error correction step on the initial models results. Also, the wind power output was estimated by using the wind speed forecasts and a wind power curve clustering and adjustment method. The major contributions of this work are summarized below:

- State-of-the-art review on subjects relevant to this research, namely ML algorithms for forecasting and their recent applications for energy demand, wind speed and power forecasting;
- Preliminary analysis approaches and techniques that can be applied for time-series data;
- FFNNs forecasting models built with well-established Python's libraries for ML;
- Forecasting models error analysis;
- Clustering method which can be used to identify outliers in wind power curves;
- Comparison of results achieved with datasets with different characteristics and suggestions for future improvements of the proposed models;
- A paper entitled "Wind Speed Forecasting using Feed-Forward Artificial Neural Network" was accepted for the 18th International Conference on Distributed Computing and Artificial Intelligence (DCAI) 2021;
- A paper entitled "Electrical Load Demand Forecasting using Feed-Forward Neural Networks" was submitted for the Energies journal.

1.3 Structure

The present work is organized as follows:

- In Chapter 1, an introduction to the subject of this research is presented together with the context in which it is inserted, the motivation for its development and the objectives targeted by this work;
- In Chapter 2, a state-of-the-art review on forecasting with ML and current applications for load demand, wind speed and power forecasts are provided;

- In Chapter 3, the database used for load demand forecasts, the preliminary analysis conducted and the proposed methodology are presented;
- Chapter 4 is organized in the same fashion as Chapter 3 but for the wind speed and power forecasts;
- In Chapters 5 and 6, the results obtained with the load demand and the wind speed and power models are discussed respectively;
- In Chapter 7, the major findings of this research are summarized and suggestions for future works are proposed.

Chapter 2

Background

This Chapter provides an overview of the concepts and methods on the subject of this research. To do so, an introduction to forecasts with ML models is presented in Section 2.1. Furthermore, recent studies with ML applications on electrical load, wind speed and power forecasting are discussed in Sections 2.2 and 2.3.

2.1 Forecasting with Machine Learning

With the rise of the Third Industrial Revolution, also known as Digital Revolution, and the creation of the Internet, since the second half of the 20th century, the presence of computers in everyday life has been increasingly frequent and indispensable. The information, which was previously stored mostly in paper documents, turned into digital. The data digitalization allows easy access to information, which requires less space and can be stored safer, saving money and time. Also, data analysis was facilitated and impacted every sector in society, changing how governments and companies make decisions and carry their business [8]. Coupling the available data with expertise and computer-processed mathematical/statistical models, the decisions, which were formerly based on human knowledge and basic analysis, started to be more appropriate and precise. However, with the present technological development, data is being collected at such a fast rate and in a such large amount that it became extremely hard for a human being and simpler models to analyze.

In order to analyze and make decisions based on all this data, the use of ML models is hugely beneficial. ML refers to computer algorithms that are able to learn from data and previous experiences without being explicitly programmed to do so [9] and is a subset of artificial intelligence, which concerns the ability that a computer has to perform tasks that are associated with and normally require human intelligence [10]. With systems that can learn from data as humans but at a much quicker pace, the decision-making process can be significantly improved. This can be done by either employing these systems to extract hidden information from data and generate insights or using them to make the decisions on their own [11].

Tasks that are usually associated with ML models are: image and speech recognition, medical diag-

nosis, products recommendations, spam filtering and traffic prediction. Depending on the task and on the available data, ML problems can be classified as: supervised learning, unsupervised learning and reinforcement learning [12].

In supervised learning problems, the available data comprises features that are associated with labeled data. The name supervised learning actually comes from the fact the every input has a target value to learn from [10]. Normally, they are used for classification tasks, where the model labels the inputs (discrete output); and regression tasks, where the program returns a numerical value given some input (continuous output). Examples of classification and regression problems are, respectively, spam filtering and energy price prediction.

In unsupervised learning problems, the available data is not labeled and the system tries to identify useful properties and patterns on the data by itself. Commonly, they are used to learn a probability distribution function on the dataset. The learned pattern can then be used to denoise or divide the data into groups of similar examples (clustering). One typical application of these methods is to identify credit card frauds [10].

Finally, the reinforcement learning models does not use solely the provided data and also interact with the environment by means of a feedback loop that evaluate its output [11]. Based on the evaluation of previous experiences, it must learn by itself the best strategy to avoid bad feedbacks and maximize good ones. Examples of reinforcement learning are inventory management and traffic signal control.

Since the object of this research is a supervised learning problem and, more specifically, a regression task to predict electrical loads, wind speed and power, the next paragraphs cover details about this type of task, such as its mathematical foundation, problems, common practices and models examples.

Before launching a system and starting to providing predictions, these systems must learn how to perform the task. The way that they learn from data is a stage called training and it can be divided into two groups: batch or offline and online learning [11]. In batch learning, the model must use all the available data to train at once, which usually requires a lot of computing resources and time. Due to these requirements, they are generally trained offline before being deployed to execute what it learned. If an updated version of the system is needed, a new training must be performed every time. On the other hand, when the data is too much to process, comes continuously or changes rapidly, the use of online learning is more appropriate [11]. In this type of learning, the model is trained with small groups of data, called mini-batches, which reduce the cost of updates and improve the training time.

Regardless of the learning strategy, the models make predictions y of observed (target) values t using an input vector *x*. Furthermore, the forecasts are also a function of the models parameters p, which depend on the chosen model and can be, for example, polynomial coefficients in a linear regression model or the weights and biases in a neural network. Equation 2.1 presents the mathematical formulation of such models.

$$y = f\left(x, p\right) \tag{2.1}$$

During the training, the learning algorithm estimates the models parameters to make predictions of

the target values and, then, update it to enhance its performance. This is usually done by minimizing a loss function (also known as cost or error function), which measures how far the predictions are from the true values [12]. There are several different loss functions that can be used in ML problems, some of which for classification and others for regression tasks. Among the ones used for regression, the most common are: mean absolute error (MAE), also known as Manhattan or L1 norm, and mean squared error (MSE), also called Euclidean or L2 norm. They are presented in equations 2.2 and 2.3 respectively. In these equations, y_i is the forecasted value, t_i is the target value and n is the number of points in the dataset. The major difference between these two metrics is that the latter is more sensitive to outliers due to the squared term [11].

$$E(p) = MAE = \frac{1}{n}\sum_{i=1}^{n}\left|f\left(x_i, p\right) - t_i\right| \tag{2.2}$$

$$E(p) = MSE = \frac{1}{n}\sum_{i=1}^{n}\left(f\left(x_i, p\right) - t_i\right)^2 \tag{2.3}$$

The search for the loss function minimum is commonly done by computing its gradient ($\nabla E(p)$), which is the vector containing the partial derivatives of $E(p)$ [10]. The partial derivative $\frac{\partial}{\partial p}E\left(p\right)$ indicates how the function changes with a small change in one of the parameters. Therefore, the gradient vector points to the direction of the steepest increase of the function. As the learning algorithm goal is to minimize the error, with this approach the parameters can be updated at each iteration i by going in the opposite direction. This is an optimization method known as gradient descent and its mathematical formulation is shown in equation 2.4, where η is called the learning rate (LR).

$$p_{[i+1]} = p_{[i]} - \eta\nabla E(p) \tag{2.4}$$

The LR is a model hyperparameter and is the rate at which the algorithm updates the models parameters. The difference between a parameter and a hyperparameter is that the former is adapted by the learning algorithm, while the latter is defined by the user in order to control the behavior of the model or the learning algorithm [10]. Examples of other hyperparameters are the number of layers and nodes of a neural network, the kernel type for Support Vector Machines (SVM) and the maximum depth of the tree in a Decision Tree. Regarding the LR, setting higher values to it means that the program will quickly forget old data and adapt itself to the new one. It is important to do this carefully, as it can result in a system more sensitive to noise and outliers. On the other hand, slower changes will be verified with lower values.

If the gradient calculation is done using batch learning, the optimization method is typically called (batch) gradient descent. With this approach, the algorithm can take too long to converge, as it computes the partial derivatives using the whole data. However, once it finishes, the solution is optimal as the loss function minimum can be surely found. On the other hand, with online learning, the method is known as stochastic or mini-batch gradient descent [12]. In this case, the calculations can be much faster and a solution close to the optimal can be reached. There are other optimization methods based on the

gradient descent such as Adam, which uses the concept of momentum by updating exponential moving averages of the gradient [13], Adagrad and RMSprop.

Before building any model, a common practice is to set a fraction of the dataset apart. This set is called test set and is not used to train the model at all, only to evaluate it. The way to split the data into train and test sets depends on the type of problem and the characteristics of the data itself, such as its size and probabilities distributions. The most important aspect of this practice is to create a test set that is representative of the whole population [11].

When the model performs badly on the training set, it indicates a problem called underfitting. It can happen because the model was too simple to identify and learn the structure of the data [11]. Underfitting can be prevented by increasing the number of parameters of the model or feeding better features to it. On the other hand, when the model performs well on the training set where it learned from and poorly on the test set, which it has not previously seen, it is a signal of overfitting. This problem occurs when the model is too complex to the size and noisiness of the training data and it can be avoided by simplifying the model either by reducing the number of parameters or the number of features in the dataset, gathering more and denoising training data [11]. Also, a useful way to reduce the model complexity is by using a technique called regularization [12]. This approach involves the addition of a penalty term to the error function so that the coefficients can not reach large values. The penalty term is typically the L1 or L2 norms of the parameters which is controlled by a hyperparameter named regularization factor In Figure 2.1, an example of the learning curve where these types of problems can be seen during training is presented.

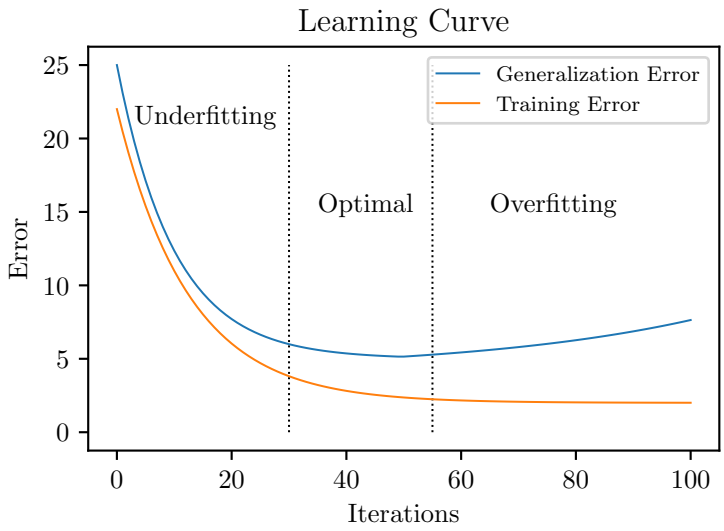

Figure 2.1: Example of a typical learning curve

Tuning the hyperparameters using the same test set to evaluate the model performance multiple times does not assure that the results will be satisfactory. Sometimes, even with low errors on the test set, the models performs poorly on new data. One of the reasons for this to occur is that the model was adjusted to perform well only on that specific test set [11]. A common solution to avoid this problem is to split the data not in two, but in three sets. This third set is called validation set and it is usually about 20% of the training set. The validation set is used to evaluate the model performance after training and adjust

hyperparameters while the test set is left to assess only the final optimized model. A further alternative to achieve better results is to use cross-validation [10]. This technique is typically based on dividing the training set into k-folds and repeating the training for k times using a different validation set at each time. Then, the validation error is taken as the average across all folds. This approach is appropriate when data is scarce but it increases significantly the computational cost.

Although the above-mentioned approaches can help to optimize a model performance, the main driver for a high ML model effectiveness is still the available data. To solve complex problems, these models generally need a large amount of good quality and representative data [11]. A sufficient quantity of examples is necessary so that the model can learn to identify the patterns. To do so, they must be fed with features that explain the desired output which should be carefully selected. Furthermore, the data should preferably come with few or without errors, outliers and/or noise. Given the importance of the data on the results, most of the time to prepare ML models is spent on data preparation [11], i.e. selecting features and cleaning data. In this sense, there are plenty of methods and algorithms that can be employed to do this. However, they are out of the scope of this research and, thus, will not be detailed. Finally, there is a large amount of ML algorithms that can be applied for regression tasks such as SVM, Decision Trees, Linear Regression and FFNN. The FFNNs will be presented in detail in Subsection 3.3.1, as this was the model applied in this work. For other algorithms, details can be found in [10–12].

2.2 Electrical Load Forecasts

2.2.1 Overview

In order to reduce GHG emissions and mitigate climate change, substantial changes on the energy sector will be required [3]. In that regard, the increased need for electricity, the change of power generation mix and load pattern are some of the already observed transformations. With all this, the electricity grid behavior becomes more unpredictable and its operation and control more complex which can lead to greater supply instability. Therefore, techniques and methods capable of increasing system reliability are extremely important.

One way to ensure more trustworthiness and better manage the system is by anticipating the load demand. When accurate forecasts are made, the decisions regarding the power system operation, maintenance and planning become more efficient [7, 14]. Furthermore, improvements on energy policies and tariffs can be achieved. In recent years, much of the research have been focused on the development of models to forecast the electrical load in different time horizons. These periods are often classified as short-term, which goes up to 1 week ahead; medium-term, from weeks to 1 year; and long-term, for future years [15]. Also, each of these timeframes have different applications, with the first being more important for daily operation and cost minimization [16] and the others for fuel reserves estimation, maintenance and capacity expansion planning [17].

Several approaches have been employed recently to make these forecasts. These approaches can

be separated into three categories: statistical based, computational intelligence based and hybrid approaches [6]. Statistical models usually embrace uni or multivariate time-series models and regression techniques, such as Autoregressive Integrated Moving Average (ARIMA) and linear regression, while computational intelligence models are mainly related to ML approaches. Commonly, statistical based methods are less memory intensive due to its simplicity and, thus, faster to execute. On the other hand, ML models are capable of identifying nonlinear relationships between inputs and outputs and can be extremely time consuming. However, this level of complexity can be necessary to achieve better results [15]. Finally, hybrid models combine features from statistical and computational intelligence models. They generally use the former to preprocess and/or select the input data that will be fed to the latter.

These forecasting models can be implemented using a large range of inputs that can be divided into four major categories: socio-economic, such as the region average income and GDP; environmental, such as mean temperature; building and occupancy, which is related to building sizes and dwelling types; and time index, that is related to the date stamps used as inputs [18]. Also, electricity demand historical data is generally taken into consideration. However, the choice of these inputs will depend on the time scale and type of the region of the study. Usually, historical, environmental and time index data are more common for short-term forecasts in a region scale [18].

2.2.2 Literature Review

Recently, several neural networks structures have been employed to develop for short-term load forecasting. In [19], a model composed of a Convolutional Neural Network, a bidirectional Gated Recurrent Unit (GRUNN) and a Long Short Term Memory (LSTM) recurrent neural networks (RNN) was presented. First, the authors computed hourly autocorrelation coefficients of hourly loads and temperature, which were used to calculate the kernel size of convolutional layers. Later, two-dimensional convolutional layers with the load and temperature time-series were used to extract features from this data. The extracted features were used as inputs for the RNNs to forecast the one-hour-ahead load demand. Bidirectional RNNs combines RNNs which one moves backward and the other forward through a sequence in such a way that the output of the model depends on both the past and the future [10]. The proposed model [19] was evaluated by testing it with two electrical load datasets in North America and comparing it with the performance of other ML methods. The results have shown that the proposed model had the lowest MAE, generally close to 60MW (load range approximately between 1000 and 3500MW), and Mean Average Percentage Error (MAPE), less than 1%, and it was able to accurately forecast the demand with similar performance at every hour of the day. Furthermore, the computational time demonstrated the possibility to implement it in real time despite the high complexity of the model.

In [20], a hybrid Artificial Neural Network (ANN) to forecast the day-ahead load in a smart grid was proposed. The strategy was divided into three modules: preprocessing, forecasting and optimization. The goal of the first module was to remove irrelevant and redundant features from the dataset by applying a mutual information based technique. The mutual information (MI) represents the uncertainty reduction about one variable as a consequence of observing another one and is a concept related to the

information theory [12]. The forecasting module used the features selected from the previous module as inputs for an ANN trained by the Multivariate Autoregressive Algorithm to make predictions of the load demand. Finally, the last module implemented a heuristic optimization technique, namely a modified Enhanced Differential Evolution algorithm, to minimize the forecasting error. The proposed model was tested using historical load data collected from two grids in the USA and had its results compared to the ones obtained with two other ML models. The results suggest that the proposed methodology was able to generally achieve better performance than the benchmark models in terms of the MAPE, which was less than 2%, and variance at a lower computational cost.

In [21], the authors introduced an Advanced Wavelet Neural Network to forecast one-step-ahead load demand. The proposed approach was composed of four stages: load decomposition, feature selection, prediction models creation and forecasting. The main idea of data decomposition is to break the data into its constituent parts to find universal or functional properties that are not observed in its usual representation [10]. In the first stage, a Wavelet Packet Transform with best wavelet basis selection was used to decompose the electricity demand into low and high frequency components. For each wavelet component, a feature selection process based on MI was conducted to find the best feature set. In the next stage, one neural network was created for each component which had the corresponding selected features as inputs and the predicted load as output. Finally, the one-step-ahead forecasts were obtained by adding the individual contributions of each component. This approach was evaluated with two datasets from two different countries (Australia and Spain) and with different time resolution (5min and 60min). The proposed methodology was able to slightly outperform the benchmark models, namely Linear Regression and Model Tree Rules, in terms of MAE (between 20 and 90MW for Australia and 330 and 930MW for Spain) MAPE (less than 1% for Australia and 5% for Spain) for both datasets. It is worth mentioning that the Spanish model showed large accuracy variation over different seasons and time of the day than the Australian, which was justified by the higher load variability in Spain. Furthermore, a multi-step ahead forecast (up to 1 hours for Australia and 12 hours for Spain) was tested and a performance deterioration was observed for longer forecast horizons.

In [22], a LSTM network using a cross-correlation based transfer learning approach is proposed to forecast 15-minutes-ahead load demand. Transfer learning is a methodology that identifies similarities in different datasets and allows the use of knowledge from other tasks on related ones [10]. This approach can be extremely useful when the available data is scarce. In this work, energy demand data from several randomly selected buildings over approximately one year was collected for the transfer learning step while the load demand to be estimated came from data collected from a university building in Turkey over one month. Also, both of them was sampled at every 15-minutes. In a first moment, the cross-correlation between the data of the buildings to be transferred and the one to be predicted was computed. Then, the most suitable building data, i.e. the one with the highest cross-correlation score, was used to train a LSTM model. After the training step, the learned weights were transferred to another LSTM model, which was retrained using a fraction of the data (training set) from the building to be predicted. Finally, the performance of the model was assessed on the remaining data (test set) and compared with other ML models, namely Random Forest (RF), Extreme Gradient Boosting, Light Gradient Boosting Machine

and a LSTM trained without the transfer learning approach. The results showed that the proposed model was able to outperform the benchmark models in terms of the RMSE, MAE and MAPE. Moreover, the contribution of the transfer learning approach was evident, as a significant improvement of the LSTM model could be verified. The authors also observed that the best proposed models came from those weights obtained with the data from the buildings with the highest cross-correlation. However, the MAPE was higher than the one observed in other works (about 8%) and details about the performance of these models for each day and time was not provided.

In [17], ML approaches including ANN, Multiple Linear Regression, Adaptive Neuro-Fuzzy Inference System and Support Vector Regression (SVR) were employed to forecast week and year-ahead electricity demand in Cyprus. The inputs for the models were time, environmental data such as temperature, humidity, solar irradiation and socio-economic data such as population, gross national income per capita and electricity price. The results have shown that the ANN and the SVR models performed significantly better than the other two for both short and long term forecasts, with the MAPE around 2% in the first scenario and 5% in the second.

In [23], a hybrid model that combined ARIMA with SVM was applied to forecast hourly load demand. This work used historical data from a state in the south of India to estimate the ARIMA parameters and generate initial load forecasts. Then, the outliers of the initial predictions were detected by means of the percentage error method and corrected using the deviation method. The forecast errors data of the corrected ARIMA model output and other two variables, namely day and average temperature of the week, were given as inputs for the SVM model estimate the initial forecast error. Finally, the initial forecasts and the expected errors were added to obtain the final load prediction. The authors found that the proposed ARIMA-SVM model was able to outperform single ARIMA and SVM models in terms of MAPE (4.15% versus 5.16% and 4.97% respectively). Furthermore, the performance of the proposed model without the outlier detection approach was worse (6.23% versus 4.15%).

Another hybrid model was proposed in [24] which combined a SVR model and a two-step parameter optimization algorithm using Grid Traverse Algorithm (GTA) and Particle Swarm Optimization (PSO) to forecast the load demand in several short-term scales (from 5-minutes to 16-hours-ahead). The tested data was comprised of 80 days of load from a distribution feeder. In a first moment, this data was preprocessed to eliminate excessively deviating samples using a mapping algorithm. Then, a GTA designed with cross-validation was used to narrow the SVR parameters search area. After, the PSO searched for the best parameters of the SVR model in the GTA solution space also using cross-validation. Finally, with the best SVR parameters set obtained in the optimization step, the forecasts were conducted. The results showed that the MAPE increased for longer forecasts horizons, going from 1.31% for the 5-minutes-ahead to 3.19% for the 16-hours-ahead forecasts. Furthermore, the proposed methodology achieved better results than three benchmark models, namely ARIMA (11.21%), SVM with genetic algorithm (5.27%) and ANN (6.62%) with faster execution (except when compared to the ARIMA model).

In [25], three ensemble learning algorithms, namely RF, Gradient Boosted Regression Tree and Adaboost (AR2), were employed to forecast one-hour-ahead electricity load. Ensemble models are those that combine several models which are trained separately in order to reduce the generalization

error [10]. The main advantage of this type of model is that, on average, they perform at least as well as any of its members and, if the errors of its members are independent, they will perform significantly better. In this work [25], historical electricity demand from an office building was collected at every 10 seconds and averaged to an one-hour basis. Different training strategies merging various features such as historical, time index and environmental data were tested to forecast every hour of a single day. The authors found that using the time index data with the most recent temperature measurements and a few past load observations provided the best forecasts in general. The results of this work were compared to the ones in [26], which generated forecasts for the same building using SVM and three fuzzy rule based models. It was observed that the AR2 model outperformed the SVM model in terms of MAPE (5.34% vs 5.82%). Moreover, the other two ensemble models performed better than the fuzzy rule based models.

In [27], a short-term load forecasting model based on error correction using dynamic mode decomposition (DMD) is proposed. Using two years of electricity demand data from a city in China, the authors built several load forecasting models including ANN, SVR and ARIMA. They used historical data, namely previous day, same day in previous week and similar day loads, as inputs. The latter was obtained by grey relational analysis which is a method that aims to find highly correlated data. With the errors achieved by these models, the DMD was applied to forecast the errors. The DMD is a data-driven method that can extract complex spatiotemporal features from data. Extensive experiments on small and large geographical scales were conducted to evaluate the proposed methodology. Nearly in all cases, combining the DMD for the error correction with the load forecasting model resulted in better results. The authors tested different decomposition techniques, including a Wavelet Transform, which generally had a worse performance. Also, it was possible to notice a smaller prediction accuracy for the small area forecasts which was probably caused by its higher load variability.

Finally, it was observed that plenty of recent works are focused on the application of complex ML models that require large amount of data. However, this data is not always available which implies a need for models that can provide good estimates using limited historical data. Furthermore, despite the good results of these works, the errors are analyzed only in their dimension and not in their pattern. Therefore, specially when working with limited data, the development of models that can identify the error patterns can help to improve the original forecasts.

2.3 Wind Speed and Power Forecasts

2.3.1 Overview

Over the last decades, power production from renewable sources have experienced a sharp expansion. With growing concerns about the global climate crisis, the expectation is that this trend will continue to escalate. Amid those sources, wind energy arises as one of the most attractive due to its high generation capacity, efficiency and cost-benefit ratio [28]. However, as others RES, wind power generation also suffers from resource stochasticity and intermittency which impose a challenge to its large scale penetration as it can undermine the whole electrical system operation [29].

To surpass these issues, accurate wind energy forecasts can play an essential role. They can, for example, help to optimize the market prices, producer profits [30] and electricity supply reliability [31]. Several factors, such as the environmental conditions, weather and time of the day can affect the predictions [32]. Nonetheless, the volatile characteristics of the wind hamper forecasts models to be precise [33]. Hence, minimizing the uncertainty associated with the wind is a keystone to improve wind energy forecasts [32].

As for electrical load forecasts (Section 2.2), the wind speed and power forecasting models can be classified according to the time horizons and methodology. In Table 2.1, the most common time scales and its applications found in the literature are presented. With regard to the methodology, the models can be divided into physical, statistical, ML and hybrid models. Physical models work with complex mathematical equations that relate the terrain, atmosphere and external conditions to generate weather forecasts [30]. They are also known as NWP models and can make forecasts for different time horizons, height levels and parameters. However, they tend to be extremely time consuming due to the high level of complexity [34]. On the other hand, statistical models are much simpler and can run faster. They usually have good results for the short-term but their performance quickly worsen for longer terms [31]. As mentioned in Section 2.2, ML models can identify nonlinear relationships between inputs and targets by working with algorithms that enable them to learn from data. They usually are not as time consuming as physical models and have better results than statistical methods. Finally, hybrid models can combine some techniques to make forecasts.

Table 2.1: Wind speed and power forecast horizons and applications (based on [31, 34, 35])

Forecast Horizon	Range	Applications
Very short-term	Up to 30 minutes ahead	Regulation actions, market clearing, real-time grid operations, turbine control.
Short-term	From 30 minutes to 6 hours ahead	Load dispatch planning, market operational security.
Medium-term	From 6 hours to 1 day ahead	Unit commitment decisions, market operational security, energy trading.
Long-term	From 1 day to 3 days ahead	Unit commitment decisions, reserve requirements, energy trading, maintenance planning.
Very long-term	From 3 days to longer forecasts	Maintenance planning, operation management, feasibility studies.

2.3.2 Literature Review

In [36], the authors proposed a bidirectional GRUNN based model for NWP wind speed error correction and used the adjusted wind speed to forecast wind power up to 24 hours in advance. The model used NWP wind speed error standard deviation as a weight time series, which was later decomposed into trend and detail terms by means of Empirical Mode Decomposition (EMD). These two terms and the NWP wind speed were taken as inputs for the GRUNN so that corrections in the latter can be made. Finally, the corrected forecast was used to estimate wind power by using the wind power curve. An

experiment using wind speed, power and NWP wind speed data sampled every 15 minutes from a wind farm in the Sichuan Province, China was carried out to validate the proposed methodology. The results suggest that the proposed model tended to reduce high NWP wind speeds and increase low speeds. However, no metric was given to analyze the results. With regard to the wind power forecast, it was able to outperform predictions made using the original NWP wind speed and benchmark models, namely SVM and ANN, with slight improvements on the RMSE and the MAE.

In [37], an NWP downscaling model with two different configurations was tested for hourly and sub-hourly 100-meter wind speed forecasts: one using variables that described the boundary layer, winds and temperature which were available from NWP outputs and another adding the error between measured and NWP wind speed as an input. The downscaling model used a parametric approach with linear regression, which was developed in [38], coupled with stepwise regression based on Bayesian Inference Criterion (BIC). To verify the methodology, two years of data from a wind farm close to Paris, France was used and the results were compared with those of the original NWP forecast and benchmark models, namely Autoregressive Moving Average, ANN and persistence, by means of the normalized RMSE (NRMSE), the improvement over persistence (ΔRMSE) and the MAE.

In the hourly scenario [37], forecasts were made up to 12 hours in advance. The results showed that the proposed methodology was able to outperform all the benchmark models, except for the first configuration for 1-hour ahead forecasts. Compared to the NWP forecasts, the results were very similar. The second configuration presented the best performance for all time horizons and tended to converge to the one of the first configuration as forecast horizon increased. Also, the improvement of the proposed methodology over the benchmark models was more significant when the forecast horizon increased, especially when compared with the persistence model. It is worth mentioning that the performance of the models that used past observations as inputs worsened as the forecast horizon increased, while it was kept constant for models that did not.

In the second scenario [37], 10-minute forecasts were made up to 3 hours in advance. The results for sub-hourly forecasts were very similar to the ones found in the hourly forecasts, with the second configuration presenting the best performance and with more significant performance differences for longer forecast horizons. The authors also computed the power output by training the proposed models with production data and by using the wind speed forecasts and a adjusted power curve. The power curve was obtained by fitting the power output means every 0.5 m/s. They found similar results for both approaches and also more notable differences for longer forecast horizons.

In [39], a short-term wind power forecasting with NWP adjustment model was proposed. The authors developed a framework composed of three modules: wind power forecast, abnormality detection and data adjustment. In the first module, a “moving” input vector with previous NWP wind speed outputs and measured/forecasted wind power together with NWP wind speed outputs for the next hours was used to train an ANN in order to predict the next hour wind power. The second module used the wind power forecast error rate to label the ANN outputs as normal or abnormal forecasts. The samples with an error rate greater than a pre-defined threshold were classified as abnormal and used as inputs to adjust the model. Several features such as peak-valley difference, average, standard deviation, maximum ramp

up and maximum ramp down were extracted from the input vector and fed in a K-Means algorithm coupled with BIC to define clusters of abnormal patterns in the input data. After the model was trained, new observations went through the feature extraction procedure and had its Euclidean distance to each cluster center evaluated. The points that belonged to any cluster were classified as abnormal while the others as normal forecasts. The final module consisted of an ANN similar to the one of the first module but trained to predict the error rate of the abnormal forecasts. The inputs of this model were chosen by means of a feature selection algorithm, namely the random probe, which was applied on the features extracted at the previous stage and only for abnormal samples. Also, to consider the possible differences on the error pattern between different clusters, one single model was trained for each cluster. To validate this methodology, a data set of 530 days with wind power measurements and NWP data of a wind farm from the Global Energy Forecasting Competition 2012 – Wind Forecasting was used. The results suggest that the proposed model was able to correctly identify abnormal forecasts and to reduce the wind power forecast RMSE in comparison with the same method without the adjustment step and with the persistence model. Also, it could be noticed that the forecast error increased for longer horizons, which was also a finding in [37].

In [40], day-ahead wind power forecasts were made in a two-stage approach which was based on the combination of Hilbert-Huang Transform (HHT), Genetic Algorithm (GA) and ANN. The HHT is a decomposition method based on EMD employed to extract features from data while the GA is an optimization algorithm which relies on genetics and natural selection principles. Initially, an ANN trained with the GA used the decomposed NWP meteorological variables as inputs to predict the wind speed at wind turbine hub height recorded by the SCADA system. In the second stage, another ANN built in the same fashion to map the wind power curve characteristics using historical SCADA records and then used the forecasts made on the first stage to predict the output power of the wind farm. To evaluate the proposed methodology, the authors used one year of data from the SCADA system of a wind turbine in Beijing, China and from a NWP model. The proposed model performed better than four other approaches, namely persistence and ANNs that used other feature extraction and/or optimization algorithms, in the four seasons of the year (summer, autumn, winter and spring) considering the MAPE, the normalized MAE (NMAE), the error variance and the forecast skill. The authors also analyzed the impact of input data on the forecast MAPE by creating 5 subsets with different NWP inputs. Each of these subsets had one more variable than the previous one and the results showed a performance improvement every time a new variable was included.

The study in [41] presented a hybrid model that combined physical and statistical approaches to forecast wind power output up to 72 hours ahead. To do so, a downscaling model was used considering NWP outputs as initial and boundary conditions to make wind speed forecasts. Later, a Kalman Filter (KF) was employed to correct systematic errors on these predictions. Also, the authors employed the manufacturer's power curve and adjusted equations from the SCADA system to estimate the power output. Finally, another KF was employed to improve the quality of the forecasts. A case study was carried out in two different sites in Brazil to validate the proposed methodology. Each of the downscaling model simulations took 60 to 70 minutes to run and presented a performance drop at the end of each 24-

hour period. The authors found that the NWP model usually underestimated the wind speeds and that the KF was able to enhance the initial forecasts by bringing the mean error bias (MBE) close to zero and improving the RMSE, the Nash-Sutcliffe coefficient and, for one of the regions, the anomaly correlation coefficient. With regard to the power output, the Kalman-filtered values also brought MBE close to zero, but the improvements were not as significant as for the wind speed. Moreover, the predictions using either the manufacturer's or the adjusted equations were very similar.

In [42], a pure ANN and a hybrid method, based on the combination of an ANN and a computational fluid dynamics (CFD) model, were proposed to forecast the wind power output from a wind farm in Italy. For the pure ANN, a model for each turbine connected directly the inputs to the outputs, which were the mesoscale wind conditions from a NWP model and the power output of the turbine respectively. In the second case, the target of the ANN was the wind speed at a reference point in the wind farm. The forecasts of this model were given to the CFD model to estimate the wind speed for each turbine. Then, the wind power output was obtained by using these values with the theoretical power curve of the turbines. The authors observed that the pure ANN performed slightly better than the hybrid method in terms of NRMSE and NMAE. However, they were able to identify different behaviors and showed that one can complement the other. The hybrid model performed better during wind flow accelerations as it can reproduce the dynamics between the wind and the terrain. On the other hand, the pure ANN was more accurate in extreme wind speeds at which the power output did not correspond exactly to the one of the theoretical power curve.

In [43], an ANN combined with a wind power curve filtering method based on Gaussian Processes (GP) was proposed to model the power curves of the turbines in a wind farm in Chile. At first, the wind speed and direction data corresponding to known breakdowns and maintenance periods were discarded. Then, the GP filter was used to identify and remove samples which the power output deviated more than a given expected error threshold value given the wind speed and direction. The power curves obtained with the ANN and other four methods, namely the manufacturer's, the International Electrotechnical Commission, a parametric and a non-parametric, were compared using both the dataset only without the known stoppages and the one provided by the GP filter as inputs. The results showed that the GP filter was able to correctly identify samples of abnormal behavior, i.e. where the power output differs considerably from the expected. In the case without the GP filter, the ANN had the lowest RMSE in general but did not have the best results for two of the three turbines analyzed. With the filtered dataset, a significant improvement was verified, especially for the ANN, which turned out to be the most accurate for the three turbines.

Finally, as for the previous section, much of the recent works develop models that require a large amount of data. Therefore, there is also a need for models that can work with a limited amount of data. To overcome this problem, the analysis and use of the initial models error data can be beneficial. Furthermore, for wind power forecasts, it is common to assume the manufacturer's power curve as the most accurate. However, this is a general assumption and the turbines can deviate from this behavior. Thus, a method to model a specific wind power curve using data from individual turbines can be more adequate.

Chapter 3

Electrical Load Forecasting Models

In this Chapter, the database used to make the electrical load forecasts is introduced in Section 3.1. Also, in Section 3.2, the findings of the preliminary analysis of this data are discussed. Finally, the proposed methodology is presented in Section 3.3, with details about the FFNN employed in this work as well as the particularities of the initial and the error models.

3.1 Database

The database for the electrical load forecasts consisted of historical data collected from Enedis[1], which is the major distribution system operator in France. This data is freely available and it is published in a consolidated form, so that the information from producers and consumers is kept anonymous. The collected data refers to the measured load demand in an industrial area connected to the medium voltage grid (for subscribed power up to 36kVA) and was sampled at every 10-minutes. Furthermore, it came in a .csv format, which could be directly used by the Python scripts that were developed later. The retrieved period ranges from 01-10-2016 to 31-03-2017 in a total of 26208 samples. In Figure 3.1, an overview of the dataset is presented.

3.2 Load Preliminary Analysis

A preliminary analysis of the electrical load data was carried out in order to find relevant information that could be used to prepare the models. At first, from Figure 3.1, it is noticeable that the retrieved data had some noisy samples close to 30-10-2016 as the load pattern drastically changes. These points were removed after the application of a statistical test that will be detailed in the next section. Also, it appears that the load demand has a growth trend in this period. Moreover, from Figure 3.2 one can observe that the load behavior is cyclical throughout the days and that several peaks take place around 02:00, 12:00 and 23:00. In Figures 3.3 and 3.4, this pattern also seems to change depending on the weekday, with

[1] Webpage: https://data.enedis.fr/

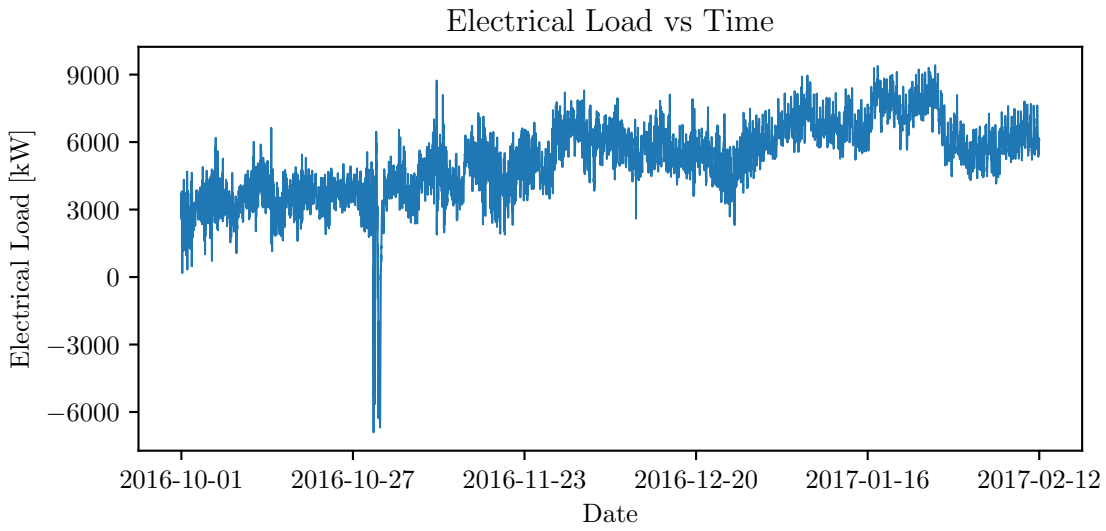

Figure 3.1: Electrical load database

lower power demand on weekends, and on the month, increasing significantly from October to January and decreasing in February.

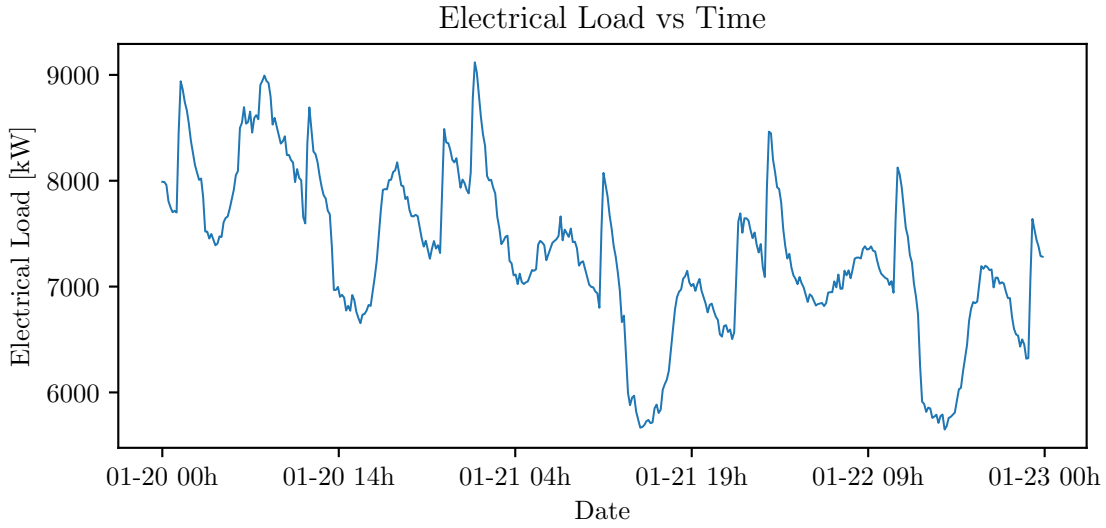

Figure 3.2: Electrical load database in detail

Figure 3.5 shows the autocorrelation function (ACF) plot for the electrical load data. This graph is frequently used to analyze time series and shows the correlation between the time series and past observations (lags). The autocorrelation coefficient ranges from -1 to 1 and measures how strong is the relationship between two variables, with 1 indicating strong positive relationship, -1 strong negative relationship and 0 meaning no relationship. The values above the blue area of the graph are those that are statistically significant ($p > 95\%$). Therefore, in this case, the plot suggests that the previous power demand is strongly related to the current one for lags up to 7 days (1008 periods). The ACF also shows that there is a descending trend of the autocorrelation coefficient as time progresses and that spikes occurs at the end of each day period (lags 144, 288, etc).

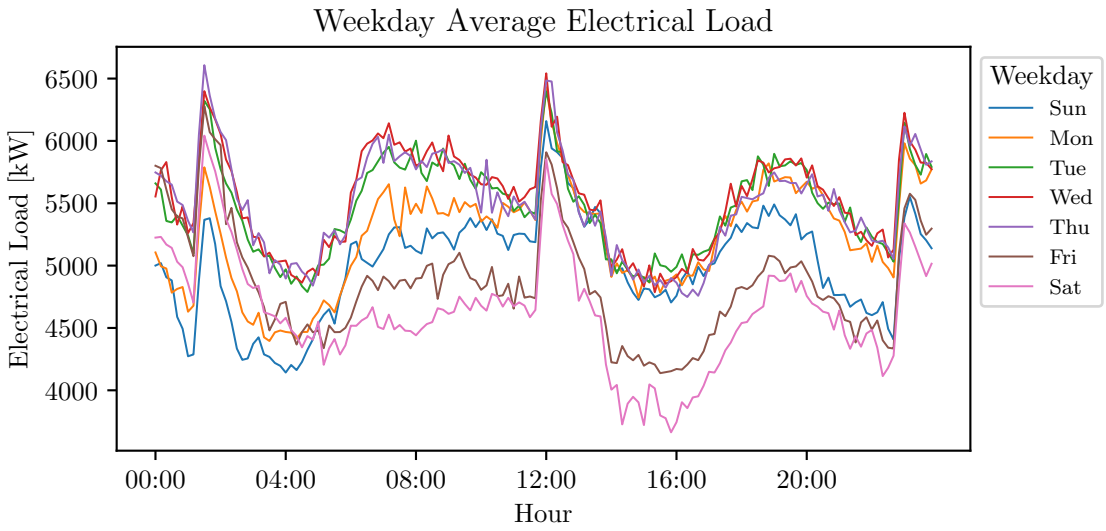

Figure 3.3: Electrical load daily average

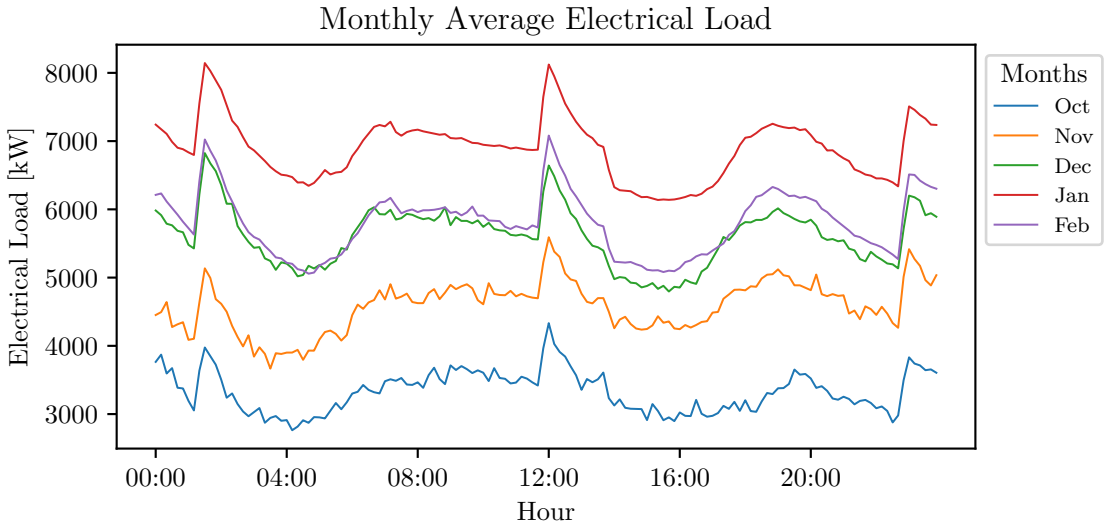

Figure 3.4: Electrical load monthly average

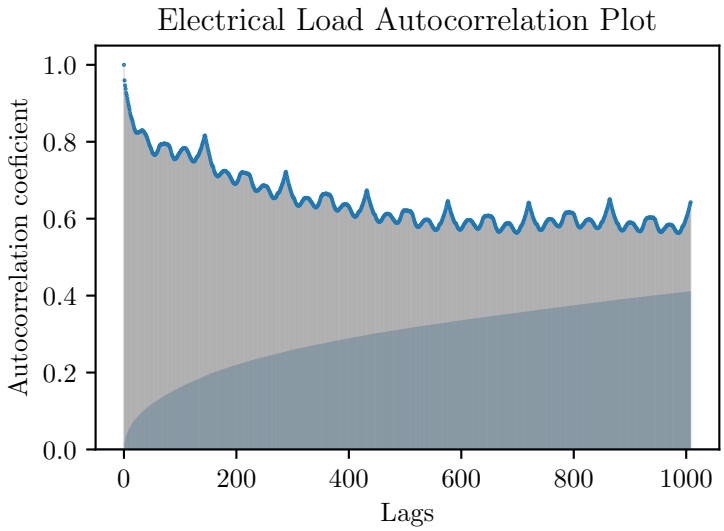

Figure 3.5: Electrical load database autocorrelation

3.3 Methodology

3.3.1 Feed-Forward Neural Networks

The term neural network comes from the fact that these models were inspired by biological brains in the sense of how they process information. A neural network model is typically composed of nodes (or neurons) that are distributed across different layers, namely input, hidden and output layers [12]. Each node in a layer is linked to the ones in the next by means of a weight parameter that measures the strength of that connection, forming a fully connected network structure that reminds the nervous system. In Figure 3.6, a general neural network is illustrated.

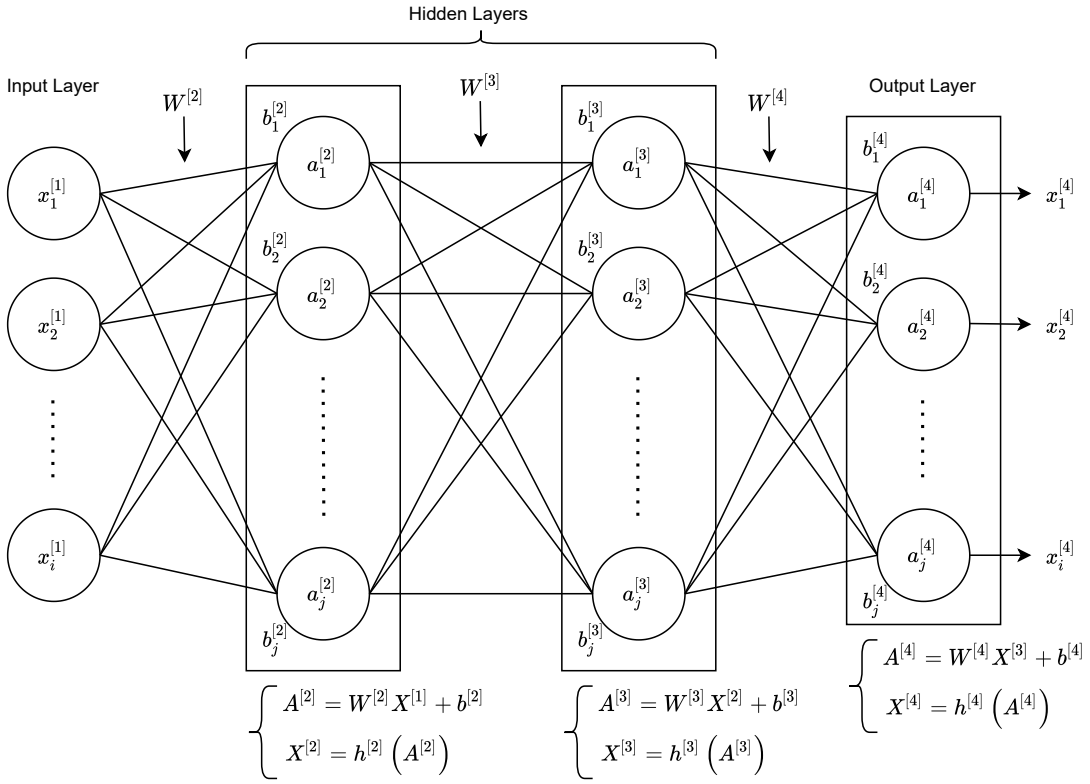

Figure 3.6: General neural network model structure

The operating principle of neural networks can be described as a sequence of functional transformations [12]. For a given layer $l \in \{1, ..., L\}$, where L is the number of layers, a quantity called the activation value $A^{[l]} = \left(a_1^{[l]}, ..., a_j^{[l]}\right)^T$ can be calculated as a linear combination of inputs $X^{[l-1]} = \left(x_1^{[l-1]}, ..., x_i^{[l-1]}\right)^T$ and weights $W^{[l]}$ in the form

$$A^{[l]} = W^{[l]}X^{[l-1]} + b^{[l]} \tag{3.1}$$

where

$$W^{[l]} = \begin{pmatrix} w_{1,1}^{[l]} & w_{1,2}^{[l]} & \cdots & w_{1,j}^{[l]} \\ w_{2,1}^{[l]} & w_{2,2}^{[l]} & \cdots & w_{2,j}^{[l]} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i,1}^{[l]} & w_{i,2}^{[l]} & \cdots & w_{i,j}^{[l]} \end{pmatrix} \tag{3.2}$$

and $b^{[l]} = \left(b_1^{[l]}, ..., b_j^{[l]}\right)^T$ is a parameter known as bias, which is used to adjust the output. The subscripts i and j represents the number of nodes or dimension of layers $l-1$ and l, respectively.

Then, the activation value $A^{[l]}$ is transformed by a nonlinear, differentiable function $h^{[l]}(\cdot)$ that is named activation function as in Equation 3.3, resulting in the next layer input vector $X^{[l]}$. For hidden layers, the activation function is generally a logistic sigmoid or hyperbolic tangent function, while for the output layer it will depend on the type of the problem, which may be the identity function for regression tasks, a logistic sigmoid for binary classification or a softmax for multiclass classification.

$$X^{[l]} = h^{[l]}\left(A^{[l]}\right) \tag{3.3}$$

The Equations 3.1 and 3.3 (also shown in Figure 3.6) present recursive calculations that constitutes a process known as forward propagation [12]. This name comes from the fact that the information is flowing forward through the network and this is the reason why this type of model is called FFNN. There are some particularities about these equations that should be mentioned: the first input vector $X^{[0]}$ comes from the features selected from the dataset, while the remaining are the result of the calculations. Also, the final result, observed in $X^{[L]}$, is the output of the model, that may be a single or multiple, continuous or discrete value depending on the problem.

As mentioned before, the parameters optimization in ML models is typically done with gradient descent based calculations. With this approach, in this case, the required partial derivatives are related to the two parameters of a FFNN model: the weights and biases. Appling the Equation 2.4 to the last layer of this model results in

$$W_{[i+1]}^{[L]} = W_{[i]}^{[L]} - \eta \frac{\partial E}{\partial W^{[L]}} \tag{3.4}$$

$$b_{[i+1]}^{[L]} = b_{[i]}^{[L]} - \eta \frac{\partial E}{\partial b^{[L]}} \tag{3.5}$$

Using the chain rule, one can write these partial derivatives as

$$\frac{\partial E}{\partial W^{[L]}} = \frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \cdot \frac{\partial A^{[L]}}{\partial W^{[L]}}^T \tag{3.6}$$

$$\frac{\partial E}{\partial b^{[L]}} = \frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \cdot \frac{\partial A^{[L]}}{\partial b^{[L]}}^T \tag{3.7}$$

where the dot $(\cdot)$ symbol stands for matrix multiplication and the circle $(\circ)$ symbol for the Hadamard or element-wise product. At this stage, it is useful to introduce the following notation

$$\delta^{[L]} = \frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \tag{3.8}$$

where $\delta^{[L]}$ is a value known as *delta* and represents the error that the layer $L-1$ sees.

Moving on to the layer $L-1$, the calculations are

$$\frac{\partial E}{\partial W^{[L-1]}} = \frac{\partial A^{[L]}}{\partial X^{[L-1]}}^{T} \cdot \left(\frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \right) \circ \frac{\partial X^{[L-1]}}{\partial A^{[L-1]}} \cdot \frac{\partial A^{[L-1]}}{\partial W^{[L-1]}}^{T} \tag{3.9}$$

$$\frac{\partial E}{\partial b^{[L-1]}} = \frac{\partial A^{[L]}}{\partial X^{[L-1]}}^{T} \cdot \left(\frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \right) \circ \frac{\partial X^{[L-1]}}{\partial A^{[L-1]}} \cdot \frac{\partial A^{[L-1]}}{\partial b^{[L-1]}}^{T} \tag{3.10}$$

$$\delta^{[L-1]} = \frac{\partial A^{[L]}}{\partial X^{[L-1]}}^{T} \cdot \delta^{[L]} \circ \frac{\partial X^{[L-1]}}{\partial A^{[L-1]}} \tag{3.11}$$

From layer $L-1$ to the first layer, it is possible to write the next deltas as

$$\delta^{[l]} = \frac{\partial A^{[l+1]}}{\partial X^{[l]}}^{T} \cdot \delta^{[l+1]} \circ \frac{\partial X^{[l]}}{\partial A^{[l]}} \tag{3.12}$$

and, therefore, the partial derivatives can be obtained as follows:

$$\frac{\partial E}{\partial W^{[l]}} = \delta^{[l]} \cdot \frac{\partial A^{[l]}}{\partial W^{[l]}}^{T} \tag{3.13}$$

$$\frac{\partial E}{\partial b^{[l]}} = \delta^{[l]} \cdot \frac{\partial A^{[l]}}{\partial b^{[l]}}^{T} \tag{3.14}$$

Finally, the parameters updates can be performed using equations 3.13 and 3.14 with 3.4 and 3.5 respectively. The process presented above constitutes the learning algorithm for neural networks and is known as backpropagation. With this approach, the algorithm goes through each layer in reverse, measuring the error contribution from each connection by means of the deltas and updating the parameters accordingly [11]. By computing the gradient in reverse, the backpropagation algorithm avoids unnecessary calculations as it reuses previous ones. This is the major reason of this method higher computational effectiveness when compared to numerical methods such as finite differences [12] and one of the cornerstones to FFNN popularity.

3.3.2 Load Forecast

Preliminary tests using different ML models such as SVM, RF and FFNN demonstrated a better forecast performance of the latter. Thus, only FFNNs were further investigated in this work. The models were created with Python's TensorFlow library, together with scikit-learn, numpy and others for three different forecast horizons: 10-minutes ahead, 1-hour ahead and 12-hours ahead. For each horizon, six different inputs combinations were tested. They were divided into two categories, one considering only the previous recorded loads and the other also including the sine and cosine transformations of time. The choice of these inputs were based on the preliminary analysis (see Section 3.2) and did not include other possibly explanatory variables such as the weekday or month to keep the models simpler and to verify if the selected inputs would be able to reach good results. Therefore, the configurations were:

A. Inputs without time:

    1. Loads at the previous six (6) periods;

2. Loads at the previous three (3) periods and three (3) days at the time of the forecast;
3. Loads at the previous three (3) periods, three (3) days and three (3) weeks at the time of the forecast;

B. Inputs with time:

1. Loads at the previous six (6) periods and time of the forecast (transformed into sine and cosine);
2. Loads at the previous three (3) periods, three (3) days at the time of the forecast and time of the forecast (transformed into sine and cosine);
3. Loads at the previous three (3) periods, three (3) days and three (3) weeks at the time of the forecast and time of the forecast (transformed into sine and cosine).

In a first moment, 20% of the data was separated (≈5000 samples) to be tested by the initial electrical load model while the other 80% was used as the training set. In order to have more reliable results, the models were trained with 3-fold cross validation. Before feeding the data into the models, the Z-score, also called standard score, of each sample in the training set was calculated as in Equation 3.15 and those that were more than 3 times the standard deviation away from the set mean were removed ($Z_{score} > 3$). In Equation 3.15, x is the sample value, μ stands for the set mean and σ represents its standard deviation. Later, the training data was scaled to be between 0 and 1. Data scaling is a common pre-processing technique which can improve the model learning process. After the training, the models were ranked based on the average of the validation set root mean squared error (RMSE) over the 3 folds. Also, the MAE was another metric used to evaluate the performances.

$$Z_{score} = \frac{(x - \mu)}{\sigma} \tag{3.15}$$

To find the best parameters for the FFNN, a grid search was performed. In an initial search, the number of nodes varied from 12 to 48 in steps of 12 with hyperbolic tangent (tanh) or rectified linear unit (ReLU) activation function (Equations 3.16 and 3.17). This resulted in a total of 8 models per input configuration that are presented in Table 3.1. At this stage, some parameters were kept constant, such as the number of layers (3), the LR (0.01), the number of maximum epochs (1500) and batch size (144). The chosen optimizer was the Adam algorithm, which is a stochastic gradient descent method based on adaptive estimation of the first and second-order moments [13]. Also, to avoid overfitting, they were programmed to stop the training if the validation MSE did not decrease between five consecutive epochs, an approach known as early stopping. After this step, the model with the best overall performance for each time horizon was further explored by changing the LR to 0.005 and to 0.001. Finally, the best configuration had its results compared to that of the persistence model in the 10-minutes horizon and to a linear regression of the same inputs in the 1-hour and 12-hour horizons.

$$tanh\,(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.16}$$

$$f(x) = max(0, x) \tag{3.17}$$

Table 3.1: Electrical load models

Model	Nodes in the hidden layer	Activation function
1	12	ReLU
2	24	ReLU
3	36	ReLU
4	48	ReLU
5	12	tanh
6	24	tanh
7	36	tanh
8	48	tanh

3.3.3 Error Forecast

A model to predict the errors of the initial electrical load forecasting model was developed for the three horizons. The data used in this step was provided by the errors observed on the test set, computed as the difference between the initial model forecasts and the actual values. After analyzing the errors and based on the error forecast methodology proposed in [44], three different input configurations were created:

A. Errors at the previous four (4) periods;

B. Errors at the previous four (4) days at the time of the forecast;

C. Errors at the previous four (4) periods and four (4) days at the time of the forecast.

The database, which, at this point, was the initial model test set, was again separated into train and test sets, scaled, trained and ranked in the same fashion as for the initial model. The major differences were on two fixed parameters: the LR was initially fixed on 0.005 and the batch size was 50 instead of 0.01 and 144 (therefore, same models as in table 3.1). The choice to reduce the batch size is due to the size of the dataset, which is smaller in this case, and, therefore, the frequency of the weights updates can be increased and lead to better performances without increasing the computational time. The other parameters were kept the same. Moreover, the best configuration had its LR adjusted to 0.0025 and 0.001 in order to look for further improvements. Lastly, the adjusted model was created by adding the predicted errors to the initial forecasts and the results were compared to the same baseline models as before but, at this time, considering only the samples in the error test set.

Figure 3.7 shows a diagram of the proposed methodology. As described in Subsection 3.3.2, the initial electrical load forecasts were made. By calculating the errors of the initial predictions, a new model was built to forecast the initial errors. Finally, the initial load and the errors forecasts were combined to generate an updated electrical load estimate.

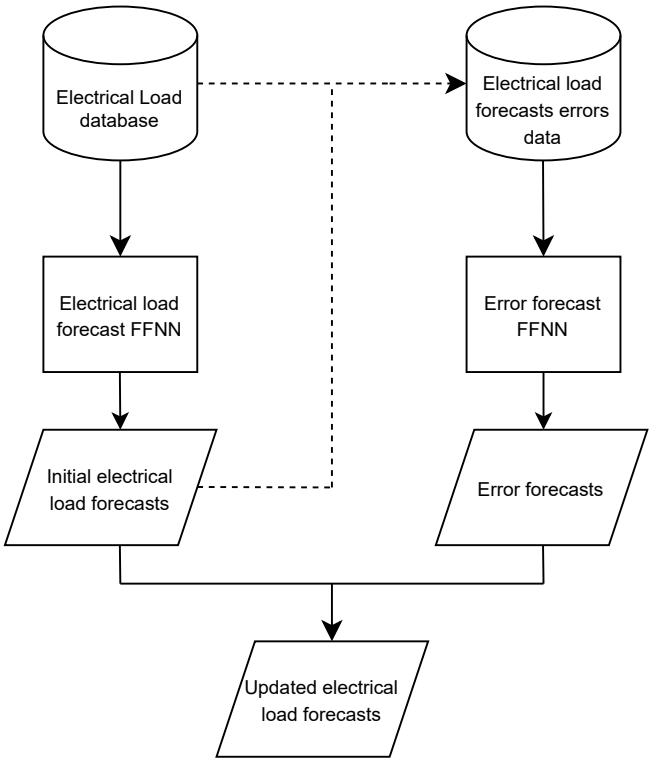

Figure 3.7: Electrical load forecast methodology

Chapter 4

Wind Speed and Power Forecasting Models

In this Chapter, the database used to forecast wind speed and power and adjust the wind power curve is introduced in Section 4.1. In Section 4.2, the observations of the preliminary analysis of this data are presented. Finally, the proposed methodology is described in Section 4.3, with the description of the initial and the error models for the wind speed forecast, wind power curve modeling and power prediction.

4.1 Database

The database for the wind speed and power forecasts consisted of NWP data from the European Centre for Medium-Range Weather Forecasts (ECMWF[1]) and historical data from the SCADA system of three wind turbines in a wind farm in the south of South America.

The ECMWF provides global forecasts from medium (≈15 days) to long (up to 3 months) range for atmospheric and ocean-wave components at different horizontal and vertical resolutions. In this work, the Atmospheric Model High Resolution 10-day forecast (HRES) was used[2]. This is a physical model that has a 9km x 9km resolution and runs twice a day, at 00:00h and 12:00h UTC (reference time), for the following 10 days. However, only 24 forecast hourly steps (steps 1 to 24) data from 31-12-2018 at 12:00h UTC to 12-12-2020 at 12:00h UTC were retrieved.

With this data, two different configurations were tested: one with 12 forecast hourly steps data in the above-mentioned period at both reference times and the other with 24 forecast hourly steps data only at the 12 UTC reference time. By doing so, it was assured that every timestep in this period had only one forecast. The reason to do this was to enable the models to use the most recent data to make forecasts, with the first being used for 12-hour and the second for 24-hour ahead forecasts.

[1] Webpage: https://www.ecmwf.int/

[2] More details in: https://confluence.ecmwf.int/display/FUG/HRES+-+High-Resolution+Forecast and https://www.ecmwf.int/en/forecasts/datasets/set-i

The data collected in this study was the U and V components of wind (IDs 131 and 132, respectively), which are the components parallel to the x and y-axis respectively, at the wind farm location and at pressure level 133 (equivalent to a geopotential and geometric altitude of 106.54m), as this height was the closest to the hub height. The data was retrieved using ECMWF Meteorological Archive and Retrieval System (MARS[3]) coupled with Python scripts. They came in .GRIB format which was exported to .csv files afterwards. Finally, the forecasted wind speed was calculated from the U and V components of wind.

The SCADA is a system that connects software and hardware elements of a wind turbine/farm in a single control system which collects, process and displays all measured data, allowing the operator to monitor turbine/farm conditions in real time. These data come from sensors and controllers installed in every subassembly of a wind turbine and it is usually sampled every 10-minutes, providing the average, maximum, minimum and standard deviation of the period. Among the recorded parameters are production status, such as active and reactive power; electrical data, such as voltages and currents; generator data, such as generator speed and bearing temperature; and environment data, such as wind speed and direction. In this work, some production status parameters, average wind speed and power data recorded from 01-01-2019 to 10-12-2020 of two turbines was used (referred as Turbines A and B), in a total of 17040 samples. Also, a shorter period from 01-01-2019 to 30-04-2019 of a third turbine was tested (named Turbine C) which resulted in 2880 samples. This data was originally sampled in a 10-minute basis and, for the wind speed forecasts, it was resampled to an 1-hour basis to match the sample rate of the NWP data.

In Figure 4.1, a fraction of the database with the measured wind speed and both 12 and 24 forecast hourly steps for Turbine A is shown. One can notice that the NWP wind speed is the same for both databases during 12 hours of each day. Furthermore, in Figure 4.2, the hourly wind speed and power averages for Turbine A in 2019 are presented.

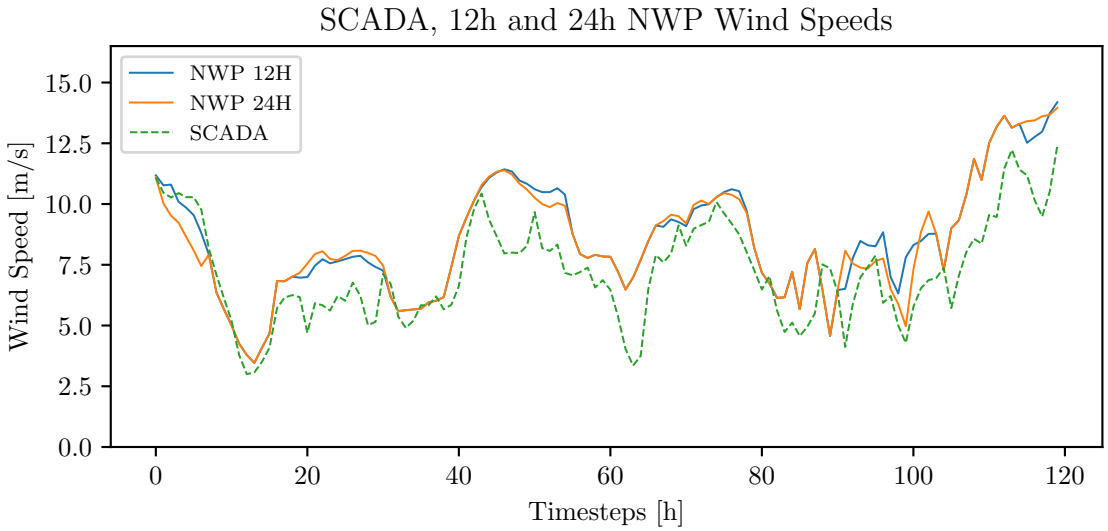

Figure 4.1: SCADA, 12h and 24h Wind Speed for Turbine A

[3]More details in: https://confluence.ecmwf.int/display/UDOC/MARS+user+documentation

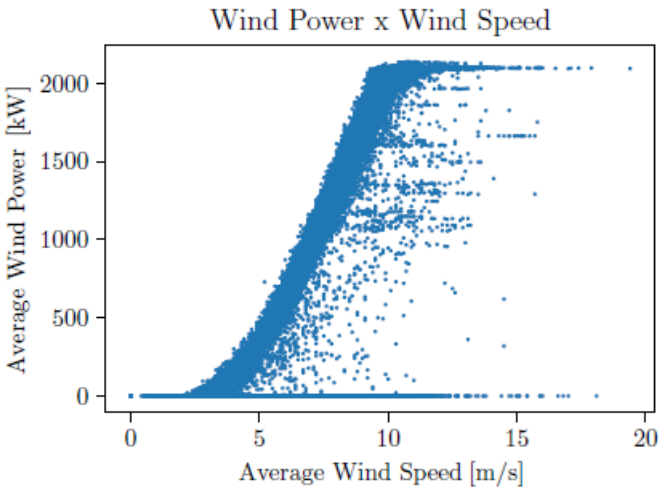

Figure 4.2: Wind Speed and Power for Turbine A

4.2 Wind Data Preliminary Analysis

In this Section, a preliminary analysis of the initial NWP predictions, wind speed and power characteristics recorded by Turbine A SCADA system is presented. They were carried out in order to better understand the available data and help to build the forecasting models. Also, it should be mentioned that these analysis were performed for the three turbines, although the results discussed here refer to only one of them due to space limits.

Typically, the wind speed distribution can be approximated by a Weibull distribution [45]. Equations 4.1 and 4.2 show the Weibull probability density function (PDF) and cumulative distribution function (CDF) respectively, where u is the wind speed in m/s; k is the shape parameter (unitless) which usually ranges from 1 to 3 with higher values representing more variable winds; and c is the scale parameter which is given in m/s and is proportional to the mean wind speed. The PDF shows the probability of a given wind speed occur while the CDF indicates the likelihood of a speed greater than a certain value.

$$f(u) = \frac{k}{c} \cdot \left(\frac{u}{c}\right)^{k-1} \cdot exp\left(-\left(\frac{u}{c}\right)^{k}\right) \tag{4.1}$$

$$F(u) = exp\left(-\left(\frac{u}{c}\right)^{k}\right) \tag{4.2}$$

Figures 4.3 and 4.4 show the Weibull distribution of Turbine A and of the 12 hour NWP. They were obtained using data only from the year of 2019 so that a full year period was considered. Regarding the distribution parameters, it is possible to notice that the shape parameters are very similar while the mean wind speed of the NWP output is about 12.8% higher than the observed wind speed (7.272 vs 6.445 m/s). This difference could be caused by the discrepancy between the model and turbine hub height or indicate that the forecasts are biased in someway.

In Figure 4.5, the monthly average wind speed of the SCADA and NWP data is shown. Again, one can visualize that the NWP forecasts presented averages higher than the observed wind speed.

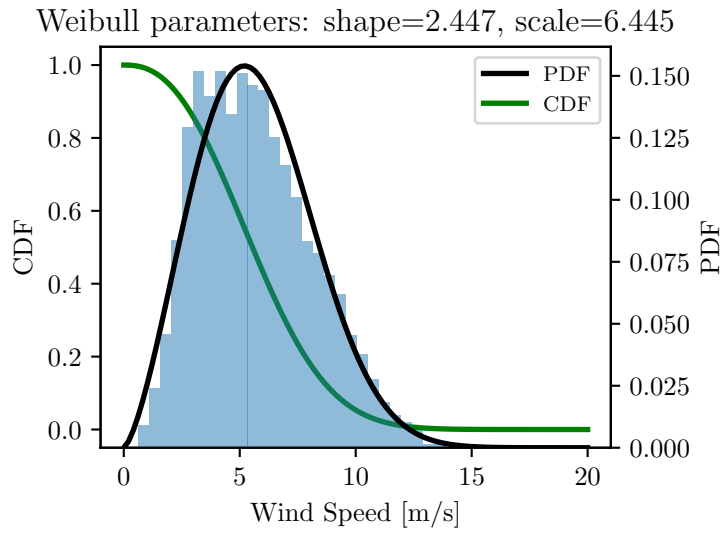

Figure 4.3: Turbine A Weibull Distribution

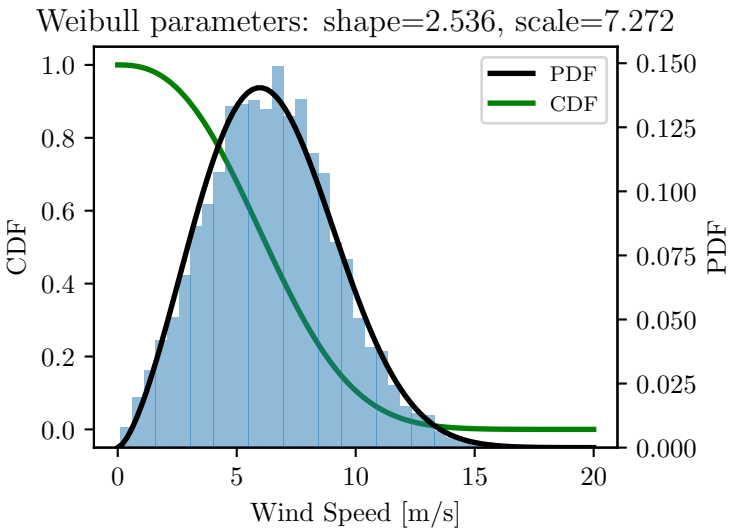

Figure 4.4: 12 hour NWP Weibull Distribution

Furthermore, the wind seasonality can also be seen, with summer months (October to March) having lower values compared to the winter months (April to September).

With regard to the wind power, it is possible to notice in Figure 4.2 regions where the turbine behavior changes and the power output is not the expected one for the observed wind speed, i.e. do not follow the wind power curve. First, there are plenty of points that recorded a power output equal or very close to zero regardless of the observed wind speed. Among the reasons for this to happen are: turbine maintenance, faults and/or grid restrictions that makes it unavailable at some periods. Another apparent deviation can be observed in regions close to an output power of 1100 and 1600 kW. In these areas, the turbine output seems to be limited somehow. These constraints are most likely related to grid restrictions that curbs the output power. Finally, several scattered points can be verified. These points are probably a consequence of the timesteps where a restriction of any kind was imposed to the turbine and, considering that the SCADA system provides consolidated results, they reflect this different behavior. Also, they can be outliers or represent measurement or aggregation errors. All of these samples

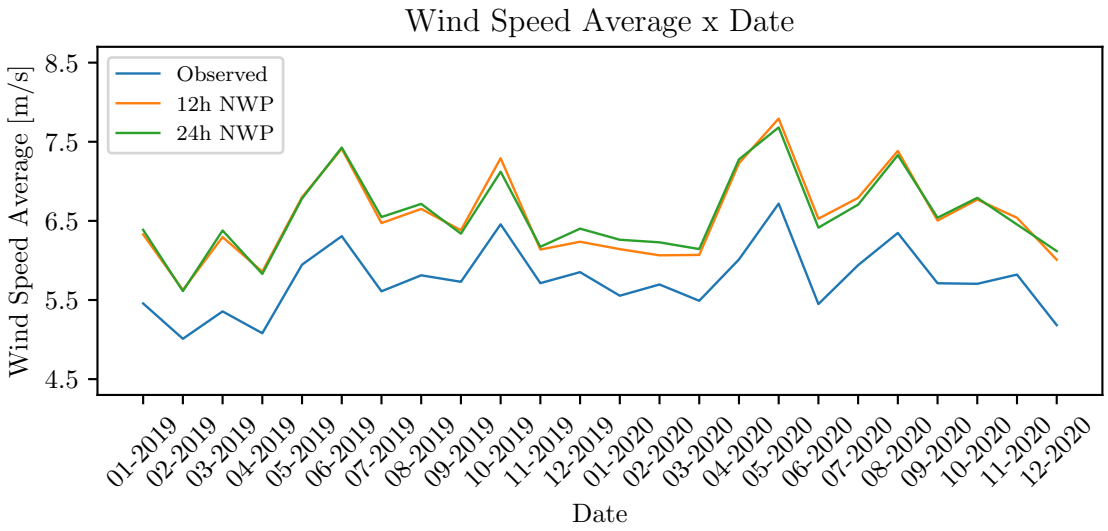

Figure 4.5: Turbine A monthly wind speed average

that can indicate that the turbine is not behaving properly can hinder the performance of wind power forecasting methods as they do not represent the optimal conditions and, therefore, should be carefully handled.

4.3 Methodology

4.3.1 Wind Speed Forecast

In order to make wind speed forecasts, a FFNN was implemented using Python TensorFlow library, together with scikit-learn, numpy, pandas and others. The details of a FFNN model were presented in Subsection 3.3.1. As mentioned before, wind speed forecasts were made for two different horizons: 12 and 24-hour ahead. In every case, the inputs were forecasted wind speeds by the NWP model and the output (target) was the SCADA wind speed average. For the first horizon, three different inputs combinations were tested, while for the second, only configuration B was tested, as it had the best results in the 12-hour horizon. The following inputs were chosen based on the analysis of the correlation of the predicted NWP wind speeds and the target value:

A. NWP at t=0, 12, 24, 36 (4 inputs) — Models PAn.12;

B. NWP at t=0, 1, . . . , 11 (12 inputs) — Models PBn.12 and PBn.24;

C. NWP at t=0, 1, . . . , 5 (6 inputs) — Models PCn.12;

The acronyms for the models were created by taking its abbreviations, i.e. NWP, ANWP or P, together with the forecast horizon (12 or 24-hour) and with the turbine (TA, TB or TC). In the case of the proposed model, the neural network configuration reference is also present. As an example, the model PA3.12.TA refers to the proposed model that used 4 inputs (PA) with neural network configuration number 3 for the 12-hour horizon for turbine A.

With regard to data preparation, at first 20% of the data was separated to evaluate the final model (test/holdout set) while the remaining was used as the training set. The model was trained with 5-fold cross validation so that the results of each one were more reliable. Before the training of each fold, training data was scaled to be in a range between 0 and 1. The fitted scaler was then applied to the validation set to prevent leakage of information at the training step. After training the models, they were ranked based on the average RMSE of the 5 folds on the validation set. Also, the MAE was used to assess model's performance. Finally, the best model was compared to the baseline models and, at this stage, the R2 metric was also considered.

For every configuration, a grid search was performed so that the best parameters for the FFNN could be found. The searched parameters were the number of nodes in the hidden layer and the LR of the algorithm. As there is no consolidated technique to choose these parameters, the following strategy was adopted: the number of nodes ranged between the input layer size and 3 times this value and the LR was exponentially decreased from 0.01 to 0.00001. The configuration of each model and the values of these parameters can be found in the Appendix A. Following preliminary investigation, all other parameters were kept constant for all models, such as activation function (rectified linear unit), number of layers (3), maximum epochs (50) and batch size (32). The chosen optimizer was the Adam algorithm [13]. To avoid overfitting, all the models were programmed to stop the training if the validation MSE did not decrease between three consecutive epochs. After finding the best model, the holdout set was then evaluated and the results were compared with two baseline models: the initial NWP forecast and the NWP model adjusted by the mean error (ANWP).

The grid search was done only for one of the three turbines (Turbine A) as it is an extremely time and memory consuming procedure which was conducted in a regular personal notebook with limited computing resources. Therefore, for the other two turbines, only the top-3 models for Turbine A were tested and appraised.

4.3.2 Error Forecast

The error forecast for the wind speed model was performed only for Turbine A in the 12-hour horizon, as the initial models were proposed for this condition. The data for the error forecasting model was obtained from the test set of the wind speed correction model by calculating the difference between the measured wind speed and the forecasts made by this model. Based on the error analysis and on the error forecast methodology presented in [44], an input vector consisting of the previous 12 observed errors and the following 12 forecasts was created to predict the next 12 errors (target vector), in a multioutput neural network approach.

To prepare the data to be trained, the database was separated into train and test sets, scaled, trained and ranked in the same fashion as in the wind speed forecasting model (see Subsection 3.3.3). The major differences were on the searched and fixed parameters. In this case, the number of nodes tested were 20, 24 and 30; the LR was either 0.0001 or 0.00001; and the batch size was 1, 10 or 32. The choice to change the batch size is due to size of the dataset and the frequency of the weights updates,

which are smaller and higher respectively and can counterbalance in terms of training time. The other parameters such as activation function (hyperbolic tangent), number of layers (3), maximum epochs (50) and chosen optimizer (Adam) were kept constant. The models configurations can be found in the Appendix A. Lastly, the adjusted wind speed was obtained by adding the predicted errors to the initial forecasts and the results were compared to the baseline models (applied to the error test set instead).

4.3.3 Wind Power Curve Modeling

Similarly to the error forecast, the wind power curve was obtained only for Turbine A but, here, only the SCADA data for 2019 was considered. A typical wind power curve can be divided into four different operating regions, as illustrated in Figure 4.6. In region I, the wind speed is lower than the minimum required for the turbine to operate (cut-in speed) and, therefore, the output power is zero. Likewise, in region IV, the wind speed is higher than the supported maximum (cut-out speed) and, thus, the turbine is disconnected. In region III, the wind speed is between the rated and the maximum speeds and the output power is the rated power. Finally, in region II, the wind speed is between the cut-in and rated speed and the output power is expected to be proportional to the cube of the wind speed. Therefore, the power curve modeling performed in this work refers only to region II. In addition to that, as mentioned in Section 3.2, there are plenty of points that do not represent a typical power curve. Thus, it was necessary to clean the dataset before fitting any curve. To do that, several conditions regarding multiple SCADA parameters were used to filter the measurements in a first moment. The points that did not follow the conditions below over a 10-minutes period were removed from the dataset.

- Turbine brake active for less than 5 seconds;
- Turbine controller active for more than 595 seconds;
- Turbine connected to the grid for more than 595 seconds;
- Turbine system 'Not OK' message for less than 5 seconds;
- Turbine without any abnormal event code;
- Turbine average active power equal to or higher than zero;
- Measured Wind speed between cut-in and rated speeds;
- Difference between yaw alignment and wind direction less than 10°.

After removing these data, it was verified that several non-representative points were still in the dataset. Hence, another approach to filter those points was needed. The chosen solution was the Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which is a clustering algorithm that aims to separate high density from low density areas of the dataset [46]. The working principle of this method is based on grouping together points that are sufficiently close to each other (neighbors samples). To do that, two parameters must be controlled: a distance measurement, which can come from any distance function, and a minimum number of samples to constitute a cluster. DBSCAN differs

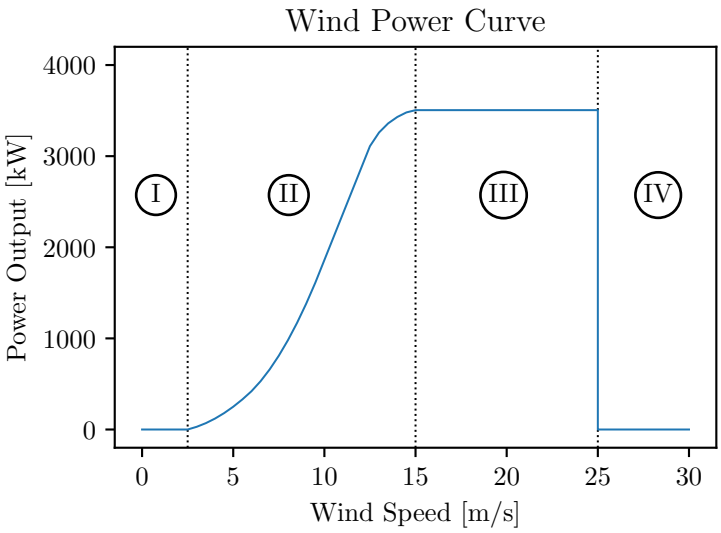

Figure 4.6: Typical Wind Power Curve

from other clustering algorithms, such as K-Means, as it does not require the number of clusters to be previously defined. Also, it allows the identification of outliers and works only with two parameters, which makes it simple to use. On the other hand, it does not work well with datasets that have different densities distributions [47].

A sample of the dataset can be categorized into three types: core, border or noise samples. The core samples are those that have at least n points within the minimum distance ϵ from it. The border samples are those that have at least one core sample within distance ϵ from it. The noise samples are those that are neither core nor border samples. Finally, the clusters are composed by the core and border samples. Algorithm 1 presents the pseudocode of the DBSCAN algorithm. One can notice that the neighborhood queries are executed only for points that are not labeled yet and those are labeled right after the query. This assures that one neighborhood query will be perfomed exactly one time for each sample in the dataset [48]. Furthermore, a sample can only be relabeled if it was previously classified as noise. In addition to the pseudocode, in Figure 4.7, an illustration of how the DBSCAN classify samples is shown, where the red points are core samples, the green are border samples and the purple are noise samples.

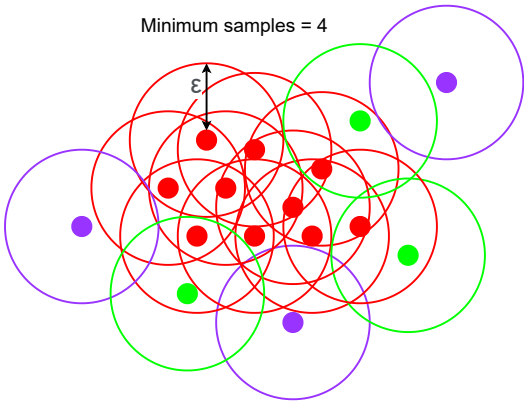

Figure 4.7: DBSCAN example (based on [48])

In this work data, as the wind speed x and power y axis values had different ranges, a scaler was

Algorithm 1: Pseudocode of the DBSCAN algorithm (based on [48])

Input: DB: Database
Input: $distFunc$: Distance function
Input: ϵ: Minimum distance
Input: $minPts$: Minimum points
Data: $label$: Point label, initially $undefined$
foreach *point* p **in** DB **do** // Iterate over the whole database
  if $label(p) \neq undefined$ **then continue** // Skip processed points
  Neighbors $N \leftarrow RangeQuery(DB, distFunc, p, \epsilon)$ // Find initial neighbors
  if $|N| < minPts$ **then** // Core point check
    $label(p) \leftarrow Noise$ // Non-core points are labeled as noise
    continue
  $c \leftarrow$ next cluster label // Start new cluster
  $label(p) \leftarrow c$ // Label point p
  Seed set $S \leftarrow N \setminus \{p\}$ // Expand neighborhood
  foreach *point* q **in** S **do** // Iterate over the every neighbor q of p
    if $label(q) = Noise$ **then** $label(q) \leftarrow c$ // Changes from noise to member of the cluster
    if $label(q) \neq undefined$ **then continue**
    Neighbors $N \leftarrow RangeQuery(DB, distFunc, q, \epsilon)$ // Find new neighbors
    $label(q) \leftarrow c$ // Label point q
    if $|N| < minPts$ **then continue** // Core-point check
    $S \leftarrow S \cup N$ // Add new neighbors to S
  end
end

applied to set the range between 0 and 1 on both axis in order to facilitate the adjustment of the distance parameter. The choice of this parameter is crucial for the method efficiency as large values can make the algorithm create a single cluster for the whole dataset while small ones can make it label most data as outliers. In addition to that, the minimum samples parameter allows to control how the algorithm will deal with noise and generally needs to be higher for very large or noisy data. The adjustment of both parameters was performed by trial and error so that the algorithm could identify the samples that seemed not to follow the expected wind speed and power relationship.

After applying the DBSCAN algorithm and removing the outliers from the dataset, it was possible to obtain the wind power curve. To do that, different polynomial curves were fitted by changing the degree from 3 to 7 and the best one chosen based on a visual analysis. Equation 4.3 presents the formulation for these curves, where a represents the coefficients and n indicates the degree of the polynomial. Finally, with the polynomial fit for region II (see Figure 4.6) and, considering the expected power output for regions I, III and IV, the wind power curve was defined.

$$y = \sum_{i=0}^{n} a_i x^i \tag{4.3}$$

4.3.4 Wind Power Forecast

In Figure 4.8, the proposed methodology for the wind speed and power forecasts is illustrated. After computing the wind speed with the methodology described in Subsections 4.3.1 and 4.3.2 and finding the wind power curve as detailed in Subsection 4.3.3, the output power was calculated by using the

estimated wind speed with the power equations. Finally, the results were compared to the ones observed using the original NWP forecasts and the SCADA wind speed with the proposed power curve.

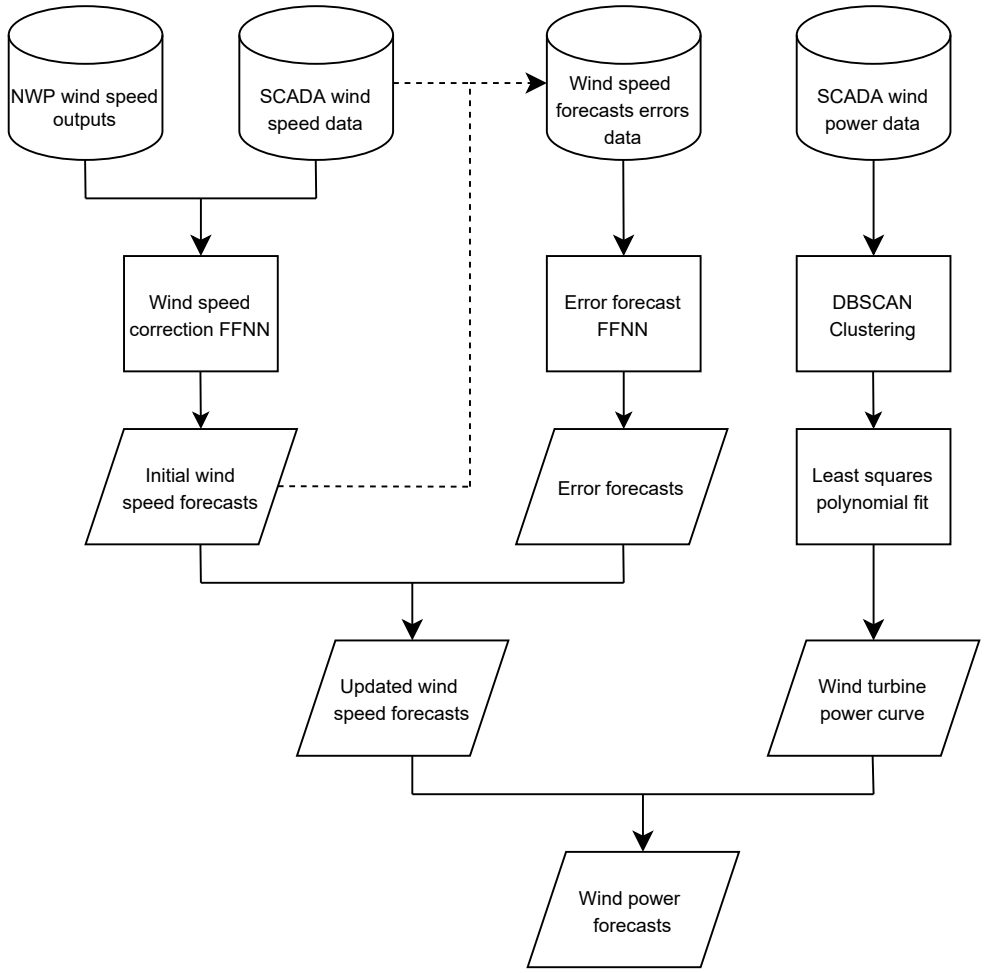

Figure 4.8: Proposed methodology

Chapter 5

Electrical Load Forecasting Results

In this Chapter, the results obtained with the electrical load forecasting models are presented as follows: Section 5.1 shows the results of the initial predictions, Section 5.2 presents the analysis of the initial models errors, Section 5.3 demonstrates the results of the error forecasting models and, in Section 5.4, the achievements of the adjusted model are discussed.

5.1 Initial Results

The initial load forecasts are presented below separately for each time horizon. The tables in this subsection present only the results of the best model for each horizon in order to make the text more fluid. The complete results can be found in the Appendix C (Tables C.1 to C.18).

10-minute horizon

The initial results obtained with the best 10-minutes ahead model are presented in Table 5.1. The configuration A2.4a, which was the original one with LR of 0.01, achieved almost the same results of the configuration A2.4b but converged considerably faster. This probably happened due to the higher LR. On the other hand, model A2.4c appeared to learn in a much slower pace and did not reached the same results. When compared to the persistence model, the RMSE and the MAE of the best proposed model were 10.6% and 23.8% higher. These findings are somehow understandable as making predictions for such a short time in the future can be easier by assuming that the next value will be the same as the one observed now. Indeed, the proposed ANN converged to a solution that generated forecasts similar to those of the persistence model (Figure 5.1). However, giving inputs other than the previous recorded value to the model seemed to hinder its predictions.

1-hour horizon

The results of the best 1-hour ahead model are shown in Table 5.2. Here, the best configuration also took the previous three periods and days as inputs, however, the arrangement with time performed

Table 5.1: Results of the 10-minutes ahead initial and baseline models

Model	LR	RMSE [kW]	MAE [kW]	Test RMSE [kW]	Test MAE [kW]	ET [s]
A2.4a	0.01	223	159.34	234.24	177.66	354.13
A2.4b	0.005	223.62	160.9	-	-	524.84
A2.4c	0.001	247.98	179.5	-	-	548.39
Baseline	-	-	-	211.72	143.53	-

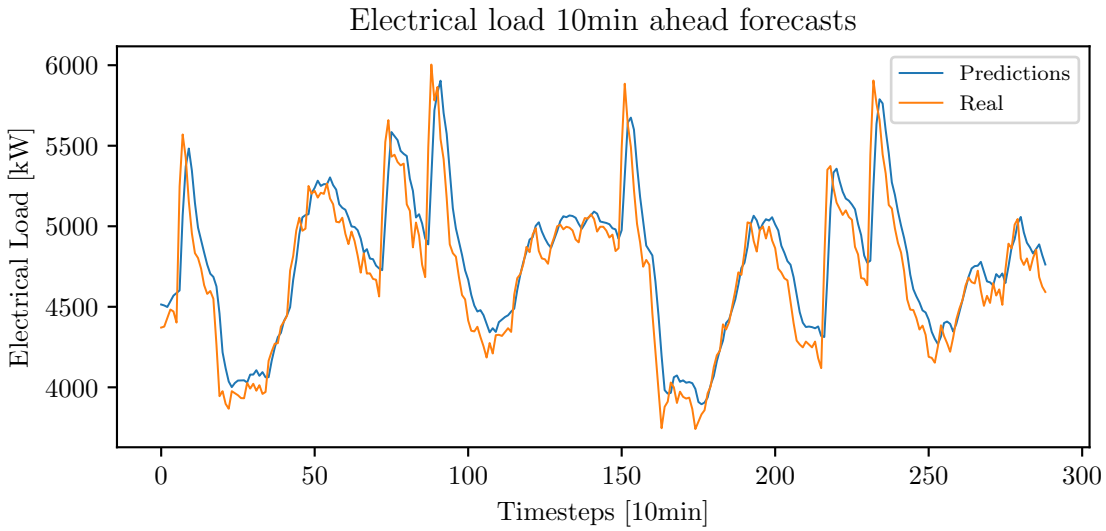

Figure 5.1: 10-minutes ahead initial forecasts

better. Similarly to the previous case, the model with the original LR had better results than the other two but with similar execution times, which suggests that the higher LR enabled the model to be closer to a loss curve minimum at the end of the training. Furthermore, it achieved superior performance than the linear regression model in the order of 10.7% over the RMSE and of 11.3% over the MAE. In Figure 5.2, the forecasts made with the ANN can be seen. It shows that the model was able to make accurate predictions but missed some of the load spikes and had a slightly smoother behavior when compared to the actual values.

Table 5.2: Results of the 1-hour ahead initial and baseline models

Model	LR	RMSE [kW]	MAE [kW]	Test RMSE [kW]	Test MAE [kW]	ET [s]
B2.8a	0.01	339.43	261.33	376.25	303.76	513.99
B2.8b	0.005	357.58	276.05	-	-	546.2
B2.8c	0.001	1037.59	736.56	-	-	515.76
Baseline	-	-	-	421.48	342.3	-

12-hours horizon

Table 5.3 presents the results of the best 12-hours ahead model. The best configuration was the same of the 10-minutes horizon with only the previous three periods and days as inputs. Here, decreasing the LR to 0.005 enhanced the forecasts when compared to the other two configurations. The lower execution time combined with the worse performance of the original configuration suggest that a LR of 0.01 was too high and the training stagnated and stopped at some point while for model A2.1c it was too

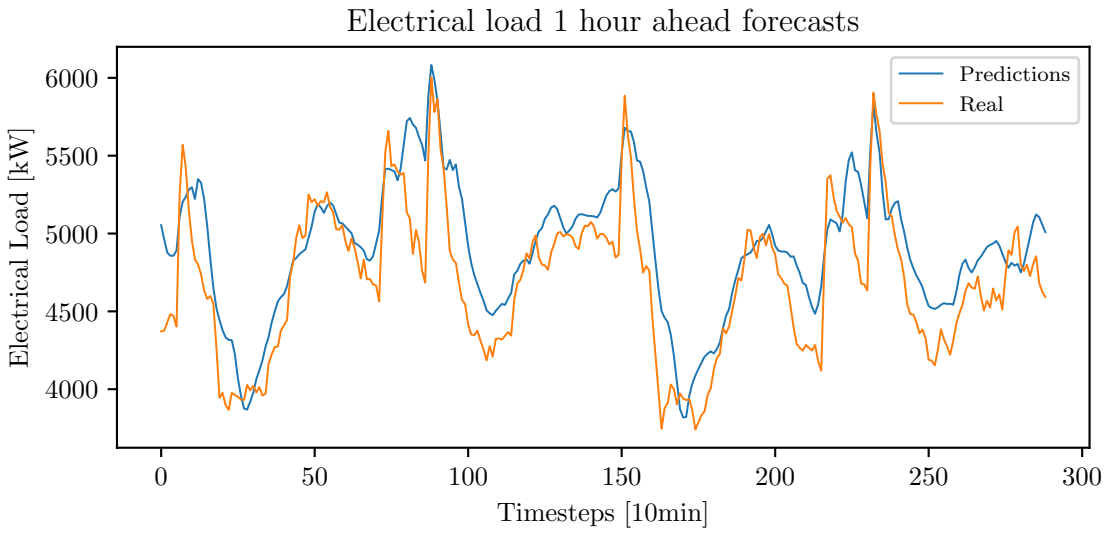

Figure 5.2: 1-hour ahead initial forecasts

low and did not allowed the results to improve sufficiently. In comparison to the linear regression model, the performance of the best proposed model was 12.6% and 13.5% superior in terms of the RMSE and the MAE. Figure 5.3 shows results that were similar to the previous case. Yet, the predictions seemed more inaccurate, with more substantial errors.

Table 5.3: Results of the 12-hours ahead initial and baseline models

Model	LR	RMSE [kW]	MAE [kW]	Test RMSE [kW]	Test MAE [kW]	ET [s]
A2.1a	0.01	548.03	432.91	-	-	184.89
A2.1b	0.005	543.55	431.31	501.46	399.92	531.64
A2.1c	0.001	647.33	520.63	-	-	507.03
Baseline	-	-	-	573.76	462.44	-

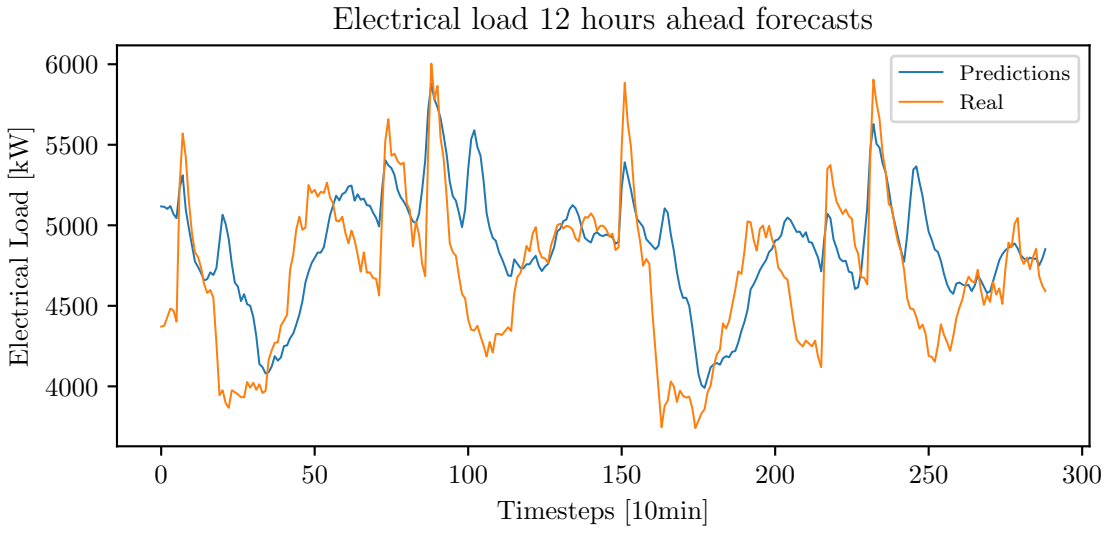

Figure 5.3: 12-hours ahead initial forecasts

Summary

An interesting observation that was valid for all the models is that often those with lower execution times had the worst performances, which can suggest that they were stuck in a local minima and were not able to progress with the chosen LR (see tables in Appendix C). Also, it seemed that the differences of the number of nodes in the hidden layer and the activation function from one model to another did not impact the execution times.

Finally, Table 5.4 summarizes the results of the proposed initial models for the three forecast horizons. These results showed that the forecast accuracy decreased as the forecast horizon increased and indicate the greater degree of difficulty to make longer forecasts due to the higher uncertainty of the data, justifying the use of more complex models. Furthermore, for very short-term forecasts, simple methods such as the persistence model can provide relatively good estimates with low computational resources.

Table 5.4: Summary of the initial forecasts in the three horizons

Horizon	RMSE [kW]	MAE [kW]
10-minutes	234.24	177.66
1-hour	376.25	303.76
12-hours	501.46	399.92

5.2 Error Analysis

Figures 5.4, 5.5 and 5.6 show the error distributions for the models proposed in Section 5.1 for the three forecast horizons. It is possible to notice that the error distribution is similar to the Gaussian distribution, especially for the 1-hour and 12-hours forecasts. Moreover, the figures also present the mean and standard deviation of these errors. The fact that the means are not centered in zero suggests that there is still some error bias and that more information could be retrieved from the data. Also, the standard deviation increases for longer forecast horizons which, again, can indicate the higher degree of uncertainty on long-term predictions.

Figures 5.7, 5.8 and 5.9 show the ACF plot for the three horizons. As explained in Section 3.2, this graph reveals the strength of the relationship between a time series and its previous values and it was used to construct the error forecasting models. One can verify that the autocorrelation coefficients rapidly decreased and showed an oscillating behavior with more significant spikes at the end of each day (lags 144, 288, 432 and 576). Therefore, configurations combining these lags and the most recent values were tested to forecast the initial models errors.

5.3 Error Forecast

The error forecasts are presented below separately for each time horizon. The tables in this subsection show only the results of the best model for each horizon in order to make the text more fluid. The complete results can be found in the Appendix C (Tables C.19 to C.27).

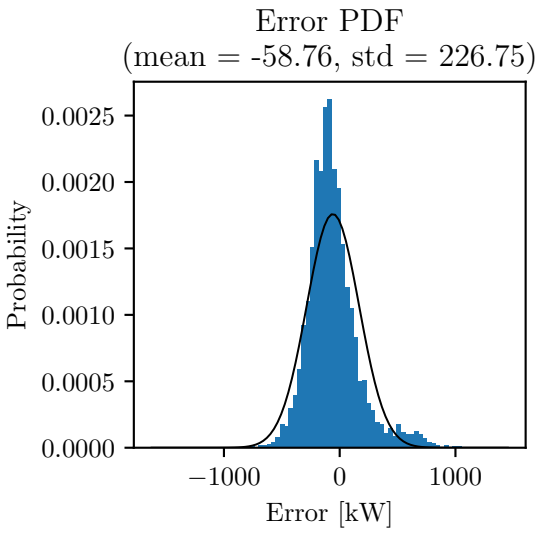

Figure 5.4: 10-minutes ahead error distribution

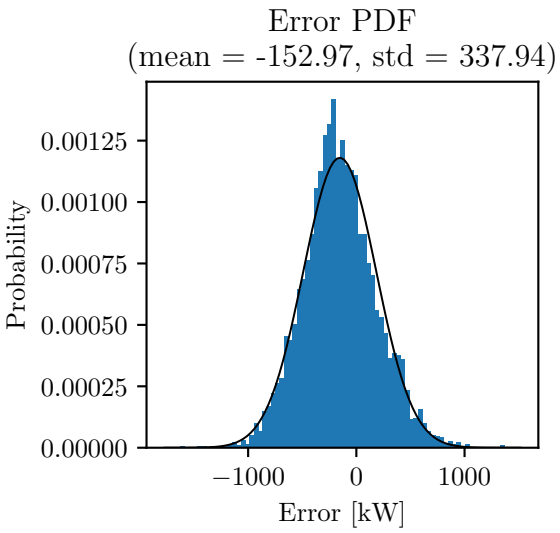

Figure 5.5: 1-hour ahead error distribution

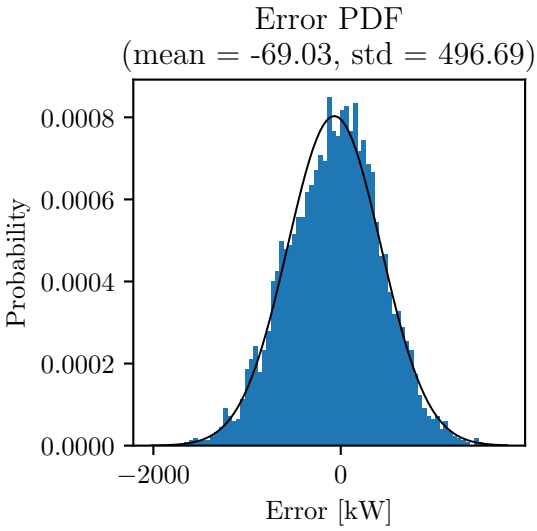

Figure 5.6: 12-hours ahead error distribution

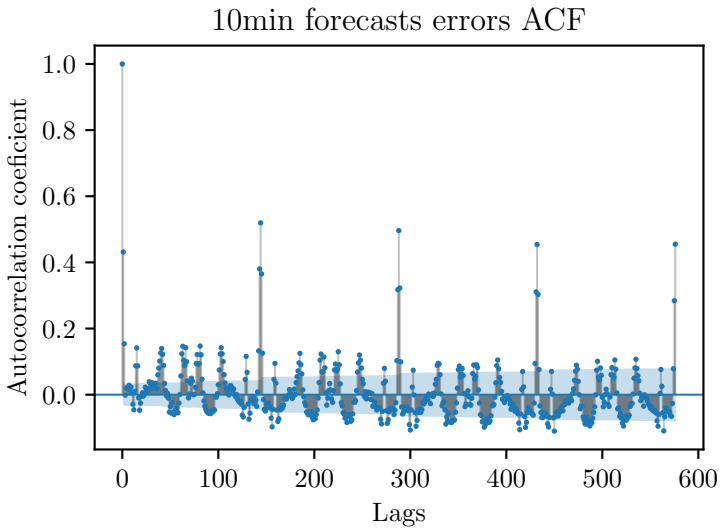

Figure 5.7: 10-minutes ahead error autocorrelation

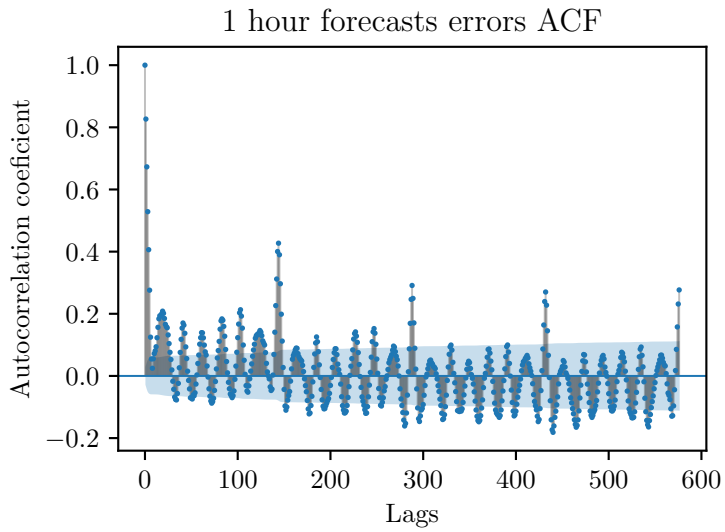

Figure 5.8: 1-hour ahead error autocorrelation

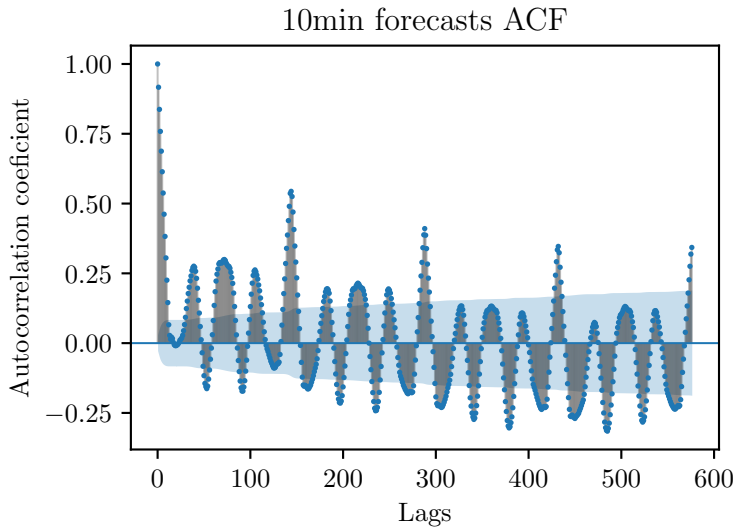

Figure 5.9: 12-hours ahead error autocorrelation

10-minute horizon

Table 5.5 shows the results of the best 10-minutes ahead error forecasting model. In this scenario, all the models had similar results in terms of error metrics. The major difference was regarding the training time, which was higher for smaller LRs. The most probable reason to these facts is that the three models were able to find the same loss function minimum but those with smaller LRs had smaller weights updates and, therefore, took longer. Figure 5.10 shows the error predictions made with model C.8b. One can notice that the predictions were very accurate, especially on the error spikes. After this step, it was expected that the error correction model would improve the initial results.

Table 5.5: Results of the 10-minutes ahead error forecasting models

Model	LR	Val. RMSE [kW]	Val. MAE [kW]	Test RMSE [kW]	Test MAE [kW]	ET [s]
C.8a	0.005	170.14	118.07	-	-	23.57
C.8b	0.0025	169.97	118.26	103.69	78.01	50.9
C.8c	0.001	172.17	119.87	-	-	96.7

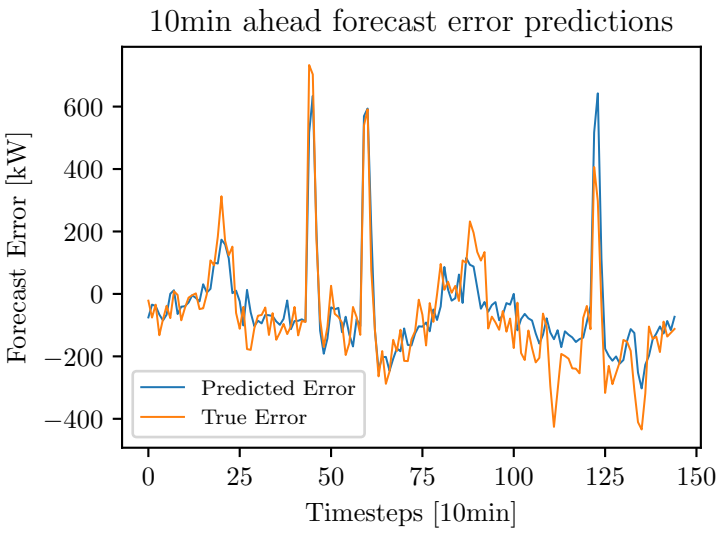

Figure 5.10: 10-minutes ahead error forecasts

1-hour horizon

In Table 5.6, the results of the best 1-hour ahead error forecasting model can be seen. The remarks made for the 10-minutes ahead error forecasting model are also valid for this case. In Figure 5.11, the forecasts made with model C.1b are illustrated. Here, the model was not as accurate as in the previous case to predict spikes and trends.

Table 5.6: Results of the 1-hour ahead error forecasting models

Model	LR	Val. RMSE [kW]	Val. MAE [kW]	Test RMSE [kW]	Test MAE [kW]	ET [s]
C.1a	0.005	284.81	230.45	-	-	16.66
C.1b	0.0025	284.43	230.86	236.54	186.57	37.54
C.1c	0.001	284.71	230.93	-	-	78.75

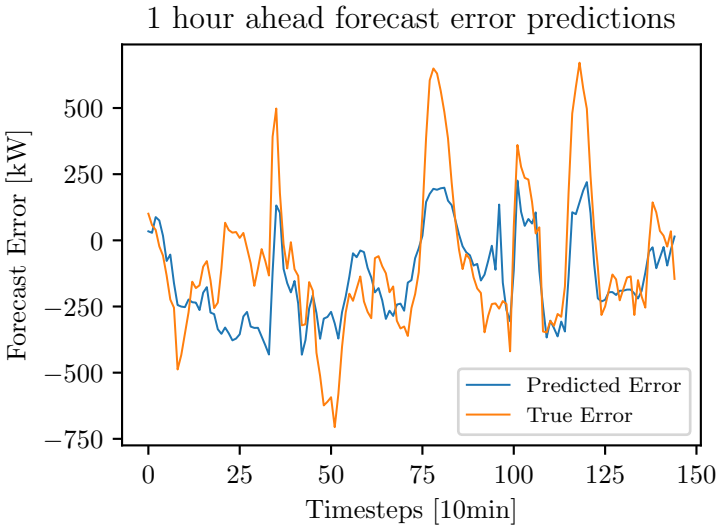

Figure 5.11: 1-hour ahead error forecasts

12-hours horizon

Table 5.7 presents the results of the best 12-hours ahead error forecasting model. In comparison to the other two scenarios, here there is a significant difference on the training time of the models. Although the results were similar, the model C.3c (with the smaller LR) was able to perform slightly better than the other two with considerably higher execution time. This was probably because the smaller LR enabled the model to continue improving despite the closeness to a loss function minimum while the other two stagnated and stopped training. Figure 5.12 shows that, as in the 1-hour horizon, the model underestimated some spikes and, also, presented smoother behavior when compared to the true error.

Table 5.7: Results of the 12-hours ahead error forecasting models

Model	LR	Val. RMSE [kW]	Val. MAE [kW]	Test RMSE [kW]	Test MAE [kW]	ET [s]
C.3a	0.005	406.92	334.34	-	-	6.83
C.3b	0.0025	407.37	334.82	-	-	9.54
C.3c	0.001	405.6	332.62	343.05	257.43	56.57

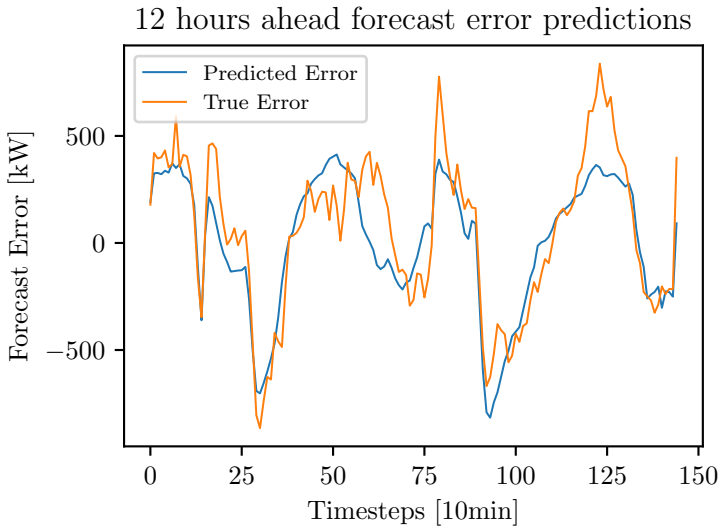

Figure 5.12: 12-hours ahead error forecasts

Summary

Unlike the initial models (see Section 5.1), here all the models in each forecast horizon presented similar performances regarding error metrics and execution times (see tables in Appendix C). In every case, the best model was the one that combined both the previous 4 periods and days as inputs. Lastly, Table 5.8 summarizes the results of the proposed error models for the three forecast horizons. Again, the forecast accuracy decreased as the forecast horizon increased which emphasize the greater degree of difficulty to make longer forecasts due to the higher uncertainty of the data. Furthermore, one can notice that both the RMSE and MAE were lower than the ones achieved with the initial models, which can possibly indicate that these models were more accurate than the initial.

Table 5.8: Summary of the error forecasts in the three horizons

Horizon	RMSE [kW]	MAE [kW]
10-minutes	103.69	78.01
1-hour	236.54	186.57
12-hours	343.05	257.43

5.4 Adjusted Results

The adjusted results are presented below separately for each time horizon. As explained in Section 3.3, the adjusted model was obtained by adding the forecasted error to the initial prediction and had its results compared to that of the baseline models considering only the error test set.

10-minute horizon

Table 5.9 shows a comparison between the results obtained with the initial, adjusted and baseline forecasting models for the 10-minutes ahead scenario. It can be noticed that the initial model performed worse than the persistence model and that, by means of the proposed methodology, it was able to outperform it. The improvements of the adjusted forecasts over the baseline was of 38.2% on the RMSE and 29.1% on the MAE. These results are illustrated in Figure 5.13, which shows that the proposed methodology apparently allowed to correct the ”persistence” behavior previously presented by the initial model and achieve more accurate forecasts.

Table 5.9: Results of the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]
Initial	200.83	152.96
Adjusted	103.69	78.01
Baseline	167.77	110.0

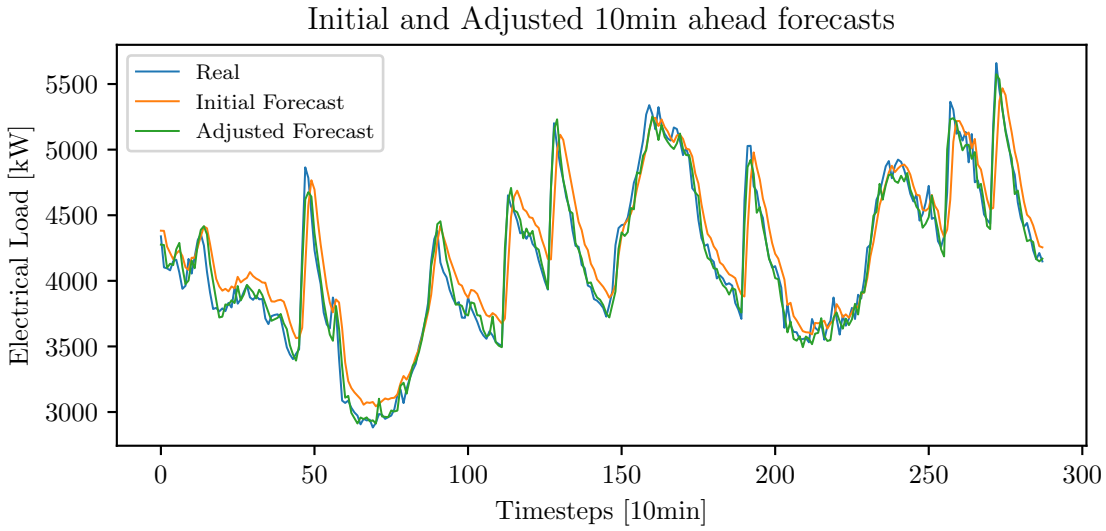

Figure 5.13: 10-minutes ahead adjusted forecasts

1-hour horizon

In Table 5.10, a comparison between the results obtained with the initial, adjusted and baseline forecasting models for the 1-hour ahead scenario can be seen. Originally, the initial model had a performance only slightly superior to the one of the linear regression model. With the proposed methodology, the initial results were improved by 32.1% on the RMSE and 36.2% on the MAE. In Figure 5.14, it is possible to notice that the model correctly adjusted the initial forecasts between timesteps 0 and 50 and close to timestep 150, while it did not change the accurate predictions of some load spikes.

Table 5.10: Results of the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]
Initial	352.95	295.43
Adjusted	239.34	188.49
Baseline	371.58	307.11

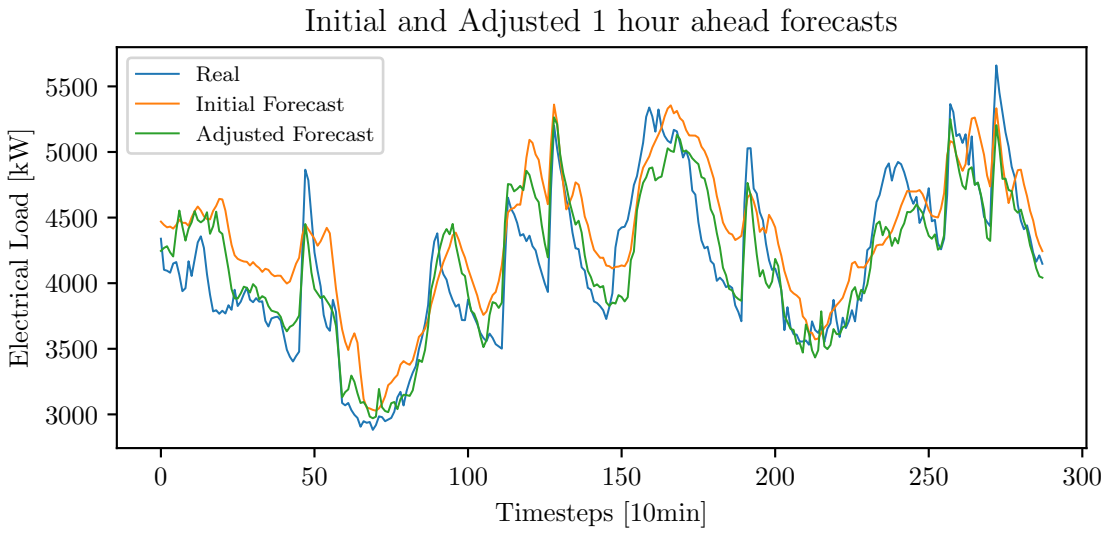

Figure 5.14: 1-hour ahead adjusted forecasts

12-hours horizon

Table 5.11 presents a comparison between the results obtained with the initial, adjusted and baseline forecasting models for the 12-hours ahead scenario. Like in the previous case, the proposed methodology was able to enhance the initial forecasts, which were only slightly better than that of the linear regression model. Here, the improvements over the initial model were of 22.5% on the RMSE and 29.0% on the MAE. Figure 5.15 shows the results of the initial and adjusted forecasts. It can be seen that, again, the proposed methodology to properly correct the forecasts close to the timesteps 50 and 150 while it did not affect the initial predictions on the load spikes.

Table 5.11: Results of the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]
Initial	442.55	362.55
Adjusted	343.05	257.43
Baseline	480.68	381.89

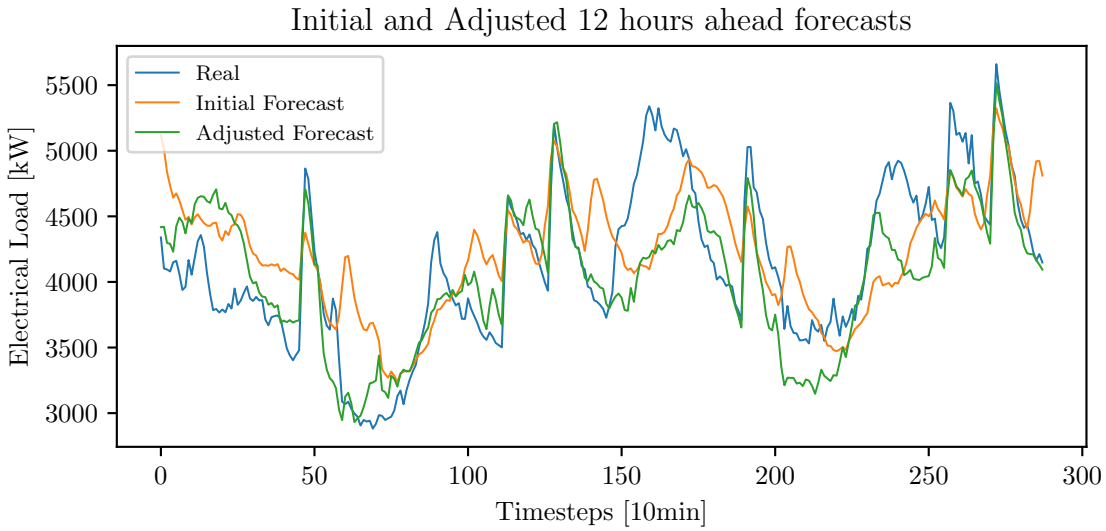

Figure 5.15: 12-hours ahead adjusted forecasts

Summary

Finally, Table 5.12 summarizes the results of the proposed methodology for the three forecast horizons. Like in the initial and error predictions, the forecast accuracy decreased as the forecast horizon increased which reinforce the greater degree of difficulty to make longer forecasts due to the higher uncertainty of the data.

Table 5.12: Summary of the adjusted models for the three horizons

Horizon	RMSE [kW]	MAE [kW]
10-minutes	103.69	78.01
1-hour	239.34	188.49
12-hours	343.05	257.43

Chapter 6

Wind Speed and Power Forecasting Results

In this Chapter, the results obtained with the wind speed and power forecasting models are presented as follows: Section 6.1 introduces the results for the wind speed forecasts, including the benchmark models, in Subsections 6.1.1 and 6.1.2, initial models, in Subsection 6.1.3, error analysis and forecast, in Subsections 6.1.4 and 6.1.5, and the adjusted model, in Subsection 6.1.6; Section 6.2 presents the results for the wind power forecasts, including the power curve modeling, in Subsection 6.2.1, and the power estimation, in Subsection 6.2.2.

6.1 Wind Speed

6.1.1 Baseline 1 – NWP Model

The first baseline model is the forecasts made by the ECMWF NWP model itself, which is detailed in Section 4.1. In the next subsections, the results for the different time-horizons are presented.

12–hour horizon

Table 6.1 shows the results of the 12-hour horizon NWP model compared to the same test set of the PB.12 model for the three turbines, as it had the best ANN performance (see Section 4.3). One can notice that there was no significant variation between the errors for different turbines (less than 0.1 m/s in all cases). This was an expected finding as the wind profile for turbines that are close to each other tends to be very similar.

Table 6.1: NWP.12 forecasts metrics

Turbine	RMSE [m/s]	MAE [m/s]
A	1.682	1.312
B	1.692	1.304
C	1.672	1.289

Figures 6.1, 6.2 and 6.3 show that, for Turbines A and B, the error distribution was similar to the Gaussian Distribution, with the mean slightly smaller than zero (-0.87 and -0.63 m/s, respectively) and the standard deviation of 1.44 and 1.57 m/s, while Turbine C showed a more irregular distribution but with comparable results (mean of -0.79 m/s and standard deviation of 1.47 m/s). The different distribution for Turbine C can be explained by the less amount of data for this turbine (≈580 samples versus ≈3400 for Turbines A and B). Also, the negative mean suggests that there is some bias in the NWP forecasts which seem to be generally higher than the actual wind speed. This is possibly explained by the difference between the NWP model height (106.54m) and the turbine hub height (90m) and/or by some inherent model errors.

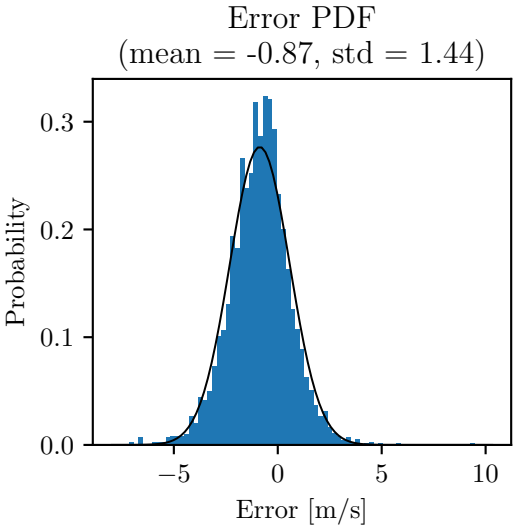

Figure 6.1: NWP.12.TA Error PDF

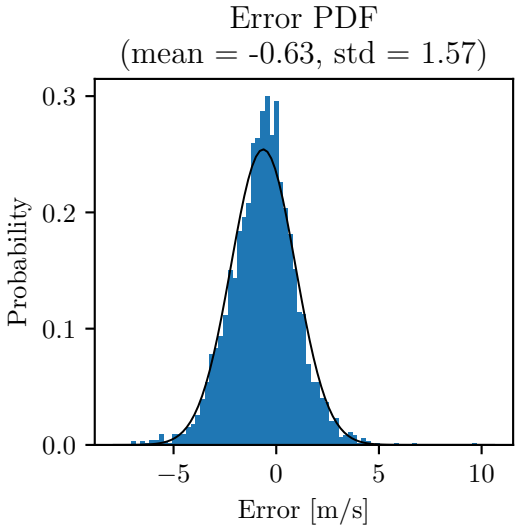

Figure 6.2: NWP.12.TB Error PDF

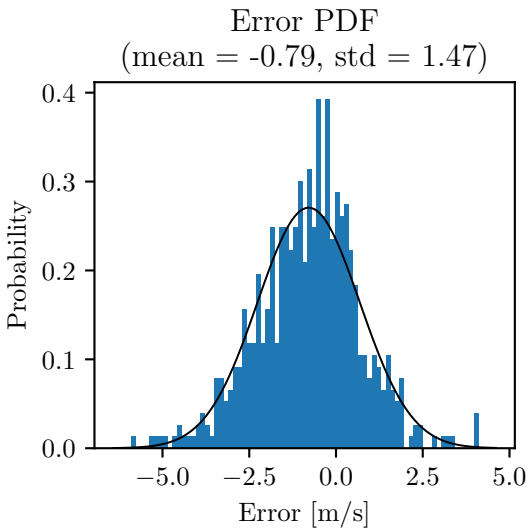

Figure 6.3: NWP.12.TC Error PDF

24–hour horizon

Table 6.2 shows the results of the 24-hour horizon NWP model compared to the same test set of the PB.24 model for the three turbines, as it also had the best ANN performance (see Section 4.3). As

in the 12-hour horizon case, there was no significant variation between the errors for different turbines. However, it can be noticed that the errors are greater, which can be explained by the higher uncertainty of longer forecasts.

Table 6.2: NWP.24 forecasts metrics

Turbine	RMSE [m/s]	MAE [m/s]
A	1.751	1.366
B	1.759	1.350
C	1.688	1.310

As the error distribution in the 24-hour horizon was very similar to the one of the 12-hour horion, the mean and standard deviation values are presented in Table 6.3 instead of figures. A small but in some way interesting difference can be seen in the standard deviation of Turbines A and B, which were slightly higher than the previous ones. This is possibly another indication of the higher uncertainty of longer forecasts.

Table 6.3: NWP.24 error mean and std. dev.

Turbine	Error Mean [m/s]	Error Std. Dev. [m/s]
A	-0.85	1.53
B	-0.62	1.65
C	-0.78	1.50

6.1.2 Baseline 2 – Adjusted NWP

As the NWP models can have some bias, another baseline model was considered by adding the average error of the NWP model to the original forecasts. The average error was calculated over the same training set of the best ANN models and added to the test set predictions so that the results could be compared.

Tables 6.4 and 6.5 show the results of both 12 and 24-hour horizon for the adjusted NWP model. This simple correction could lead to a decrease of up to 14.1% on the RMSE and to 19.1% on the MAE of the initial forecasts on the 12-hour horizon and to 12.5% on the RMSE and to 16.0% on the MAE on the 24-hour horizon. Also, one can notice that the mean error was closer to zero while the standard deviation was the same as in the previous case.

Table 6.4: ANWP.12 forecasts metrics

Turbine	Error mean [m/s]	Error Std. Dev. [m/s]	RMSE [m/s]	MAE [m/s]
A	-0.08	1.44	1.444	1.102
B	-0.09	1.57	1.572	1.205
C	-0.10	1.47	1.478	1.138

Table 6.5: ANWP.24 forecasts metrics

Turbine	Error mean [m/s]	Error Std. Dev. [m/s]	RMSE [m/s]	MAE [m/s]
A	-0.04	1.53	1.532	1.162
B	-0.06	1.65	1.648	1.257
C	-0.06	1.50	1.497	1.168

6.1.3 Initial Results

The initial results are presented for each turbine in each forecast horizon. As mentioned in Section 4.3, only for Turbine A all the models were adjusted. For the other two turbines, the top-3 models were trained.

12–hour horizon — Turbine A

Tables 6.6, 6.7 and 6.8 show the three best results obtained for the training of Turbine A models in the 12-hour horizon on the validation set. The complete tables can be found in the Appendix B. It can be seen that all the models had similar performances regarding both error metrics, with the configuration ”A” showing slightly worse results. This is possibly explained by the fact that the original NWP wind speed forecast at time t showed the highest correlation with the target (wind speed measured in SCADA) among other timesteps and, therefore, contains most of the relevant information used to adjust the NWP outputs. Also, the use of timesteps closer to t seemed to have added a few more information (models ”B” and ”C”) while further timesteps did not. However, in order to improve these forecasts, searching for other explanatory variables should be likely considered.

Another remark that can be made is related to the almost nonexistent difference between the results within each model. This means that the models were able to reach nearly the same minimum despite the different parameters and can be an indication of overcomplexity. Hence, further investigation concerning the choice of those parameters is encouraged.

In relation to the execution time, the first configuration also presented worse performance while the two other models had similar results. By analyzing the number of epochs over the 5 folds that each model took to train, it was not possible to establish a direct relation with the computational time. Still, potential reasons to this difference are the gradient calculation complexity, which can differ from one model to another.

Table 6.6: PA.12.TA best results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PA4.12.TA	1.437	1.092	33.021
PA6.12.TA	1.439	1.094	33.892
PA1.12.TA	1.441	1.094	29.042

Table 6.9 shows a summary of the best proposed model and the two baseline models on the test set. One can notice that the proposed model had significant better results for the three error metrics analyzed. The improvements over the initial NWP forecast were about 23.3% on the RMSE, 24.8% on

Table 6.7: PB.12.TA best results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB10.12.TA	1.384	1.052	15.675
PB3.12.TA	1.388	1.058	15.525
PB12.12.TA	1.389	1.059	18.157

Table 6.8: PC.12.TA best results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PC11.12.TA	1.390	1.057	15.034
PC3.12.TA	1.392	1.059	18.052
PC1.12.TA	1.393	1.059	18.849

the MAE and 41.3% on the R2, while for the ANWP they were 10.7%, 10.5% and 11.7%. This result suggests that the errors in the NWP model is not only caused by some bias and/or the difference between the model height and the turbine hub height. The use of an ANN was able to find and correct other error patterns and is, thus, a reasonable and efficient choice to enhance NWP wind speed forecasts.

Table 6.9: Turbine A 12–hour horizon summary

Model	RMSE [m/s]	MAE [m/s]	R2
PB10.12.TA	1.290	0.986	0.705
ANWP	1.444	1.102	0.631
NWP	1.682	1.312	0.499

Figure 6.4 shows the wind speed forecasts made with the NWP model, the adjustment made with model PB10.12.TA and the measured wind speed. It is notable that for higher wind speeds, the proposed model decreased the forecasts, while for lower wind speeds they were slightly increased. A Weibull Distribution of the forecasts was fitted and can be seen in Figure 6.5. The proposed model could not generate predictions that followed the typical wind distribution, being more similar to a Gaussian Distribution and with the shape parameter above the common upper limit (3.366 vs 3). On the other hand, it was able to improve the scale parameter by bringing it closer to the that of the true distribution for Turbine A than the NWP outputs were (see Figure 4.3 and 4.4).

12–hour horizon — Turbine B

As explained in Section 4.3, only the models PB10, PB3 and PB12 were retrained for Turbine B, as they had the best results for Turbine A. The results of this step can be found in Table 6.10. First, it is possible to notice that the three models were ranked in the same way as they were for Turbine A. However, the results of the first two were slightly worse while the last one were considerably worse. Although the turbines are close to each other and the wind profile is probably very similar, there can be small differences that impact the models fitting. It was possible to notice that Turbine B models went through all the epochs, but the loss decreased slower than for Turbine A. Therefore, it is possible that increasing the number of epochs to train would result in better performance and closer to the one observed for Turbine A.

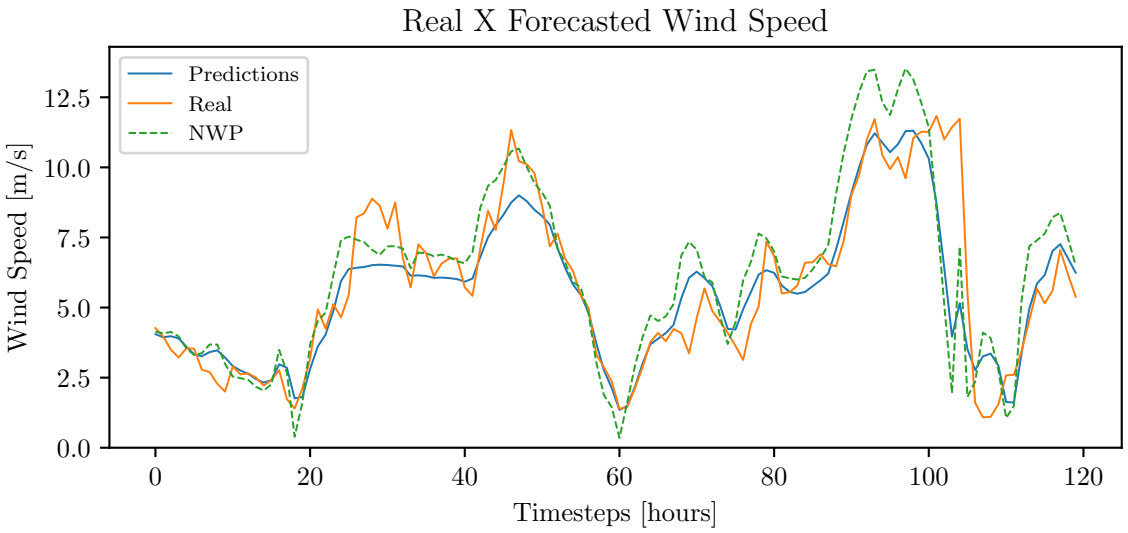

Figure 6.4: PB10.12.TA forecast

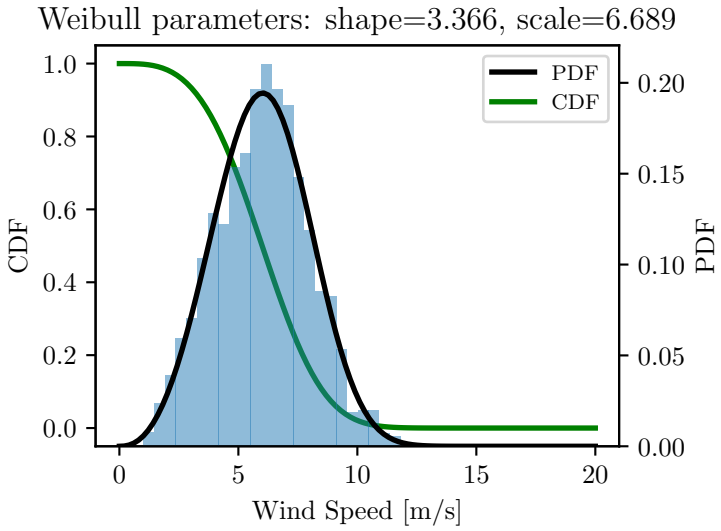

Figure 6.5: Turbine A forecasts Weibull Distribution

Table 6.10: PB.12.TB results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB10.12.TB	1.534	1.177	13.427
PB3.12.TB	1.580	1.230	12.618
PB12.12.TB	4.303	3.685	12.402

Table 6.11 shows a comparison between the results obtained with the proposed model and the two baseline models on the test set. As in the previous case, the proposed model was able to improve the baseline models significantly. However, this improvement was not as notable as before, with a decrease of 14.4% on the RMSE, 14.6% on the MAE and 22.7% on the R2 of the NWP and 7.9% on the RMSE, 7.6% on the MAE and 9.9% on the R2 of the ANWP. It should be mentioned that, in this case, the application of an ANN is also justifiable as it enhance the initial results considerably. Finally, Figure 6.6 shows the NWP forecasts, the proposed model adjustment and the actual wind speed.

Table 6.11: Turbine B 12–hour horizon summary

Model	RMSE [m/s]	MAE [m/s]	R2
PB10.12.TB	1.448	1.113	0.666
ANWP	1.572	1.205	0.606
NWP	1.692	1.304	0.543

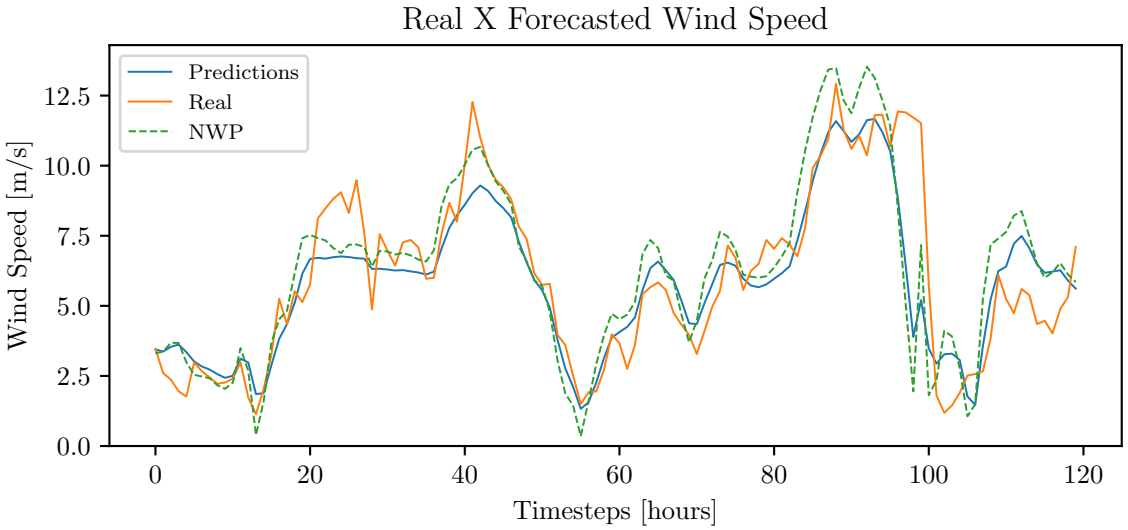

Figure 6.6: PB10.12.TB forecast

12–hour horizon — Turbine C

Similarly to Turbine B, only the three best models for Turbine A were retrained for Turbine C and the results are presented in Table 6.12. Here, the models were also ranked in the same way as for the other turbines, but with a larger difference between each model. The performance of the best model (PB10.12.TC) was in between the ones observed for the other turbines. The computational time was smaller in this case as the dataset was limited to 4 months instead of 2 years.

Table 6.12: PB.12.TC results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB10.12.TC	1.455	1.098	11.075
PB3.12.TC	1.811	1.376	10.871
PB12.12.TC	5.635	5.162	11.679

Table 6.11 summarizes the results obtained with the proposed model and the two baseline models for Turbine C on the test set. As for the other two turbines, here it was also possible to enhance the initial results considerably. The improvements were approximately 27.2% on the RMSE, 27.7% on the MAE and 79.3% on the R2 for the NWP and 17.6% on the RMSE, 18.1% on the MAE and 30.8% on the R2 for the ANWP. Again, the deployment of an ANN showed better performance than the baseline models and was able to produce more accurate predictions. In Figure 6.7, forecasts made by the NWP model and the proposed model can be seen together with the measured wind speed.

Table 6.13: Turbine C 12–hour horizon summary

Model	RMSE [m/s]	MAE [m/s]	R2
PB10.12.TC	1.218	0.932	0.667
ANWP	1.478	1.138	0.510
NWP	1.672	1.289	0.372

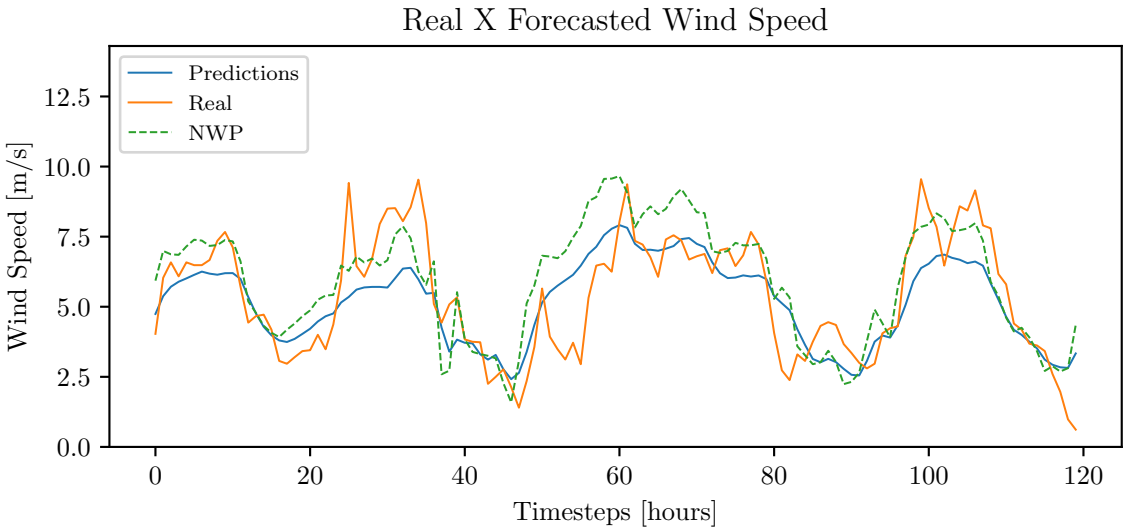

Figure 6.7: PB10.12.TC forecast

12–hour horizon — Best Results

Table 6.14 shows the best results obtained for each turbine on the 12-hour horizon. In every case, the proposed model achieved better results than the baseline models. For Turbine A, 16 models were proposed and trained in three different input configurations while, for the other two turbines, only the three best for configuration B were adjusted. It can be noticed that the Turbine C model performed better than the other two turbines models. This can be possibly explained by the smaller seasonality observed on the wind dataset of Turbine C when compared to the other turbines datasets (see Section 4.2). As described in Section 4.1, Turbines A and B used 2 years of data while Turbine C had only 4 months. Due to this reason, it is believed that the Turbine C model was able to find more consistent error patterns and make better adjustments. Therefore, it is suggested that new experiments are made considering a dataset with a smaller period of time.

Table 6.14: Best results for each turbine (12–hour horizon)

Turbine	RMSE [m/s]	MAE [m/s]	R2
A	1.290	0.986	0.705
B	1.448	1.113	0.666
C	1.218	0.932	0.667

24–hour horizon — Turbine A

As mentioned in Section 4.3, only the input configuration B was tested for the 24-hour horizon, as it had the best results in the previous case. Table 6.15 shows the top-3 models for Turbine A in the 24-hour

horizon during the training. The complete tables can be found in Appendix B. Comparing the results with the ones found in the 24-hour horizon (see Table 6.7 and Appendix B), it can be noticed that the best models were not the same. However, the differences between them were again very small.

Table 6.15: PB.24.TA best results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB11.24.TA	1.459	1.106	23.269
PB5.24.TA	1.461	1.107	38.018
PB4.24.TA	1.461	1.107	38.144

Table 6.16 shows the results obtained on the test set with the best proposed model and baseline models. Again, the proposed model was able to improve significantly the initial results. The improvements were 20.6% on the RMSE, 21.7% on the MAE and 43.8% on the R2 for the NWP and 9.2% on the RMSE, 8.0% on the MAE and 12.5% on the R2 for the ANWP. These findings were similar to the ones of the 12-hour horizon. It should be mentioned that the results of the 24-hour horizon were worse than those of the previous case (see Table 6.9) if a comparison between the models in each scenario is made. This reinforces the fact that predictions on longer terms are more and uncertain and difficult to be made. Finally, Figure 6.8 shows the forecasts made with the proposed model, the NWP and the measured wind speed.

Table 6.16: Turbine A 24–hour horizon summary

Model	RMSE [m/s]	MAE [m/s]	R2
PB11.24.TA	1.391	1.069	0.657
ANWP	1.532	1.162	0.584
NWP	1.751	1.366	0.457

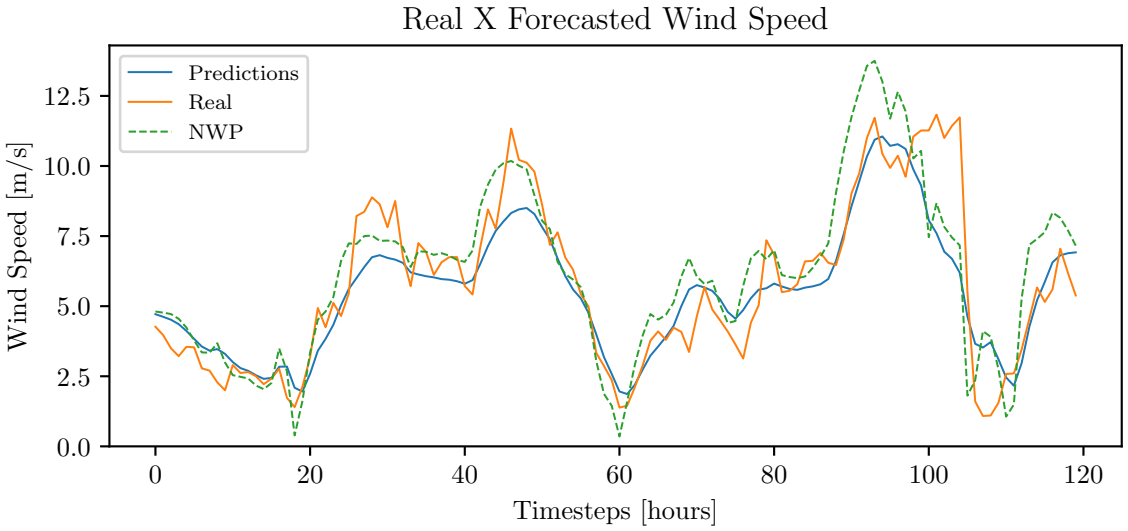

Figure 6.8: PB11.24.TA forecast

24–hour horizon — Turbine B

As in the 12-hour horizon, only the top-3 models found for Turbine A were retrained for the other two turbines. By looking at Table 6.17, it can be seen that the best models were not ranked the same way as they were for Turbine A, which is a different from the findings of the 12-hour horizon. The execution time was also distinct. The reason to this was that these models needed fewer epochs to reach a minima that made the training stop. On the other hand, one of the models also had considerably worse results while the other two were similar.

Table 6.17: PB.24.TB results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB5.24.TB	1.591	1.211	5.018
PB11.24.TB	1.669	1.297	13.183
PB4.24.TB	3.757	3.115	12.742

Table 6.18 summarizes the results of Turbine B on the 24-hour horizon. Once again, the proposed model had better performance than the baseline models, with improvements of 13.7% on the RMSE, 13.4% on the MAE and 24.7% on the R2 for the NWP and 7.9% on the RMSE, 7.0% on the MAE and 11.5% on the R2 for the ANWP. Again, by comparing this table with Table 6.11, the results of every model for Turbine B on the 24-hour horizon were worse than those on the 12-hour horizon. Lastly, Figure 6.9 shows the SCADA wind speed and the outcomes of the proposed model and the NWP.

Table 6.18: Turbine B 24–hour horizon summary

Model	RMSE [m/s]	MAE [m/s]	R2
PB5.24.TB	1.518	1.169	0.632
ANWP	1.648	1.257	0.567
NWP	1.759	1.350	0.507

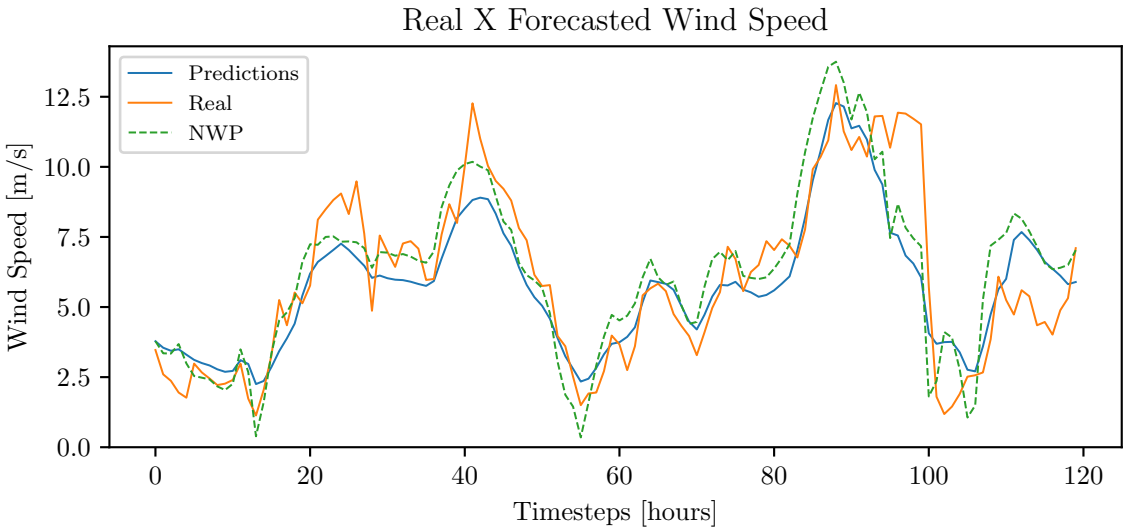

Figure 6.9: PB5.24.TB forecast

24–hour horizon — Turbine C

Table 6.19 shows the results obtained for Turbine C on the 24-hour horizon. As for Turbine B, the best model was not the one found for Turbine A and the results of one of the models were significantly worse than the others. Also, the execution time differed but, in this case, in addition to fewer epochs to stop the training, the dataset size was smaller.

Table 6.19: PB.24.TC results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB5.24.TC	1.511	1.137	2.933
PB11.24.TC	2.003	1.505	4.412
PB4.24.TC	5.245	4.763	4.095

Table 6.20 presents similar findings as before. The proposed model improved the initial results by 22.2% on the RMSE, 22,1% on the MAE and 70.3% on the R2 for the NWP and 12.3% on the RMSE, 12.6% on the MAE and 23.3% on the R2 for the ANWP. In the same way as in the previous cases, the results for the 24-hour horizon were worse than those of the 12-hour horizon when the same models are compared (see Table 6.13). Ultimately, Figure 6.10 shows illustrates the predictions made with the proposed model and the NWP together with the recorded SCADA wind speeds.

Table 6.20: Turbine C 24–hour horizon summary

Model	RMSE [m/s]	MAE [m/s]	R2
PB5.24.TC	1.313	1.021	0.613
ANWP	1.497	1.168	0.497
NWP	1.688	1.310	0.360

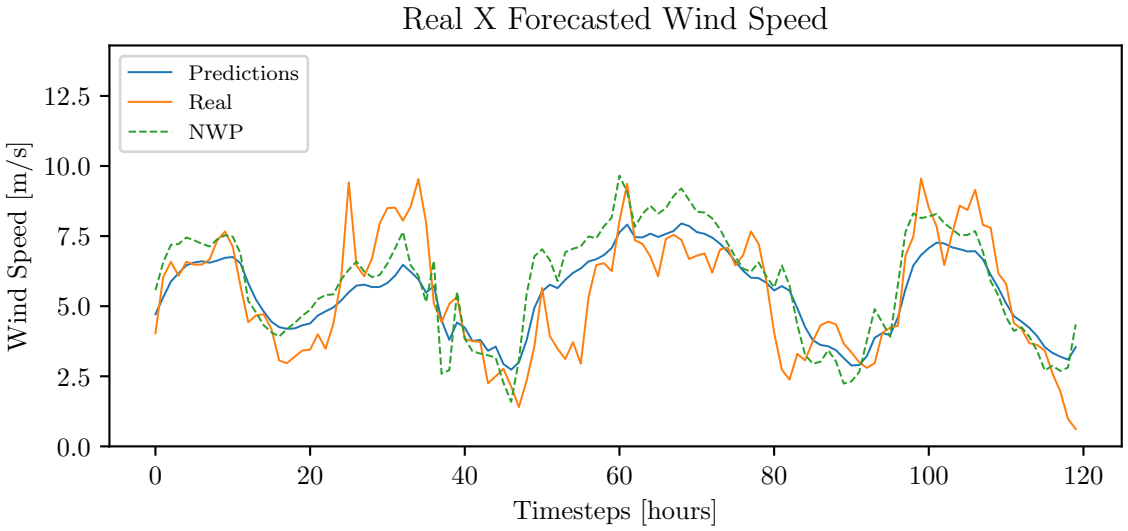

Figure 6.10: PB5.24.TC forecast

24–hour horizon — Best Results

Table 6.21 presents the best results obtained for each turbine on the 24-hour horizon. The proposed model was able to provide better results than the baseline models in every case. In this scenario, 16 models were proposed and trained using input configuration ”B” for Turbine A while, for the other two turbines, only the three best found for Turbine A were evaluated. Again, Turbine C model performed better, followed by Turbines A and B (see Table 6.14). These results strengthens the possibility of finding better results when using smaller datasets, where the wind variability is reduced such as in Turbine C dataset.

Table 6.21: Best results for each turbine (24–hour horizon)

Turbine	RMSE [m/s]	MAE [m/s]	R2
A	1.391	1.069	0.657
B	1.518	1.169	0.632
C	1.313	1.021	0.613

6.1.4 Error Analysis

Figures 6.11 and 6.12 show the error distributions for the best Turbine A models in the two forecast horizons (PB10.12.TA and PB11.24.TA). One can notice that both error distributions seem to follow a Gaussian distribution and presented similar results, with the mean slightly more distant from zero and smaller standard deviation in the first horizon. Comparing to the results of the baseline models, the proposed model was able to reach an error mean close to the one obtained with the adjusted NWP with smaller standard deviation (see Table 6.4).

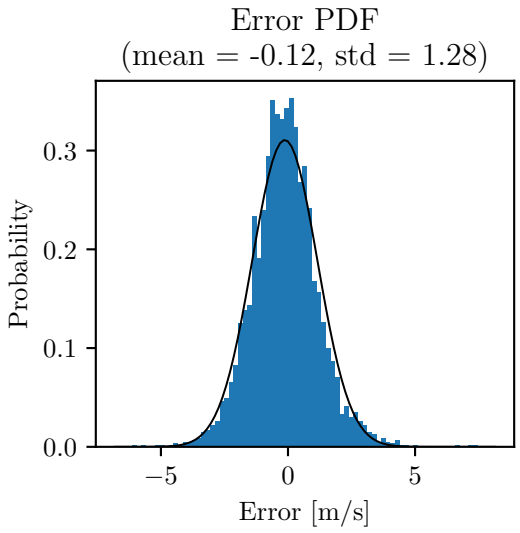

Figure 6.11: PB10.12.TA Error PDF

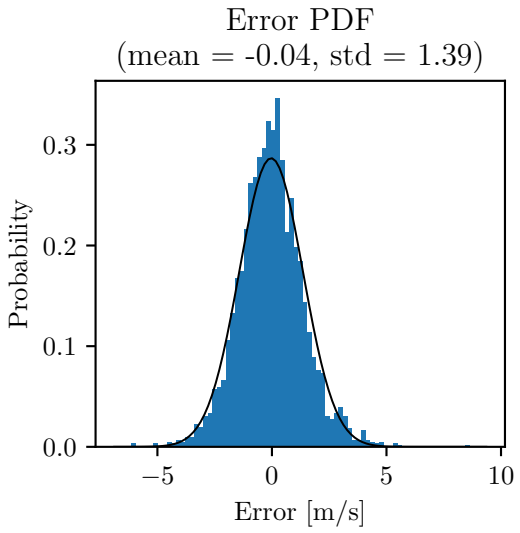

Figure 6.12: PB11.24.TA Error PDF

In Figures 6.13 and 6.14, the RMSE for each forecast horizon is presented. In this case, the persistence method result was also included. The results show that this method had the worst performance by far, with a significant error increase for longer horizons. This finding is very similar to the one in [37].

Also, the proposed model outperformed both the NWP and Adjusted NWP models for every forecast horizon. It is worth mentioning that the proposed model RMSE showed a behavior similar to the one of the other models, with an increasing trend for longer horizons. However, the results indicate that the model was able to find error patterns that can not be corrected only by adding the average error.

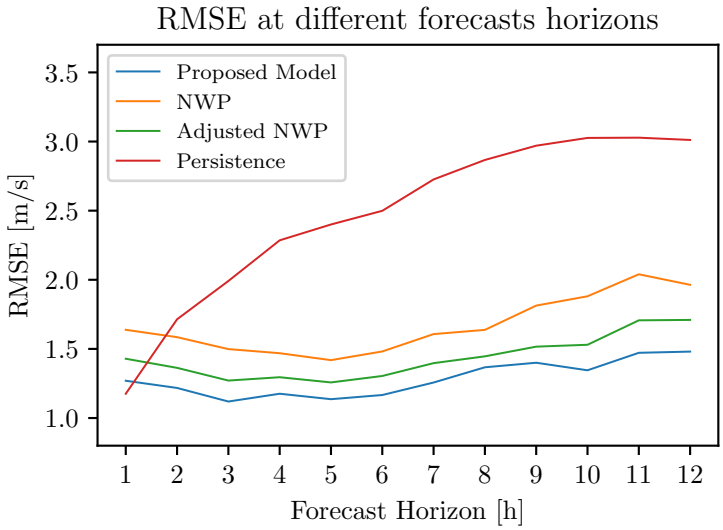

Figure 6.13: RMSE for the model PA10.12.TA

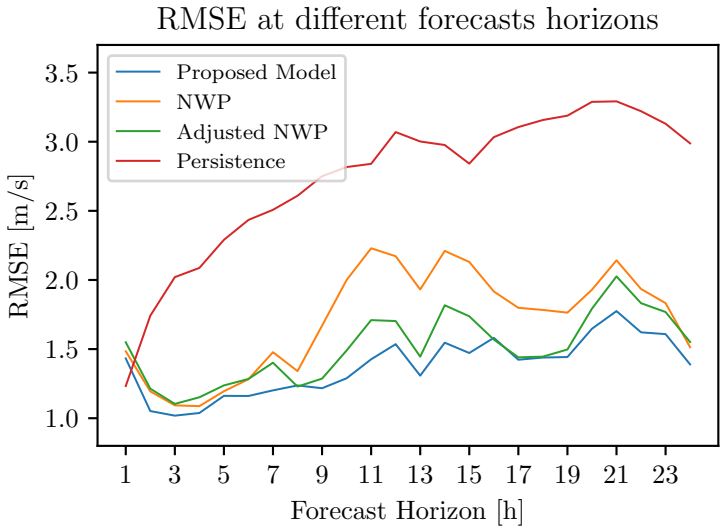

Figure 6.14: RMSE for the model PB11.24.TA

Finally, in Figure 6.15, the ACF plot for Turbine A in the 12-hour horizon can be seen. Unlike the relationship observed for the electrical load forecasts, the autocorrelation coefficients here dropped quickly to zero and had only a few values above the significance range (see Section 5.2), which may suggest that previous errors do not provide useful information about future errors. However, based on this result and on the proposed error forecast methodology, an input vector with the last 12 errors (from H-11 to H) and the most recent 12 forecasts (from H+1 to H+12) was used to estimate the next errors (from H+1 to H+12).

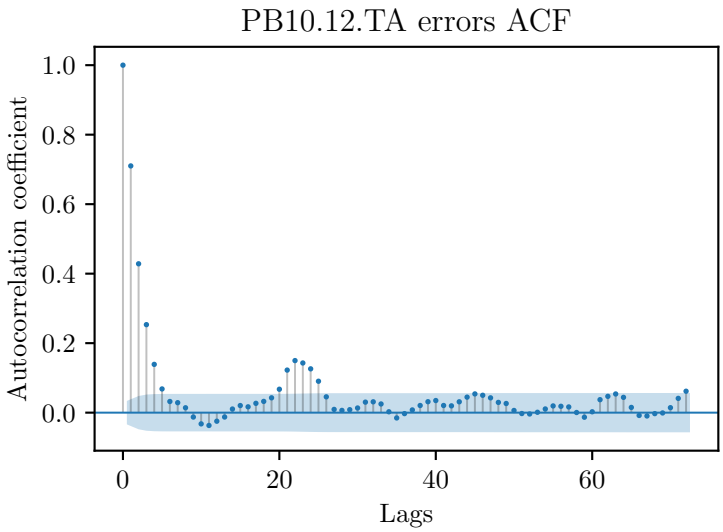

Figure 6.15: PB11.12.TA Error ACF

6.1.5 Error Forecast

Table 6.22 shows the results of the three best wind speed error forecasting models. The complete results can be found in Appendix B (Table B.5). As observed for the electrical load error forecast (see Section 5.3), all the models presented similar performances regarding both metrics and execution times and, therefore, any strong conclusion about the different configurations can be made.

The results in Table 6.9 and in Figure 6.16 show the bad performance of the error forecasting model, which was not able to find any pattern in the data. This is in contrast with the findings of the previous section, where the proposed error model was able to provide good estimates of the error. One of the possible reasons is related to the chosen input parameters. For the load forecast case, the ACF plots showed a significant correlation between previous observations and the target values (see Figures 5.7, 5.8 and 5.9 in Section 5.3) while, in this case, this relationship was not observed (Figure 6.15). Thus, it is reasonable to conclude that more features would be necessary to improve the error forecasts with this approach. Also, due to the stochastic nature of the wind, it is possible that the error pattern is random and, therefore, much harder to predict.

Table 6.22: Turbine A best error models

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
10	1.244	0.945	4.818
11	1.245	0.947	4.519
8	1.247	0.951	4.423

6.1.6 Adjusted Results

As explained in section 4.3, the adjusted results were obtained by adding the predicted error to the initial forecasts. In Table 6.23, the metrics for both initial and adjusted models are presented. One can notice that the proposed wind speed error forecasting model was not able to improve the results as it was observed for the electrical load scenario. On the contrary, a slight performance decline was observed.

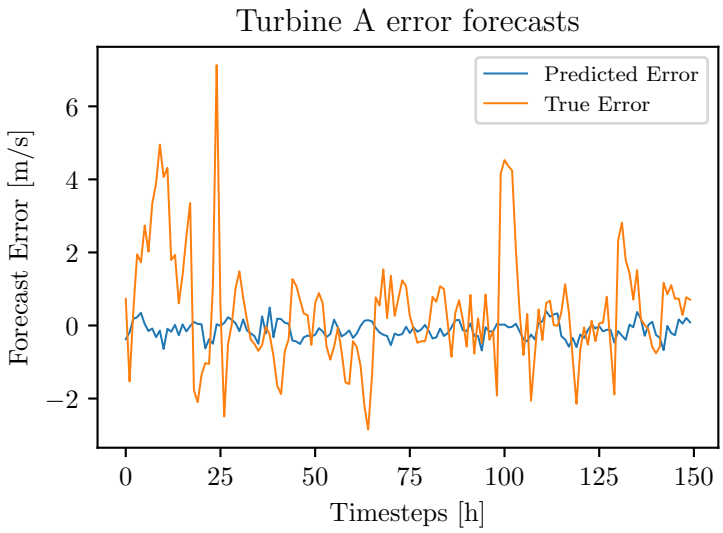

Figure 6.16: Turbine A error forecasts

These results are illustrated in Figure 6.17. This outcome was somehow expected after the findings in the previous section, which showed the poor performance of the proposed model to forecast the errors. However, it is worth mentioning that the two proposed models had better results than the NWP model.

Table 6.23: Turbine A Initial and Adjusted models metrics

Model	RMSE [m/s]	MAE [m/s]
Initial	1.386	1.050
Adjusted	1.393	1.058
NWP	1.655	1.268

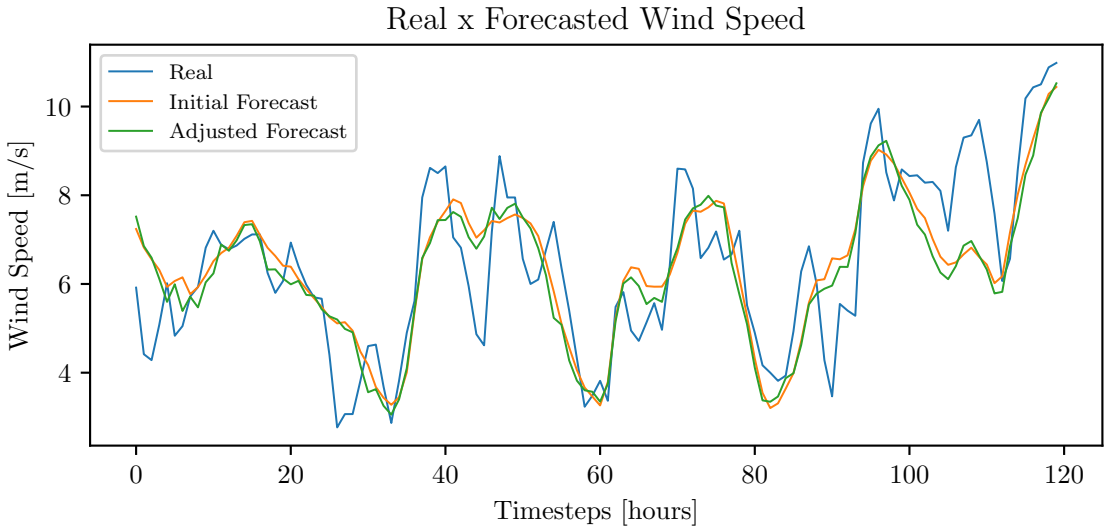

Figure 6.17: Turbine A initial and adjusted Forecasts

6.2 Wind Power

6.2.1 Wind Power Curve Modeling

As explained in Subsection 4.3.3, after removing measurements that did not follow regular operating conditions, several samples in the operating region II still behave differently from what it was expected and, to filter those points, the DBSCAN clustering algorithm was used. The results obtained with this method can be seen in Figure 6.18. The proposed approach was able to correctly classify the samples more distant to the turbine's expected behavior as outliers. Those points were removed from the dataset and not used to adjust the wind power curve.

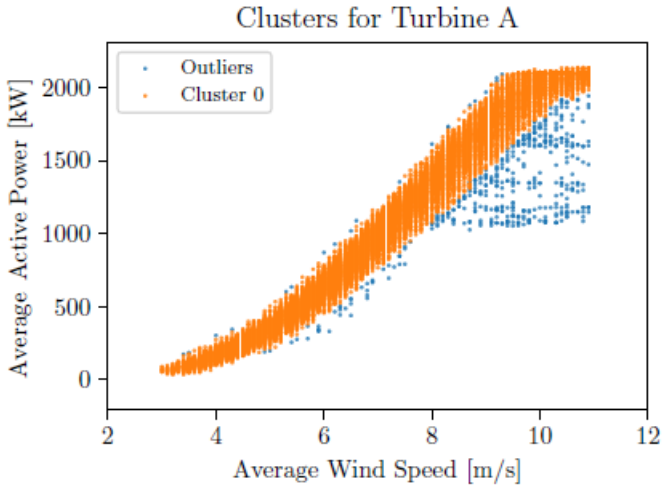

Figure 6.18: DBSCAN Clusters for Turbine A

With the remaining samples, polynomials with degrees from 3 to 7 were fitted so that the wind power curve could be found. In Figure 6.19, a 4th degree polynomial is shown as this was the best curve found for this data. Finally, with the curve fitted for the region II, and considering an output power of zero for regions I and IV and the nominal power in region III, the wind power curve for this turbine was defined.

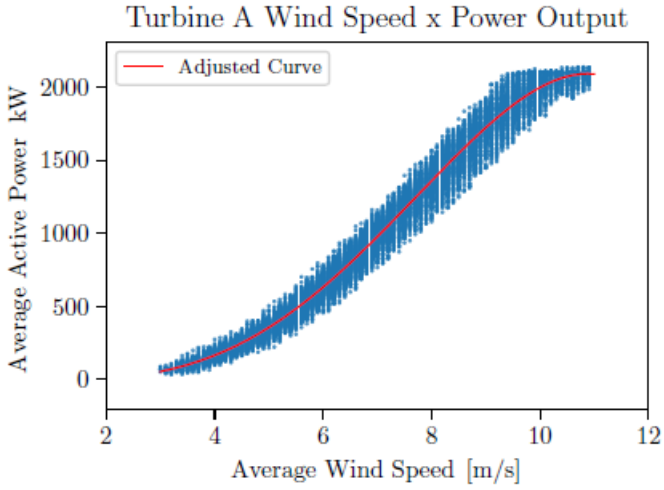

Figure 6.19: Adjusted Wind Power Curve for Turbine A

6.2.2 Wind Power Estimation

The adjusted wind power curve was used to estimate the wind power production using the wind speeds measured with the SCADA and the ones of the proposed initial model and the NWP. The first was chosen so that the best possible result with the available data could be analyzed, as the error corresponding to this estimate is probably related to noise and sampling of the data. The choice for the initial model wind speed is due to its better results when compared to the adjusted model, which was not able to improve the initial results. Finally, the NWP wind speed was used to generate the power forecasts of the baseline scenario.

The results of the wind power forecast can be seen in Table 6.24. One can notice that the power estimates made with proposed model wind speed presented an improvement of 26.5% and 24.5% over the RMSE and MAE of the ones obtained using the NWP wind speed, moving closer to the results found using the SCADA wind speed. These findings validate the proposed methodology and show that having more accurate wind speed forecasts is crucial to improve the power predictions.

In order to illustrate the achieved results, Figure 6.20 shows the actual wind power output and the forecasts made with both the proposed model and NWP wind speeds. First, it is noticeable the better performance of the proposed model when compared to the baseline during most of the time. The main exception can be seen during nominal power production (from timestep 0 to 30 approximately). At those timesteps, the NWP seemed to perform better. A probable explanation to this is that the proposed model primarily adjust the average wind speed and, therefore, the higher wind speeds turn out to be reduced and the power output do not correspond to the nominal power. However, in general, the proposed model tends to be more accurate and generates better estimates.

Table 6.24: Results of the wind power forecast for turbine A

Input Data	RMSE [kW]	MAE [kW]
SCADA Wind Speed	73.40	214.45
Proposed Initial Model Wind Speed	266.86	385.56
NWP Wind Speed	363.18	510.90

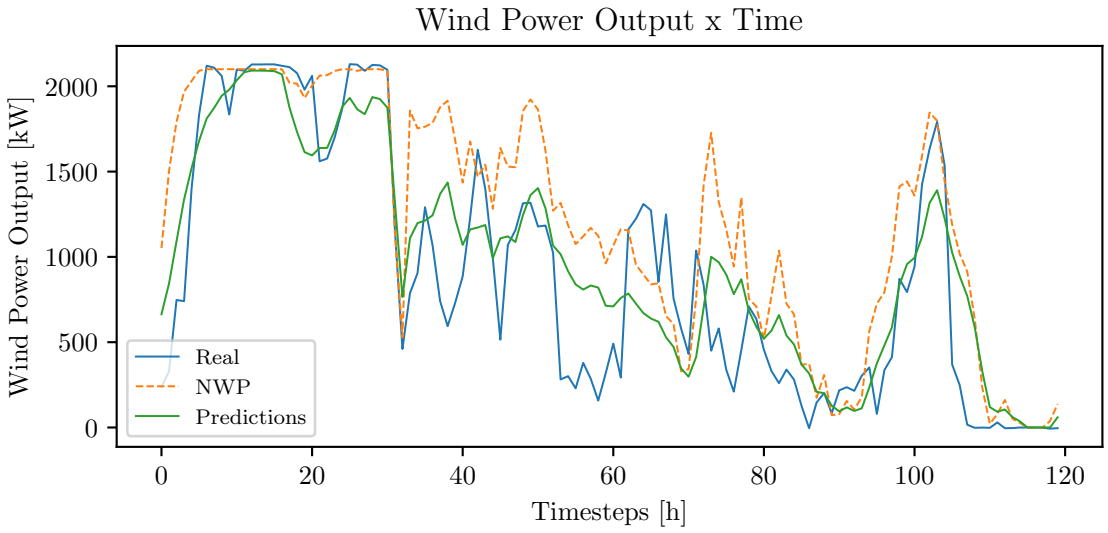

Figure 6.20: Wind Power forecast for Turbine A

Chapter 7

Conclusions

In this Chapter, a summary of the achievements of this work and suggestions for future research are presented in Sections 7.1 and 7.2 respectively.

7.1 Achievements

To overcome the global climate crisis, multiple adaptation and mitigation measures are required. As the energy sector accounts for most part of the GHG emissions, it has a fundamental role. Despite the recent advances, such as the sharp increase of RES in the electricity mix, there is still much to improve to meet the sustainability and climate goals. The ongoing transformations has been increasing the electrical system complexity and volatility, imposing new challenges to the sector. To avoid energy supply issues and provide a reliable service, it is essential to anticipate the system behavior. In this sense, this work presented a ML based methodology to forecast electrical load demand, wind speed and power in different time scales by incorporating an error correction step following the initial models results.

The proposed methodology utilized a FFNN for both initial and error forecasts as the error time series often contains useful information which can be used to improve initial estimations. After a careful state-of-the-art review and an extensive data preliminary analysis, the models were created. For that, well-established Python's libraries for ML, such as TensorFlow, scikit-learn and others were used. A search for the best models inputs and parameters configurations was also conducted. Furthermore, benchmark models, namely persistence and linear regression for the electrical load demand and NWP for the wind speed, were used to perform the same task and had its results compared with the proposed methodology.

With regard to the electrical load demand forecasts, historical data collected from a distribution system operator in France was used. The data referred to the measured load demand in an industrial area connected to the medium voltage grid and was sampled at every 10 minutes. The forecasts were made for three time horizons: 10-minutes, 1-hour and 12-hours ahead. The results demonstrated that the proposed initial models outperformed the linear regression model for the last two horizons, while, for

the first, the results were worse than those achieved with a simple persistence model. By comparing the results for the three time scales, it was verified that the forecast accuracy decreased for longer time horizons which highlights the higher difficulty to make longer forecasts due to the higher uncertainty of the data. Also, for very short time scales, the findings suggest that a simpler model can provide better estimates.

The initial electrical load demand forecasting errors analysis showed that the model might not have extracted all the information from the input data as the error distribution was not centered in zero and there was a correlation between a given error and the previous ones. Indeed, the error forecasting models achieved good forecasting accuracy, with lower RMSE and MAE than the initial models. Furthermore, by combining the predicted error to the initial forecasts it was possible to improve the initial results significantly for all time scales. Especially for the 10-minutes ahead forecasts, the application of the proposed methodology resulted in a more accurate model than the persistence model.

For the wind speed forecasts, historical data from a NWP model and the SCADA system of a wind farm in the south of South America was used. The data was sampled at every 1 hour and forecasts were made for two time horizons: 12-hours and 24-hours ahead. Several models were tested for one of the turbines (Turbine A) and the three best were used for other two. The results showed that the proposed initial models were able to slightly outperform the benchmark models, namely the NWP and an adjusted NWP, for all turbines in every forecast horizon. The findings suggest that the proposed model was able to find and correct error patterns other than the one caused by the difference between the NWP model height and the turbine hub height. Also, the forecasting accuracy was higher for Turbine C, which had only 4 months of data instead of 2 years for the other turbines. In this case, the use of less data leaded to better results probably because of the smaller seasonality present in Turbine's C data.

Unlike the initial electrical load demand forecasting error analysis, the errors of the wind speed forecasting models presented a distribution centered very close to zero and almost no correlation between present and previous errors. As an expected consequence, the error forecasting models were not able to achieve better accuracy and improve the initial models results. As the wind speed is extremely related to weather and terrain characteristics, considering this type of data might be beneficial for wind speed forecasting.

The historical data from the wind farm was also used to adjust the wind power curve of one of the turbines. For this, the DBSCAN clustering algorithm was employed. The main goal of this method is to identify and classify samples in high and low density areas of the dataset. The clustering algorithm had a satisfactory performance with regard to identifying samples that apparently did not have the expected wind speed and power relationship. However, the choice of the parameters was done by trial and error, which is not the most efficient way. Those points were removed from the dataset and a wind power curve was approximated. With the adjusted power curve and the estimated wind speed, wind power forecasts were made. The results accomplished with the proposed wind speed model were superior than those of the original NWP.

The findings of this research suggest that the proposed methodology can be beneficial for the energy sector, as it provided enhanced forecasts with ML models that work with limited amount of data. When

data is scarce, it is essential to extract as much information as possible from it. The analysis of the errors showed that sometimes there is more information in the data than the models are able to find and that including a error correction step can improve the forecasts.

7.2 Future Work

During the development of this research, some ideas for future works emerged:

- For both the electrical load and wind speed forecasting models, it would be interesting to take into consideration other input variables that have been largely used in the literature, such as socio-economic and/or environmental indicators for the load demand predictions and other NWP outputs and/or terrain characteristics for the wind speed estimates;
- Other types of neural networks that are adequate for time-series data processing could be used to make these forecasts, such as LSTM and GRUNN;
- The performance of the error correction step could be tested using other initial forecasting models, such as the abovementioned neural networks, SVM and RF. Also, including a information gain measurement could help the models to weight how much the error forecast should influence the adjusted result;
- It could be beneficial to develop forecasting models for different seasons as both the load and wind pattern showed a considerable seasonality;
- For the wind speed forecasting models, it would be interesting to analyze other FFNN training strategies that could lead to predictions that follow a Weibull distribution;
- An automated method to choose the DBSCAN parameters should be developed, as the trial and error performed in this work would not be practical in real applications, considering that a wind farm have several turbines that require individual adjustments;
- The results of the wind power forecasts could be compared to the ones obtained using the manufacturer power curve or other adjusted power curves.

Bibliography

[1] IPCC. Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, 2014. URL https://www.ipcc.ch/report/ar5/syr/.

[2] Directive 2009/28/EC of the European Parliament and of the Council of 23 April 2009 on the promotion of the use of energy from renewable sources and amending and subsequently repealing Directives 2001/77/EC and 2003/30/EC, 2009.

[3] IPCC. *Climate Change 2014: Mitigation of Climate Change. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2014. URL https://www.ipcc.ch/report/ar5/wg3/.

[4] IEA. Secure Energy Transitions in the Power Sector, 2021. URL https://www.iea.org/reports/secure-energy-transitions-in-the-power-sector.

[5] IEA. Power systems in transition, 2020. URL https://www.iea.org/reports/power-systems-in-transition.

[6] K. B. Debnath and M. Mourshed. Forecasting methods in energy planning models. *Renewable and Sustainable Energy Reviews*, 88(August 2016):297–325, 2018.

[7] T. Ahmad and H. Chen. A review on machine learning forecasting growth trends and their real-time applications in different energy systems. *Sustainable Cities and Society*, 54(October 2019):102010, 2020.

[8] K. Schwertner. Digital transformation of business. *Trakia Journal of Science*, 15(Suppl.1):388–393, 2017.

[9] A. L. Samuel. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal*, pages 210–229, 1959.

[10] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

[11] A. Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, 2017.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pages 1–15, 2015.

[14] M. Q. Raza and A. Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50:1352–1372, 2015.

[15] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renewable and Sustainable Energy Reviews*, 54:1311–1322, 2016.

[16] H. S. Hippert, C. E. Pedreira, and R. C. Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44–55, 2001.

[17] D. Solyali. A comparative analysis of machine learning approaches for short-/long-term electricity load forecasting in Cyprus. *Sustainability (Switzerland)*, 12(9), 2020.

[18] C. Kuster, Y. Rezgui, and M. Mourshed. Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, 35(August):257–270, 2017. ISSN 22106707. doi: 10.1016/j.scs.2017.08.009. URL `http://dx.doi.org/10.1016/j.scs.2017.08.009`.

[19] H. Eskandari, M. Imani, and M. P. Moghaddam. Convolutional and recurrent neural network based model for short-term load forecasting. *Electric Power Systems Research*, 195(March):107173, 2021.

[20] A. Ahmad, N. Javaid, A. Mateen, M. Awais, and Z. A. Khan. Short-Term load forecasting in smart grids: An intelligent modular approach. *Energies*, 12(1):1–21, 2019.

[21] M. Rana and I. Koprinska. Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing*, 182:118–132, 2016.

[22] I. Ozer, S. B. Efe, and H. Ozbay. A combined deep learning application for short term load forecasting. *Alexandria Engineering Journal*, 60(4):3807–3818, 2021.

[23] S. Karthika, V. Margaret, and K. Balaraman. Hybrid short term load forecasting using ARIMA-SVM. In *2017 Innovations in Power and Advanced Computing Technologies, i-PACT 2017*, volume 2017-Janua, pages 1–7, 2017.

[24] H. Jiang, Y. Zhang, E. Muljadi, J. J. Zhang, and D. W. Gao. A Short-Term and High-Resolution Distribution System Load Forecasting Approach Using Support Vector Regression with Hybrid Parameters Optimization. *IEEE Transactions on Smart Grid*, 9(4):3331–3350, 2018.

[25] J. Silva, I. Praça, T. Pinto, and Z. Vale. Energy consumption forecasting using ensemble learning algorithms. *Advances in Intelligent Systems and Computing*, 1004:5–13, 2020.

[26] A. Jozi, T. Pinto, I. Praça, and Z. Vale. Day-ahead forecasting approach for energy consumption of an office building using support vector machines. *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, 641794(641794):1620–1625, 2019.

[27] X. Kong, C. Li, C. Wang, Y. Zhang, and J. Zhang. Short-term electrical load forecasting based on error correction using dynamic mode decomposition. *Applied Energy*, 261(December 2019): 114368, 2020.

[28] IRENA. Renewable Power Generation Costs in 2020, 2021.

[29] A. Tascikaraoglu and M. Uzunoglu. A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews*, 34:243–254, 2014.

[30] S. S. Soman, H. Zareipour, S. Member, O. Malik, and L. Fellow. A Review of Wind Power and Wind Speed Forecasting Methods With Different Time Horizons. pages 1–8, 2010.

[31] W.-Y. Chang. A Literature Review of Wind Forecasting Methods. *Journal of Power and Energy Engineering*, 02(04):161–168, 2014.

[32] M. S. Nazir, F. Alturise, S. Alshmrany, H. M. Nazir, M. Bilal, A. N. Abdalla, P. Sanjeevikumar, and Z. M. Ali. Wind generation forecasting methods and proliferation of artificial neural network: A review of five years research trend. *Sustainability (Switzerland)*, 12(9), 2020.

[33] J. Wang, Y. Song, F. Liu, and R. Hou. 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, 2016.

[34] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian. A Critical Review of Wind Power Forecasting Methods-Past, Present and Future. *Energies*, 13(15):1–24, 2020.

[35] I. Okumus and A. Dinler. Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Conversion and Management*, 123:362–371, 2016.

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

[37] A. Dupré, P. Drobinski, B. Alonzo, J. Badosa, C. Briard, and R. Plougonven. Sub-hourly forecasting of wind speed and wind energy. *Renewable Energy*, 145:2373–2379, 2020.

[38] B. Alonzo, R. Plougonven, M. Mougeot, A. Fischer, A. Dupré, and P. Drobinski. *From Numerical Weather Prediction Outputs to Accurate Local Surface Wind Speed: Statistical Modeling and Forecasts*, volume 254. Springer International Publishing, 2018.

[39] Q. Xu, D. He, N. Zhang, C. Kang, Q. Xia, J. Bai, and J. Huang. 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, 2015.

[40] D. Zheng, M. Shi, Y. Wang, A. T. Eseye, and J. Zhang. Day-ahead wind power forecasting using a two-stage hybrid modeling approach based on SCADA and meteorological information, and evaluating the impact of input-data dependency on forecasting accuracy. *Energies*, 10(12), 2017.

[41] J. M. Lima, A. K. Guetter, S. R. Freitas, J. Panetta, and J. G. de Mattos. A Meteorological–Statistic Model for Short-Term Wind Power Forecasting. *Journal of Control, Automation and Electrical Systems*, 28(5):679–691, 2017.

[42] M. Mana, M. Burlando, and C. Meißner. Evaluation of two ANN approaches for the wind power forecast in a mountainous site. *International Journal of Renewable Energy Research*, 7(4):1629–1638, 2017.

[43] B. Manobel, F. Sehnke, J. A. Lazzús, I. Salfate, M. Felder, and S. Montecinos. Wind turbine power curve modeling based on Gaussian Processes and Artificial Neural Networks. *Renewable Energy*, 125:1015–1020, 2018.

[44] T. Pinto, H. Morais, and J. M. Corchado. Adaptive entropy-based learning with dynamic artificial neural network. *Neurocomputing*, 338:432–440, 2019.

[45] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi. *Wind Energy Handbook*. 2011.

[46] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, OR, 1996. AAAI Press.

[47] H. P. Kriegel, P. Kröger, J. Sander, and A. Zimek. Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240, 2011.

[48] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems*, 42(3), 2017.

Appendix A

Wind Speed Models Configurations

Table A.1: Wind Speed Models ”A” configuration

Model	Number of Nodes in the Hidden Layer	Learning Rate
PA1	10	0.01000
PA2	10	0.00100
PA3	10	0.00010
PA4	10	0.00001
PA5	8	0.01000
PA6	8	0.00100
PA7	8	0.00010
PA8	8	0.00001
PA9	6	0.01000
PA10	6	0.00100
PA11	6	0.00010
PA12	6	0.00001
PA13	4	0.01000
PA14	4	0.00100
PA15	4	0.00010
PA16	4	0.00001

Table A.2: Wind Speed Models ”B” configuration

Model	Number of Nodes in the Hidden Layer	Learning Rate
PB1	20	0.01000
PB2	20	0.00100
PB3	20	0.00010
PB4	20	0.00001
PB5	18	0.01000
PB6	18	0.00100
PB7	18	0.00010
PB8	18	0.00001
PB9	15	0.01000
PB10	15	0.00100
PB11	15	0.00010
PB12	15	0.00001
PB13	12	0.01000
PB14	12	0.00100
PB15	12	0.00010
PB16	12	0.00001

Table A.3: Wind Speed Models ”C” configuration

Model	Number of Nodes in the Hidden Layer	Learning Rate
PC1	12	0.01000
PC2	12	0.00100
PC3	12	0.00010
PC4	12	0.00001
PC5	10	0.01000
PC6	10	0.00100
PC7	10	0.00010
PC8	10	0.00001
PC9	8	0.01000
PC10	8	0.00100
PC11	8	0.00010
PC12	8	0.00001
PC13	6	0.01000
PC14	6	0.00100
PC15	6	0.00010
PC16	6	0.00001

Table A.4: Wind Speed Error Models configuration

Model	Number of Nodes in the Hidden Layer	Learning Rate	Batch Size
1	20	0.00010	1
2	20	0.00010	10
3	20	0.00010	32
4	20	0.00001	1
5	20	0.00001	10
6	20	0.00001	32
7	24	0.00010	1
8	24	0.00010	10
9	24	0.00010	32
10	24	0.00001	1
11	24	0.00001	10
12	24	0.00001	32
13	30	0.00010	1
14	30	0.00010	10
15	30	0.00010	32
16	30	0.00001	1
17	30	0.00001	10
18	30	0.00001	32

Appendix B

Wind Speed Models Results

Table B.1: PA.12.TA results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PA4.12.TA	1.437	1.092	33.021
PA6.12.TA	1.439	1.094	33.892
PA1.12.TA	1.441	1.094	29.042
PA12.12.TA	1.444	1.097	28.890
PA8.12.TA	1.444	1.096	35.992
PA3.12.TA	1.444	1.097	33.236
PA5.12.TA	1.444	1.097	33.465
PA11.12.TA	1.444	1.098	33.461
PA10.12.TA	1.445	1.098	34.226
PA13.12.TA	1.446	1.099	32.510
PA9.12.TA	1.447	1.100	35.302
PA15.12.TA	1.447	1.100	32.841
PA7.12.TA	1.447	1.100	22.966
PA2.12.TA	1.447	1.100	28.544
PA14.12.TA	1.447	1.100	30.332
PA16.12.TA	1.448	1.100	27.381

Table B.2: PB.12.TA results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB10.12.TA	1.384	1.052	15.675
PB3.12.TA	1.388	1.058	15.525
PB12.12.TA	1.389	1.059	18.157
PB8.12.TA	1.389	1.058	15.726
PB5.12.TA	1.390	1.059	17.081
PB15.12.TA	1.390	1.060	17.214
PB7.12.TA	1.392	1.061	15.099
PB13.12.TA	1.392	1.062	18.190
PB14.12.TA	1.393	1.063	16.743
PB6.12.TA	1.393	1.063	14.213
PB16.12.TA	1.394	1.063	17.473
PB11.12.TA	1.395	1.065	17.456
PB9.12.TA	1.396	1.065	15.936
PB1.12.TA	1.396	1.065	20.049
PB2.12.TA	1.396	1.065	15.185
PB4.12.TA	1.398	1.067	15.697

Table B.3: PC.12.TA results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PC11.12.TA	1.390	1.057	15.034
PC3.12.TA	1.392	1.059	18.052
PC1.12.TA	1.393	1.059	18.849
PC4.12.TA	1.394	1.062	17.766
PC13.12.TA	1.395	1.061	15.954
PC5.12.TA	1.395	1.062	17.932
PC2.12.TA	1.395	1.062	18.000
PC6.12.TA	1.396	1.063	16.391
PC10.12.TA	1.396	1.063	14.923
PC7.12.TA	1.396	1.063	15.829
PC16.12.TA	1.396	1.063	16.817
PC8.12.TA	1.397	1.063	15.452
PC12.12.TA	1.398	1.065	17.115
PC9.12.TA	1.398	1.064	15.688
PC15.12.TA	1.399	1.066	14.905
PC14.12.TA	1.399	1.066	14.698

Table B.4: PB.24.TA results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
PB11.24.TA	1.459	1.106	23.269
PB5.24.TA	1.461	1.107	38.018
PB4.24.TA	1.461	1.107	38.144
PB10.24.TA	1.461	1.108	22.034
PB8.24.TA	1.461	1.107	25.083
PB14.24.TA	1.462	1.107	20.098
PB16.24.TA	1.463	1.111	18.879
PB7.24.TA	1.464	1.110	37.102
PB2.24.TA	1.464	1.110	37.205
PB12.24.TA	1.464	1.111	23.190
PB6.24.TA	1.465	1.112	34.725
PB15.24.TA	1.466	1.113	20.759
PB1.24.TA	1.466	1.113	38.613
PB3.24.TA	1.466	1.113	33.623
PB9.24.TA	1.467	1.114	21.015
PB13.24.TA	1.467	1.114	22.300

Table B.5: Turbine A error models results

Model	RMSE [m/s]	MAE [m/s]	Execution Time [s]
10	1.244	0.945	4.818
11	1.245	0.947	4.519
8	1.247	0.951	4.423
4	1.248	0.947	4.106
15	1.248	0.948	4.727
3	1.249	0.948	3.430
9	1.249	0.948	4.026
5	1.251	0.954	3.337
2	1.251	0.952	3.932
13	1.251	0.952	4.402
14	1.252	0.954	3.189
7	1.253	0.953	3.745
1	1.253	0.957	3.361
12	1.253	0.955	4.485
6	1.253	0.957	4.059
17	1.254	0.958	4.320
16	1.255	0.956	3.796
18	1.255	0.955	4.338

Appendix C

Electrical Load Models Results

Table C.1: Results of the A1 models for the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A1.4	242.39	174.03	180.29
A1.3	242.70	174.57	168.67
A1.2	243.15	175.66	182.00
A1.1	253.10	186.73	221.63
A1.5	256.80	182.16	235.83
A1.6	270.42	210.90	162.14
A1.7	475.26	344.22	49.71
A1.8	502.59	377.78	38.87

Table C.2: Results of the A2 models for the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A2.4	222.73	159.53	246.10
A2.2	224.22	160.96	195.67
A2.1	225.03	161.67	178.58
A2.5	242.35	170.77	226.15
A2.6	244.41	182.38	168.44
A2.7	272.26	208.88	94.76
A2.8	285.42	222.61	71.49
A2.3	398.73	294.36	148.53

Table C.3: Results of the A3 models for the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A3.6	291.71	226.74	111.85
A3.5	295.85	242.23	132.83
A3.7	315.52	231.52	81.90
A3.8	344.40	249.90	63.66
A3.1	563.28	449.06	2.93
A3.2	571.47	454.32	3.76
A3.3	580.09	460.27	3.58
A3.4	599.57	474.90	3.11

Table C.4: Results of the B1 models for the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B1.7	240.74	181.66	195.64
B1.4	242.25	175.15	180.94
B1.3	242.90	175.12	289.44
B1.2	248.88	182.27	371.79
B1.1	249.36	183.58	446.52
B1.6	254.98	189.24	302.09
B1.8	273.74	212.60	91.05
B1.5	365.09	271.97	363.77

Table C.5: Results of the B2 models for the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B2.2	223.12	160.18	249.42
B2.4	223.37	161.03	208.87
B2.1	225.61	163.17	230.61
B2.3	228.02	166.66	235.35
B2.6	250.92	178.65	127.01
B2.8	267.04	204.19	60.01
B2.7	267.28	196.28	70.16
B2.5	306.82	236.07	230.22

Table C.6: Results of the B3 models for the 10-minutes ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B3.6	311.72	239.61	49.77
B3.7	317.84	233.00	39.26
B3.8	331.41	247.05	32.46
B3.5	336.83	270.14	95.53
B3.1	615.41	504.54	1.97
B3.2	621.48	507.54	1.77
B3.4	631.64	513.84	1.50
B3.3	634.12	514.87	1.60

Table C.7: Results of the A1 models for the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A1.1	572.21	465.41	91.84
A1.3	573.56	466.60	44.24
A1.2	574.02	467.16	39.08
A1.4	673.20	536.27	25.81
A1.5	712.01	595.18	157.34
A1.8	771.64	636.35	50.05
A1.7	805.80	667.98	52.85
A1.6	846.46	693.15	59.32

Table C.8: Results of the A2 models for the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A2.6	463.31	364.12	228.18
A2.1	465.41	375.39	171.29
A2.3	465.46	376.36	224.66
A2.2	465.67	375.77	181.96
A2.7	466.20	367.29	193.99
A2.4	467.42	378.38	217.47
A2.5	482.71	379.58	225.62
A2.8	678.55	535.40	78.93

Table C.9: Results of the A3 models for the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A3.6	550.35	449.76	54.43
A3.7	560.49	461.66	43.26
A3.8	572.15	471.17	32.07
A3.2	591.09	470.79	1.99
A3.5	591.27	474.10	86.33
A3.3	603.45	480.31	1.84
A3.4	605.37	481.66	1.60
A3.1	1860.83	1761.30	72.15

Table C.10: Results of the B1 models for the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B1.7	387.76	307.92	308.68
B1.6	417.97	329.22	272.42
B1.8	538.71	447.98	128.35
B1.5	549.19	441.91	212.19
B1.1	572.03	465.67	51.27
B1.2	573.49	467.95	53.75
B1.3	573.98	469.01	69.29
B1.4	574.39	469.37	62.79

Table C.11: Results of the B2 models for the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B2.8	348.96	271.82	194.76
B2.7	356.58	276.64	205.38
B2.6	386.33	298.88	263.97
B2.1	464.69	375.80	465.34
B2.2	465.43	376.69	451.10
B2.3	467.40	378.66	251.86
B2.4	468.96	380.31	237.60
B2.5	472.04	370.68	409.50

Table C.12: Results of the B3 models for the 1-hour ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B3.5	526.16	425.09	98.43
B3.6	529.94	424.47	54.06
B3.7	546.73	439.21	41.90
B3.8	560.90	453.75	34.80
B3.1	632.82	518.71	2.56
B3.2	634.81	519.62	2.10
B3.3	648.97	527.27	1.90
B3.4	660.83	534.17	1.69

Table C.13: Results of the A1 models for the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A1.4	663.39	535.40	46.75
A1.3	663.43	535.46	51.15
A1.2	663.98	536.00	106.52
A1.1	667.36	539.02	164.66
A1.5	746.74	607.26	462.32
A1.6	795.18	653.65	346.18
A1.8	866.14	707.79	53.61
A1.7	867.05	705.31	69.46

Table C.14: Results of the A2 models for the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A2.1	583.56	465.15	103.06
A2.2	587.93	468.29	64.87
A2.3	593.15	471.88	52.65
A2.4	597.27	474.39	49.15
A2.6	663.67	508.49	247.40
A2.5	670.57	525.36	245.93
A2.7	688.58	536.14	206.75
A2.8	719.36	577.17	78.72

Table C.15: Results of the A3 models for the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A3.1	619.72	494.56	2.26
A3.2	621.48	495.45	1.93
A3.5	628.11	502.14	87.88
A3.3	635.49	504.76	1.75
A3.6	637.86	506.67	46.99
A3.4	646.70	513.25	1.39
A3.7	648.41	513.96	35.53
A3.8	672.59	531.08	27.57

Table C.16: Results of the B1 models for the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B1.2	661.66	532.41	62.83
B1.1	662.25	533.30	74.83
B1.4	663.77	532.84	38.47
B1.6	673.77	541.15	176.30
B1.3	742.62	589.03	25.17
B1.5	774.78	626.45	190.31
B1.7	805.36	633.59	79.02
B1.8	867.87	685.25	36.02

Table C.17: Results of the B2 models for the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B2.1	585.58	461.69	84.49
B2.4	592.77	458.75	93.00
B2.3	592.80	458.01	98.94
B2.2	593.05	461.49	65.38
B2.7	615.31	481.40	175.85
B2.6	639.22	502.54	165.94
B2.8	652.77	520.81	159.71
B2.5	697.02	567.22	180.31

Table C.18: Results of the B3 models for the 12-hours ahead forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B3.5	642.79	506.52	256.81
B3.1	646.44	532.24	5.20
B3.2	653.44	535.95	2.81
B3.8	656.59	513.62	67.40
B3.7	659.59	515.65	97.47
B3.3	659.69	539.60	3.52
B3.4	669.06	545.41	2.67
B3.6	669.15	522.06	104.85

Table C.19: Results of the ”A” models for the 10-minutes ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A.6	212.10	139.26	35.74
A.7	212.78	139.83	27.39
A.8	213.22	140.36	18.37
A.2	214.17	140.09	6.55
A.3	214.37	139.60	6.75
A.4	214.40	140.23	4.34
A.1	214.53	140.03	11.34
A.5	214.90	143.50	21.50

Table C.20: Results of the ”B” models for the 10-minutes ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B.8	179.61	133.94	21.75
B.6	179.67	133.87	36.83
B.7	180.36	134.43	20.96
B.5	180.42	134.05	70.47
B.4	183.34	135.73	10.75
B.2	183.47	135.83	14.13
B.3	183.72	136.06	10.65
B.1	183.81	136.00	29.04

Table C.21: Results of the ”C” models for the 10-minutes ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
C.8	172.22	120.20	23.97
C.3	172.50	119.54	22.92
C.4	172.99	119.96	21.79
C.1	173.84	121.99	35.80
C.2	173.87	121.99	18.90
C.7	174.07	123.07	29.30
C.6	176.50	124.52	25.25
C.5	181.52	128.36	43.52

Table C.22: Results of the ”A” models for the 1-hour ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A.4	319.63	252.31	22.62
A.8	319.85	251.56	160.46
A.3	320.08	252.69	24.74
A.5	320.29	252.43	210.58
A.7	320.29	252.12	118.18
A.6	320.35	252.47	136.10
A.2	320.66	253.30	22.73
A.1	321.73	254.11	33.56

Table C.23: Results of the ”B” models for the 1-hour ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B.1	290.12	236.72	12.54
B.3	290.26	236.82	9.71
B.2	290.30	236.90	10.25
B.4	290.51	236.99	6.28
B.8	294.06	239.01	22.15
B.7	294.93	239.59	25.73
B.5	295.68	239.26	43.01
B.6	296.79	240.91	21.57

Table C.24: Results of the ”C” models for the 1-hour ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
C.1	284.61	230.97	11.09
C.3	284.69	231.16	5.96
C.2	284.71	230.96	7.77
C.4	284.78	231.36	5.45
C.8	292.07	237.70	13.79
C.7	292.27	237.80	16.29
C.5	292.53	236.47	36.21
C.6	294.02	238.64	21.72

Table C.25: Results of the ”A” models for the 12-hours ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
A.1	480.29	380.19	53.31
A.4	480.31	380.35	29.67
A.2	480.37	380.34	25.91
A.3	480.42	380.35	27.45
A.5	480.72	384.38	78.71
A.6	481.43	384.26	27.17
A.7	481.49	384.04	23.90
A.8	481.50	384.17	16.42

Table C.26: Results of the ”B” models for the 12-hours ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
B.7	417.98	347.61	59.39
B.4	418.44	348.27	10.62
B.3	418.78	348.52	12.89
B.1	419.21	348.85	14.30
B.2	419.86	349.67	9.59
B.8	421.30	349.82	33.79
B.6	422.66	350.87	51.86
B.5	432.32	357.77	57.48

Table C.27: Results of the ”C” models for the 12-hours ahead error forecasts

Model	RMSE [kW]	MAE [kW]	Execution Time [s]
C.3	405.68	332.63	15.17
C.2	406.67	334.04	8.24
C.4	406.72	334.02	8.90
C.1	407.42	334.82	12.86
C.8	416.01	342.91	33.52
C.6	420.33	345.90	43.05
C.7	421.29	346.06	27.81
C.5	430.50	352.54	48.93