TÉCNICO LISBOA

Wind Forecast
at Medium Voltage Distribution Networks

Herbert Amezquita Ortiz

Thesis to obtain the Master of Science Degree in

Energy Engineering and Management

Supervisors: Prof. Hugo Gabriel Valente Morais
Prof. Pedro Manuel Santos de Carvalho

Examination Committee

Chairperson: Prof. Duarte de Mesquita e Sousa
Supervisor: Prof. Hugo Gabriel Valente Morais
Member of the Committee: Prof. Rui Manuel Gameiro de Castro

October 2022

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

This work was created using LaTeX typesetting language
in the Overleaf environment (www.overleaf.com).

Acknowledgments

Firstly, I would like to express my sincere gratitude to my supervisors, Professor Hugo Morais and Professor Pedro Carvalho for taking me on board of the project at INESC-ID and for their guidance, insight and constantly sharing of knowledge. It has been a pleasure working with you.

I would also like to thank my parents and my brother for their support, encouragement and caring over all these years, for always being there for me, even when I am pretty far from home and without whom this project would not be possible.

Last but not least, to all my friends and colleagues that were always there during this journey of a master programme in Lisbon, Portugal.

This work was partially supported by E-REDES, the Portuguese Distribution System Operator under the scope of the project CAST-72 with reference number BI2022/273.

To each and every one of you – Thank you.

Abstract

Due to the intermittent and variable nature of wind, Wind Power Generation Forecast (WPGF) has become an essential task for power system operators, who are looking for a reliable wind penetration into the electric grid. Since there is a need to forecast wind power generation accurately, the main contribution of this thesis is the development, implementation and comparison of WPGF methods to be used by Distribution System Operators (DSOs). The methodology applied comprised five stages namely, pre-processing, feature selection, forecasting models, post-processing and validation. For training and testing the models, historical wind power generation data (measured at secondary substations) of 20 wind farms connected to the Medium Voltage (MV) distribution network was provided by the Portuguese DSO, while meteorological data was obtained from IPMA and ISTMeteo.

After comparing the accuracy of eight different models in terms of their Relative Root Mean Square Error (RRMSE), Extreme Gradient Boosting (XGBOOST) appeared as the best-suited forecasting method for wind power generation. Thus, XGBOOST was chosen for further tests and improvements (tuning) in order to reduce the error as much as possible. At the end, the best average RRMSE achieved by the proposed XGBOOST model for 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021) corresponds to 13.48%, outperforming the predictions of the Portuguese DSO by more than 20%, which for the same period of analysis present a RRMSE of 16.88%.

Keywords

Extreme Gradient Boosting (XGBOOST), Medium Voltage Distribution Network, Short-Term Forecasting, Wind Power Generation Forecast

Resumo

Devido à natureza intermitente e variável do vento, a Previsão da Geração de Energia Eólica (PGEE) tornou-se uma tarefa essencial para os operadores dos sistemas de energia, que procuram uma penetração confiável do vento na rede elétrica. Uma vez que existe a necessidade de prever com precisão a geração eólica, a principal contribuição desta tese é o desenvolvimento, implementação e comparação de metodologias de PGEE a serem utilizadas pelos Operadores da Rede de Distribuição (ORD). A metodologia desenvolvida compreende cinco etapas, nomeadamente pré-processamento, seleção das variáveis, modelos de previsão, pós-processamento e validação. Para o treino e teste dos modelos, foram fornecidos dados históricos de geração eólica (medidos nas subestações secundárias) de 20 parques eólicos ligados à rede de distribuição de média tensão, fornecidos pelo operador do sistema de distribuição de Portugal, enquanto os dados meteorológicos foram obtidos do IPMA e do ISTMeteo.

Após a comparação da precisão de oito modelos em termos do erro quadrático médio relativo (RRMSE), o Extreme Gradient Boosting (XGBOOST) foi escolhido como sendo o método mais adequado para a PGEE, no dataset utilizado. Assim, XGBOOST foi escolhido para a realização de testes mais aprofundados e melhorias na sua parametrização com o objetivo de reduzir ao máximo o erro das previsões. O melhor desempenho alcançado pelo modelo XGBOOST proposto, considerando a análise dos valores RRMSE, para 1 ano de treino (JAN-DEZ de 2020) e 6 meses de previsão (JAN-JUN de 2021) foi de 13.48%, superando em mais de 20% as previsões do ORD, que para o mesmo período de análise apresentam um RRMSE de 16.88%.

Palavras Chave

Previsão de Curto Prazo, Previsão de Geração de Energia Eólica, Rede de Distribuição de Média Tensão, Extreme Gradient Boosting (XGBOOST)

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
ANFIS	Adaptative Neural Fuzzy Inference System
ANN	Artificial Neural Networks
AR	Auto-Regressive
ARX	Auto-Regressive with Exogenous Variable
ARIMA	Auto-Regressive Integrated Moving Average
ARMA	Auto-Regressive Moving Average
BPNN	Back-Propagation Neural Network
CART	Classification and Regression Tree
CV	Cross Validation
DSO	Distribution System Operator
DT	Decision Trees
EDA	Exploratory Data Analysis
EU	European Union
FFNN	Feed Forward Neural Network
GIS	Geographical Information System
IPMA	Instituto Português do Mar e da Atmosfera
LGBM	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory
MA	Moving Average
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

ML	Machine Learning
MLP	Multilayer Perceptron
MM5	Mesoscale Model Fifth Generation
MSE	Mean Square Error
MV	Medium Voltage
NNWT	Neural Network Wavelet Transform
NWP	Numerical Weather Prediction
PV	Solar Photovoltaic
RBFNN	Radial Basis Function Neural Network
REN	Redes Energéticas Nacionais
RES	Renewable Energy Sources
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RRMSE	Relative Root Mean Square Error
SDE	Standard Deviation Error
SVM	Support Vector Machine
TCN	Temporal Convolutional Network
TEPCO	Tokyo Electric Power Company
TSO	Transmission System Operator
WEP	Weather Ensemble Predictions
WPGF	Wind Power Generation Forecast
WT	Wavelet Transform
XGBOOST	Extreme Gradient Boosting

1

Introduction

Contents

1.1 Motivation

Nowadays, the world is going through an energy transition process from fossil fuels to renewable energies, that aims to reduce the environmental impact of the energy sector. Since power generation from conventional units are large contributors to the emission of greenhouse gases, using Renewable Energy Sources (RES) instead of coal and oil-fired power plants have become a global trend during the last decades [1].

To increase the penetration rate of RES in power systems, significant incentive schemes and policies have been considered by governments. For example, the European Union (EU) under the 2030 climate and energy framework[1] for the period 2021-2030 is part of the ambitious European Green Deal. This framework commits the EU to reduce greenhouse emissions by at least 40% (as compared to 1990 levels), to increase the amount of renewable energy in the energy mix by at least 32% and to improve energy efficiency by at least 32.5% [2]. To achieve those targets, the penetration of RES such as solar (photovoltaic and concentrated thermal), wind, hydropower, geothermal, biomass, biofuels, waves or tidal must continue growing at an accelerated rate.

Over the last years, a rapid expansion of Solar Photovoltaic (PV) and wind has been seen mainly because the cost of PV and wind power installations has declined sharply. Out of all available RES, PV and wind are considered now the most abundant, developed, economically viable and commercially accepted worldwide [3].

Without considering hydropower, wind has the higher installed capacity of the renewables. According to the Global Wind Report 2021[2], year 2020 was the best year in history for the global wind industry. The report shows a year-over-year growth of 53% considering that for 2020 more than 93 GW of wind power were installed, with 86.9 GW allocated to the onshore market and 6.1 GW to the offshore market [4]. The new installations brings the global cumulative wind power capacity up to 743 GW. Regarding Europe, a steady growth was recorded with the Netherlands taking the lead and followed by Belgium and Germany for offshore installations while Spain, France and Germany were the leaders for onshore installations [4].

In Portugal, wind energy also plays an important role. Since 2000 wind industry has seen a continuous growth year by year, motivated by a political strategy at European and national levels and with the aim of diversifying sources, improving the security on supply, decreasing the energetic dependency and reducing the environmental footprint [5].

[1] https://climate.ec.europa.eu/eu-action/climate-strategies-targets/2030-climate-energy-framework_en
[2] https://gwec.net/global-wind-report-2021

In 2019 for instance, renewable energy covered more than 50% of Portugal's electricity needs and 23% of that energy came from wind energy alone, placing Portugal in the top European countries for wind energy. The Portuguese government has also pledged to one of the most ambitious 2030 targets in the EU: it wants renewable energy to cover 80% of the country's electricity needs by 2030, of which 31% would come from wind [6].

That means that wind power generation will continue growing exponentially in the next years and not only in Portugal but also worldwide. However, the uncertainty in wind power generation is very large due to the inherent variability of wind speed [7]. Then, wind variability needs to be understood by operators of power systems and wind farms in order to ensure that supply and demand are balanced and the power network operates without constraints.

Since supply and demand should be equal at all times but wind power generation depends on the availability of wind, that is a weather dependent source, the integration into the existing electrical supply system brings some challenges at the level of secondary substations that need to be addressed by Distribution System Operators (DSOs) of power networks.

Some of the challenges include system stability and reliability, due to grid congestion or intermittency of supply; system balance, that requires a strong information exchange between the DSO and the Transmission System Operator (TSO) or flexibility services (voltage support and demand-side response) to ensure that the network is stabilized amid the varying energy generation and consumption. Other challenges associated to optimise the grid include technical imbalances in existing equipment and saturations in the Medium Voltage (MV) network or in the substations [8].

Here is where Wind Power Generation Forecast (WPGF) appears as one of the most efficient ways to overcome some of these problems and to help the power system operators to reduce the risk of unreliable electricity supply. Weather variables such as wind speed, wind direction, temperature, pressure and humidity, among others, influence wind power generation. The development of new techniques to improve understanding of these variables, through simulation, forecasting, distribution curve fitting, filtering and modeling, allows making better decisions about expansion of the wind sector and better management of the electricity system [9].

Additionally, accurate estimation of wind speed and wind power generation might improve the safety, reliability and profitability not only in the operation of the wind farms but also in the secondary substations managed by DSOs.

WPGF accuracy is directly connected to the need for balancing energy and hence to the cost of wind power integration. Consequently, a large amount of research has been directed towards the development and improvement of good and reliable wind forecasts in recent years and different forecasting systems with different approaches have been developed [10]. A comprehensive review of the literature about wind speed and wind power forecasting is presented in Chapter 2 'State of Art'.

1.2 Objectives

The main objective of this thesis is to develop and implement a framework with several forecasting models for wind power generation in wind farms connected to secondary substations of the MV distribution network of Portugal. The models are implemented and compared using Python, to determine which method gives the lowest percentage of error between predictions and measured values. The final model needs to be efficient, which means accurate and run in a short computation time.

Specifically Persistence, Auto-Regressive (AR), Auto-Regressive with Exogenous Variable (ARX), Long Short-Term Memory (LSTM) neural network, Extreme Gradient Boosting (XGBOOST), Random Forest (RF), Decision Trees (DT) and Support Vector Machine (SVM) models are developed and tested using real data measured at the secondary substations and provided by the Portuguese DSO.

This data covers seven years of information (2015-2021) of power generated by 20 wind farms in Portugal mainland. It also includes the DSO predictions for the years 2020 and 2021, that are used to compared with our models results (through an error metric). The final goal of this work is to improve the DSO performance by reducing the error as much as possible.

Different meteorological parameters that might influence the forecast results like temperature, radiation, wind speed or wind direction are also considered into the models and that weather data comes from two different sources, one is the Instituto Português do Mar e da Atmosfera (IPMA)[3] and the other one is the meteorological investigation group of IST: meteoTécnico[4] (also refer as ISTMeteo throughout this document). For IPMA two years of meteorological data are available for the analysis, specifically 2020 and 2021; while for ISTMeteo just seven months of 2021 (from June to December) are available. Both sources of data are compared, to identify which one offers better results (meaning better data quality and lower percentage of error) and the best option is used to run the models.

[3] https://www.ipma.pt/pt/index.html
[4] https://meteo.ist.utl.pt/fdata.php

It is important to mention that performing the predictions at MV level presents several challenges comparing with methods already proposed. In comparison with forecast methods proposed at wind farm level, in MV, the information regarding wind farms does not exist. Additionally, only the Numerical Weather Prediction (NWP) in areas of $14\ km^2$ is available, which is less accurate when compared with information available exactly at the wind farms. In comparison to the forecast at regional or national level, the prediction at MV level is more complicated because only one wind farm is considered, which means that the error in the forecast has a direct impact on the accuracy of the model. When the forecast models include several wind farms, the error in the power generation forecast of a wind farm can be minimal compared with the global system.

1.3 Organization of the Document

After this introductory chapter, the remainder of the thesis is organized as follows:

- Chapter 2: State of Art, presents a literature review related to wind power and wind speed forecasting. Regression, Artificial Intelligence (AI) and Machine Learning (ML) methods are explained and a theoretical background of the concepts necessary to understand the work is detailed.
- Chapter 3: Methodology, explains systematically how the work was done, starting from data sets used, pre-processing of the initial data, Exploratory Data Analysis (EDA) and feature selection, implementation of the forecasting models, post-processing and validation conducted.
- Chapter 4: Results and Discussion, shows the forecast results obtained for each method and the comparison in terms of error performance between them and also with the DSO predictions provided. It also includes the different tests or improvements performed to the final method chosen, in order to reduce the error as maximum as possible.
- Chapter 5: Conclusions, summarizes the main outcomes of the thesis, the limitations encountered in the process and suggests future work related to the topic.

2

State of the Art

Contents

Wind speed and wind power generation forecasting have been a topic of interest for many researchers during the recent years, due to importance of integrating RES to the power system and all the implications that it brings. This section presents a review of regression, AI and ML forecasting methods and a general overview of different publications and studies related to wind power generation and wind speed forecasting (based on time scales).

2.1 Wind Forecast Classification

A forecast system is characterized by its time horizon, which is the future time period for which the wind generation or wind speed will be predicted. There is not a strict classification and the time interval defined for each category varies between different authors. Based on *C.Monteiro et al* [11], wind forecasting can be separated according to the prediction horizon, into the following categories:

- Very-short-term forecasting: Few seconds to 30 minutes ahead.
- Short-term forecasting: 30 minutes to 6 hours ahead. Mainly useful for operational purposes (economic load dispatch planning, load increase/decrease decisions).
- Medium-term forecasting: 6 hours to 1 day ahead. Aim to increase operational security of day ahead electricity markets and corroborate online/offline decisions.
- Long-term forecasting: Multiple days ahead to 1 year or more. Provide information for power system risk assessment and also to identify potential for wind power generation in specific areas, providing valuable data for energy planners [9].

When specifying a wind power prediction model the desired time horizon dictates the final choice. Different models are differently suited to certain power system planning and market activities, which occur over different time scales [12].

2.2 Wind Forecast Methods

Based on the analysis of the literature, wind forecast methods can be divided into six overall groups: Persistence method, Physical methods, Statistical methods, Artificial Neural Networks (ANN) based models, Hybrid methods and New models. Persistence method is normally used to benchmark other methods. Physical methods use forecast values from a NWP model as an input to calculate the wind power generation using the power curve, while statistical methods are based upon analysis of historical time series of wind.

ANN make non-linear relationships between input features and output data. Hybrid methods are a combination of different methods and new models refer to some novel models developed in recent years. The six groups are detailed in the next subsections.

2.2.1 Persistence Method

The simplest method to forecast wind speed or wind power is to use persistence. It is also called the naive predictor and it is common between time series forecasting models. This method uses the simple assumption that the value at a certain future time will be the same as it is when the forecast is made or it can be the average of past values.

It is based on the assumption of a high correlation between present and future values and it produces accurate predictions for very-short term forecasts [13]. As expected, the accuracy of this model degrades rapidly with the increasing prediction lead time [14], so it is normally used as a reference to evaluate the performance of advanced methods. Any advanced forecasting is worth implementing, only if it out performs the persistence model [15].

When looking at the literature, *S.Dutta et al* [16] presents a study that uses persistence method for short-term electrical demand, PV power and wind power forecasting. In this case persistence was selected because wind power varies very frequently and there is no fixed daily pattern for wind generation. Unlike most other forecasting algorithms, persistence relies neither on weather forecast data nor on in-built toolboxes in software for implementation. The results of the study showed that the accuracy of the forecast using persistence depends on two factors: the look-back time and the extent of change of the data over time. Thus, the persistence algorithm may be improved by using previous day patterns along with more recent historic data and assigning weighting factors to account for variation of data over time.

2.2.2 Physical Methods

Physical systems use parameterizations that describes the physical relationship between atmospheric condition, wind, speed, local topography and the output from the wind power plant [15]. The idea is to refine the real resolution of the Numerical Weather Prediction (NWP) model, in order to achieve an accurate prediction of the weather.

NWP models are the most important component in wind power prediction, they represent the first input to any wind power prediction system and understanding the uncertainty in the NWP model will ultimately help to understand the uncertainty in the WPGF [17].

NWP are mathematically complex and are usually run on super computers because require a high computation time to produce forecasts, which limits it usefulness. Hence, the performance of physical models is often satisfactory for more than 6 hours ahead time horizons (medium to long-term forecasting). They are inappropriate for short-term prediction (several minutes to a few hours) due to difficulty of information acquisition and computation time [14].

The NWP system usually provides wind speed forecasts for a grid of surrounding points around the wind generators and the physical approach uses a meso-scale or micro-scale model for the downscaling, which interpolate these wind speed forecasts to the level of the wind generator. For running the downscaling model, it is necessary to have a detailed description of the terrain surrounding the wind generators. However, collecting the information of terrain conditions is one of the main difficulties in the implementation of physical models [14].

The refined wind speed data at the hub height of the wind turbines is then plugged into the corresponding wind power curve to calculate the wind power generation. Predicting the wind power output from each individual wind farm can be time consuming, so instead an approach called upscaling is used. In upscaling, the wind power output from a sample number of wind farms forms the basis of reference data. Upscaling can have the apparent effect of reducing forecast error because it becomes averaged over the whole region [12]. The basic process followed by a physical method is illustrated in Figure 2.1.

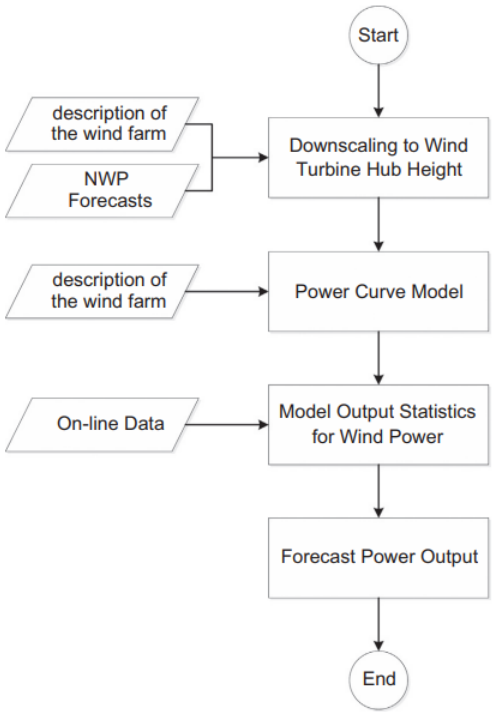


Figure 2.1: Physical approach to forecast wind power [18]

In the literature there are several papers focused on physical methods, for example *J.Taylor et al* [7] developed a new type of physical method to predict the probability density function of wind power generation for 1-to-10 days ahead forecast using Weather Ensemble Predictions (WEP). WEP are generated from atmospheric models and consist of multiple scenarios for the future value of a weather variable (in this case wind speed). The results of the forecast were compared with the statistical time series method Auto-Regressive Moving Average (ARMA) and it was found that WEP gave more accurate and comfortably superior results, therefore, the author mentions that WEP have a strong potential for WPGF.

Another interesting study focused on offshore wind energy potential in the supply area of Tokyo Electric Power Company (TEPCO) was investigated by *A.Yamaguchi et al* [19]. It uses a Mesoscale Model Fifth Generation (MM5) to investigate the wind climate and its spatial distribution and a Geographical Information System (GIS) to consider the social and economic criteria. The results of this research showed that wind climate (wind speed and wind direction) predicted by MM5 model are in good agreement with the observation and the prediction error of annual mean wind speed was 4.8%. Concerning the economical and social criteria, the available potential becomes 94 TWh/year, accounting for 32% of the annual demand of TEPCO.

Despite the major progress made by NWP in the last decades, meteorological models are usually unable to provide reliable surface wind speed forecasts, especially in complex topography regions, because of shortcomings in horizontal resolution, physical parameterisations and initial and boundary conditions [20]. In order to reduce these drawbacks, *F.Cassola et al* [20] applies a Kalman filtering procedure to locally adjust the low-resolution numerical prediction of wind speed and wind power genration of a NWP model at the wind farm site of Varese Ligure (Northern Italy). The Kalman filter is an algorithm that provides an efficient computational (recursive) mean to estimate the state of a process, in a way that minimises the mean of the square error. The procedure was tested with wind speed and wind energy output data from a wind farm located between two anemometric stations and the results obtained showed that this methodology is capable to provide significant forecast improvements with respect to model direct outputs, leading to the elimination of systematic errors.

2.2.3 Statistical Methods

The statistical approach is based on training with measured data (time series). This methods are mostly used for short-term forecasting because the accuracy of the predictions drops significantly when the time horizon is extended and since they provide timely predictions based on patterns, the errors are minimized if the patterns are met with the historical ones.

Typical time series models are developed based on historical values. They are easy to model and capable to provide timely prediction [14]. According to the forecasting process, which was proposed by Box–Jenkins, to make a mathematical model of the problem four steps, which include model identification, model estimation, model diagnostics checking and forecasting are necessary [18].

Several types of time series models may be considered, but the most popular is Auto-Regressive (AR) and its variants Auto-Regressive with Exogenous Variable (ARX), Auto-Regressive Moving Average (ARMA) and Auto-Regressive Integrated Moving Average (ARIMA). Moreover, some of the papers that can be found in the literature regarding time series wind forecast are the following:

C.Gallego et al [21] presents a study focused on modelling the influence of local wind speed and wind direction on the dynamics of a wind power time series, using a generalized linear AR model. What they found is their study is that local measurements of both wind speed and wind direction provide useful information for a better comprehension of wind power time series dynamics, at least when considering the case of very-short term forecasting. In particular, local wind direction contributes to model some features of the prevailing winds, such as the impact on wind variability, whereas the non-linearities related to the power transformation process can be introduced by considering local wind speed.

A study made by *M.Duran et al* [22] tested an ARX model for wind power prediction using wind speed as exogenous variable. The results for a 24 hours time horizon showed a significant improvement in accuracy, when comparing the mean error of their model with persistence and a traditional AR model. According to [22], when compared with AR the improvement of ARX is about 14% and about 26% when compared with persistence.

J.Torres et al [23] presents an ARMA model to predict hourly average wind speed. In this study it was necessary to carry out a transformation and standardization of the time series in order to adjust the ARMA model, given the non-Gaussian nature of the hourly wind speed distribution and the non-stationary nature of its daily evolution. Regarding the results, ARMA model outperformed persistence; in fact, the errors for ARMA are 12% to 20% smaller than for persistence for a 10 hours forecast. However, for a forecasting horizon of 1 hour, persistence model had less errors than ARMA model.

Another study made by *M.Milligan et al* [24] applied an ARMA model to both wind speed and wind power output, to investigate the extent to which time series analysis can improve on simplistic persistence forecasts. Results are presented for operating wind farms in Iowa and Minnesota and indicate that a significant improvement over persistence model is sometimes possible but in some cases there is no

improvement when changing the order of the AR and the Moving Average (MA). Thus, the performance of the model is highly dependent on the parameters.

E.Yatiyana et al [25] mentions in their paper that wind speed has a significant influence on the power generated, since wind power has a cubic relationship with wind speed. Also, different wind flow direction has dynamic effects on the power flow, as related mechanical systems responses play a vital role to recover the maximum kinetic energy. This study focuses on an ARIMA model to predict wind speed and wind direction, with the aim of improving the accuracy of the power forecast. The results obtained showed that the forecasting error is less than 5% for wind speed and 16% for wind direction, when compared with the real values of wind speed and wind direction collected for a seven days period on a site located in Western Australia.

One last study related with time series wind forecast was made by *R.Kavasseri et al* [26] and it examines the use of fractional$-$ARIMA models to forecast wind speeds on the day-ahead $(24h)$ and two-day-ahead $(48h)$ horizons, using records obtained from four potential wind generation sites in North Dakota. Fractional$-$ARIMA or $f-$ARIMA model arises as a special case of ARIMA processes where the differencing parameter d assumes fractionally continuous values in the range $(-0.5,\ 0.5)$. The results of this study suggest that significant improvements in forecast accuracy of 42% in average are obtained with the proposed models compared to the persistence method.

2.2.4 Artificial Neural Networks (ANN)

ANN are one of the most commonly used methods for wind power forecast, since they can identify the non-linear relationships between input features and output data. The term neural network comes from the fact that these models were inspired by biological brains in the sense of how they process information [27].

ANN are typically composed of nodes (or neurons) that are distributed across different layers, namely input, hidden and output layers. 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 resembles the nervous system [27]. If the desired output is known at the beginning of the process it is called supervised; contrarily, it is called unsupervised [28].

The typical structure of an artificial neural network model for wind speed or wind power forecasting is presented in Figure 2.2.

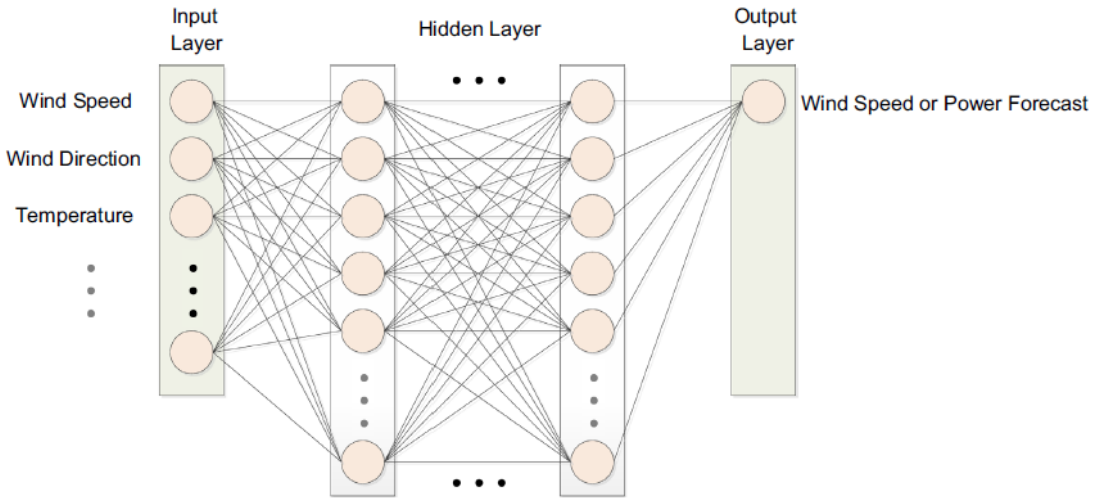


Figure 2.2: Typical ANN model structure [18]

The performance of ANN is dependent on many different factors, including data pre-processing, data structure, learning method or the connections between input and output data. Designing a neural network requires dealing with two steps: first, the selection of the proper structure of the network, where the direction of the passed information is specified and second, picking the right learning algorithm among supervised, unsupervised or reinforcement learning [28].

There are several kinds of ANN but the most common neural networks used for wind forecasting are: Feed Forward Neural Network (FFNN), where the data passes through the input nodes and exit on the output nodes, Back-Propagation Neural Network (BPNN), that tunes the weights of the neural network based on the error rate obtained in the previous iteration or Recurrent Neural Network (RNN), that takes information from prior inputs to influence the current input and output. Regarding the last type, since RNN suffers from short-term memory, a more advanced version of RNN called LSTM neural network is commonly used due to its effectiveness in learning long-term dependencies between time steps of sequence data.

Now looking in the literature, some of the papers that employ ANN for wind speed or wind power forecasting are the following:

A study from *A.More et al* [29] uses ANN to forecast daily, weekly and monthly wind speeds at two coastal locations in India. Both FFNN and RNN are tested and compared with ARIMA, since occurrence of the wind is highly uncertain in time and space. The forecasting results of the ANN models were fairly close to the corresponding measurements over one month, one week and one day time step, with average errors restricted to 4.3%, 5.4% and 6.3%, respectively. According to [29], the accuracy

of the forecast decreased as the interval forecasting period was reduced from one month to one day. Moreover, the superiority of one neural network over the other was not decided; but ANN definitively produced much more accurate forecasts than the traditional stochastic time series model ARIMA.

The paper of *J.Catalao* [30] presents a successful application of ANN in combination with Wavelet Transform (WT) for short-term wind power forecasting in Portugal. Historical wind power data available at Redes Energéticas Nacionais (REN) website are the main inputs to train the model and no exogenous variables are considered. The model proposed, called as Neural Network Wavelet Transform (NNWT), predicts the value of wind power for 3 hours ahead and it is compared with persistence, ARIMA and other neural network approach. The results of the study confirmed that this model is novel and effective since the Mean Absolute Percentage Error (MAPE) has an average value of 6.97%, outperforming the other methods analysed in [30]. Also, the introduction of the wavelet transform enables a reduction of the error when compared with the normal neural network.

M.Mabel et al [31] developed an ANN model to forecast wind power generation of seven wind farms in Muppandal, India. A BPNN is implemented using three input variables: wind speed, relative humidity and generation hours. Wind speed has direct influence on power generation, relative humidity affects the air density and this in turn affects power generation and generation hour is also an important parameter, where maximum generation hours should be obtained during the seasonal period by reducing the down time of the wind turbine by all possible mean. The model accuracy is evaluated then by comparing the predicted power with the actual measured values, using two years of training and one year of forecast. The results are satisfactory and in agreement, since the overall percentage error obtained was 4%.

2.2.5 Hybrid Methods

The combination of different forecasting methods is called hybrid approach. The main aim of this method is to retain the merits of each technique and improve the overall accuracy [28]. Combining forecasting models does not always perform better than the best individual model, however, in some cases it is viewed as less risky to combine forecasts than to select an individual forecast [18].

In a hybrid method different types of combinations can be found. Among the most popular are:

- Combination of physical and statistical methods.
- Combination of models for short-term and medium-term forecasting.
- Combination of alternative statistical methods.

Furthermore, some of the studies that can be found in the literature about hybrid methods for wind speed and wind power forecasting are presented in the next paragraphs:

A study from *S.Khazaei* [32] presents a hybrid approach for short-term wind power forecasting using the historical data of Sotavento wind farm (located in Spain) and NWP data obtained from Meteogalicia numerical weather forecast system. The goal of the study is to forecast the wind power for the next 24 hours, which is carried out through three stages: wind direction forecast, wind speed forecast, and wind power forecast. In all three phases, the same hybrid method is used, and the only difference is in the input data set. Outlier detection, decomposition of time series using WT, feature selection and prediction of each time series decomposed using Multilayer Perceptron (MLP) neural network constitute the main steps of the proposed method and the results obtained demonstrate it has a very high accuracy.

X.Qin et al [33] proposes an online clustering algorithm for wind speed forecasting that combines persistence method and a Radial Basis Function Neural Network (RBFNN). The combination of both approaches in a hybrid model is decided due to its complementary, RBFNN method is more suitable for monotonically changes of wind speed while persistence method is more suitable for random data with much noise. The proposed algorithm is tested in an actual wind farm in XinQing, China to predict one hour ahead wind speed. The results demonstrate that the algorithm can accurately predict wind speed better than each method individually.

A last study from *J.Shi* [34] compares two hybrid models, namely, ARIMA-ANN and ARIMA-SVM with the single forecasting models ARIMA, ANN, and SVM. The main remarks of this study showed that hybrid approaches are viable options for wind speed and wind power time series forecasting, but they do not always produce superior performance for all time horizons. In this case, for wind speed, the hybrid methods present the best performance, while ARIMA method cannot outperform ANN and SVM methods. For wind power generation, the hybrid methodology outperforms single models only for 1-step ahead forecasting, while ANN method has the best performance for 3-step and 7-step ahead forecasting. Hence, hybrid models cannot universally outperform the single forecasting models.

2.2.6 New Models

Some novel wind forecasting models have been developed in recent years. Between the most interesting ones, Extreme Gradient Boosting (XGBOOST), Adaptative Neural Fuzzy Inference System (ANFIS), Random Forest (RF) and Support Vector Machine (SVM) models have achieved the most accurate predictions for wind speed and wind power generation.

A briefly description of these models is presented below:

- XGBOOST is an efficient system implementation of Gradient Boosting, applied to supervised learning problems using training data to predict a target variable. It uses decision trees as its base learner and by adding new base learners, the error between predictive values and target is reduced. The final predictive values are equal to the summation of all base learners [35].
- ANFIS is a system that combines fuzzy logic technique with ANN techniques, which brings the learning capabilities of the ANN to fuzzy inference systems. In an ANFIS, the neuro-adaptive learning methods are used to adjust the parameters of the membership function. The structure of the neuro-fuzzy model for wind power forecasting can be presented as a special multilayer FFNN [36].
- RF is an ensemble method that combines the prediction of several decision trees. The basic principle is called bagging (bootstrap aggregation), where a sample of size n taken from the training set S_n is selected randomly and fitted to a regression tree. This sample is called bootstrap, and it is chosen by replacement, which means that the same observation may appear several times. A bootstrap sample is obtained by selecting randomly n observations with replacement from S_n, where each observation has a probability of $1/n$ to be selected. The aggregation is performed by averaging the outputs of all trees, which makes the final prediction more reliable [37].
- SVM is a powerful and robust methodology in statistical ML, that has been successfully applied to regression problems, including problems of wind speed and wind power prediction. The foundation of SVM is that it formulates the statistical learning problem as a quadratic programming model with linear constraints. It is closely related to ANN and used effectively for nonlinear classification problems [38].

Some of the papers found in the literature that employ these new models for wind speed and wind power forecasting are the following:

A study from *Q.Phan et al* [35] focuses on a deterministic wind power generation forecast using a XGBOOST model for short-term time horizon, specifically one hour ahead. The data used in this study corresponds to historical wind power data recorded at Taiwan's wind farms, and the NWP wind speed forecasts obtained from Taiwan's central weather bureau. The performance of the model, in terms of training speed and accuracy, is compared with a traditional ANN, a LSTM neural network and a Temporal Convolutional Network (TCN).

The results presented in [35] demonstrate the superiority of the proposed XGBOOST model, since it achieved the highest accuracy among all models. The author of the paper also mentions that the parameters of the XGBOOST model need to be modified through a tuning process, otherwise, XGBOOST could not provide a good forecasting for wind power generation.

H.Zheng et al [39] proposes a model for short-term wind power generation forecast based on XGBOOST, with weather similarity analysis and feature engineering. Hourly wind power generation is predicted for the week between April 21st and 28th of 2017, using the data from January 1st of 2016 to April 20th of 2017 as training. The results of the proposed model are compared with a BPNN, RF, Classification and Regression Tree (CART), SVM and a single XGBOOST model. Among all the methods, XGBOOST produced the highest accuracy of prediction, while weather similarity analysis and feature engineering significantly improved the accuracy of the forecast results when comparing with the single XGBOOST model.

A study from *F.Castellanos et al* [40] explores an ANFIS approach to forecast average hourly wind speed. To determine the characteristics of ANFIS that best suited the target wind speed forecasting system, several ANFIS models were trained, tested and compared by changing different parameters. Regarding the results, they proved to be in accordance with the actual data, since after a trial and error process, four ANFIS models gave predictions with errors in the range of 25.5% to 32.5%.

Y.Kassa et al [41] presents an ANFIS based approach for one day ahead hourly wind power generation forecast. The proposed model is trained with historical wind speed and wind power data of a 2.5 MW rated wind turbine installed in Beijing, using one year of data. The performance of the ANFIS model is therefore evaluated against persistence, a BPNN and a hybrid method. The results demonstrated that ANFIS outperformed all other methods tested, achieving an average MAPE of 6.88%, highlighting the accuracy and reliability of ANFIS approach.

Another paper from *L.Fugon et al* [42] evaluates three different models for short-term WPGF. The models analyzed are ANN, RF and SVM, while three wind farms in France located in different terrain complexity and climatic conditions are considered in the analysis. The data used corresponds to a time series of hourly power production for a 18 months period, specifically from July 2004 to December 2005. For the same period, NWP of Meteo France is used, considering two meteorological variables, wind speed and gust wind direction. The forecast is made once a day for time horizons from 0 to 60 hours ahead, with 3-hour resolution. The results obtained in [42] revealed that RF outperformed the rest of the models.

A last study from *M.Moantes et al* [43] presents a SVM model to forecast wind speed. Twelve years of wind speed data available for Madina city, Saudi Arabia are used by dividing it in three parts: training data that is used to build up the model, validation data that is used to select the parameters of the system that best perform on the data, and testing data that is used to make the predictions of the model and to measure its performance. Moreover, the SVM model is also compared with a MLP neural network and the results showed that SVM obtained a lower Mean Square Error (MSE) than MLP, when comparing actual and predicted data. In fact, SVM outperformed MLP for all systems with orders ranging between 1 and 11, where the order of the system determines the number of previous wind speed days used as inputs to forecast the wind speed of the next day.

Finally, after reviewing all the papers mentioned in this Chapter 2, it is possible to conclude that wind speed and wind power generation forecasting is an extend task that depends on different factors, such as the time horizon of the forecast, the resolution and quantity of data used for training and testing or the meteorological variables considered. Therefore, there is not a clear method that outperforms all others for wind power generation forecasting and that is the primary reason why this thesis develops and compares different methods. The main focus is to find the model that best fits the characteristics of the wind farms analyzed in this case. Considering that the sample of 20 wind farms studied represent 10% of the total number of wind farms connected to the MV distribution network of Portugal, the results might be significant for the DSO.

3

Methodology

Contents

As mentioned before, all the implementation is done in Python and the data used correspond to 20 wind farms of Portugal mainland. The initial data provided by the DSO is organized in 3 different folders: Measurements, Meteorology and Predictions.

- Measurements, contains the power generation (measured every 15 minutes in the secondary substations) of each wind farm for the period 2015-2021.

- Meteorology, contains the weather data for each wind farm. In this case there are two different files: IPMA that contains 3 hour records of temperature, radiation, wind speed and wind direction for the period 2020-2021 and ISTMeteo that contains 15 minute records of temperature, radiation, rain, accumulated rain, wind speed and wind direction for the period 2021-06 to 2021-12.

- Predictions, contains DSO power forecasts for the period 2020-2021. Those values are used as benchmark to compare with the models results.

The methodology proposed in this thesis corresponds to the five stages presented in Figure 3.1 and described in this chapter.

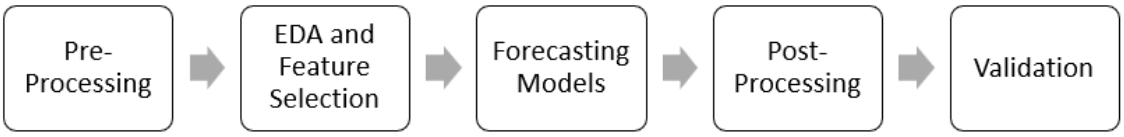


Figure 3.1: Methodology stages

Pre-Processing, is where the initial data is treated and cleaned by removing outliers and by dealing with missing values. Then, *EDA and feature selection* gives a statistical understanding of the data and defines the features to use in the models. In *Forecasting Models* the final datasets are divided into training and test sets and all the forecasting methods are implemented to make the power predictions. Then, in the *Post-Processing* the forecasting results are saved after being checked and adjusted if necessary and the plots are generated. Finally, *Validation* calculates and reports the error metric used to compare the performance of the models.

Based on the error metric and after comparing the results of the different models, the best model is chosen for further tests and improvements and the results are discussed. Important to mention that presenting all the data and plots of the 20 wind farms would be extensive, therefore just some wind farms are used as reference. Although, the algorithms implementation, tests, results or error calculations are done for all of them.

3.1 Pre-Processing

This stage intends to prepare the raw data and make it suitable for the forecasting models. The first step is to create a complete dataset joining the power measurements with the meteorological information for each wind farm. For IPMA data an upsampling (increasing the frequency of the samples) is required to transform the 3 hour data into a 15 minutes resolution. To assign the values of the new points created, a linear interpolation is done between the known data. For ISTMeteo data resample is not necessary because the time resolution is already 15 minutes.

The next step is to identify and handle outliers, inconsistent data points and missing data. One example is shown in Figure 3.2, where the power generation of wind farm 3 is presented. There are some values (out of range) highlighted with a red ellipse. After verifying the installed capacity of wind farm 3, that corresponds to 25.8 MW, it is clear that it is impossible to generate power above 25.8 MW, therefore those values higher than the installed capacity correspond to outliers and are removed.

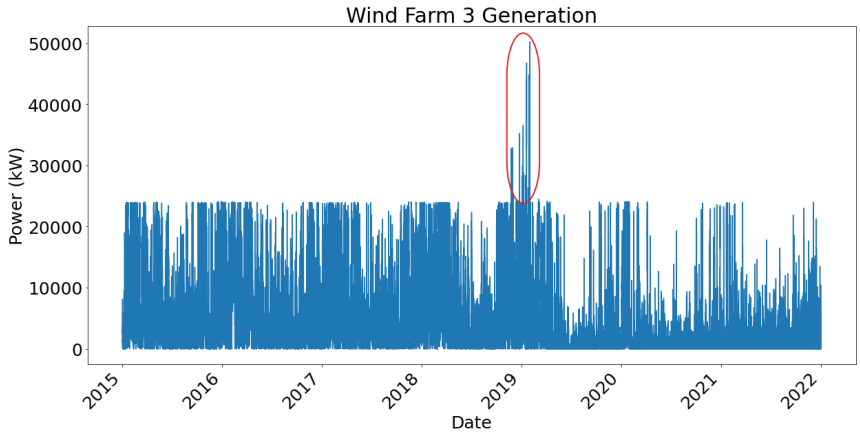


Figure 3.2: Power generation data of wind farm 3

In the same way, all data points in the datasets that present a value of power higher than the installed capacity of the wind farm to which they belonged, similar to the example shown in Figure 3.2, are considered outliers and are removed from the datasets. Negative values of power, if they exist, are considered inconsistent data points and are adjusted to zero.

Another aspect that is addressed in the data pre-processing is the missing data. An example where missing data can be observed is presented in Figure 3.3. In this case and for unknown reasons, there is no information of power generation of wind farm 1 for some months of 2021.

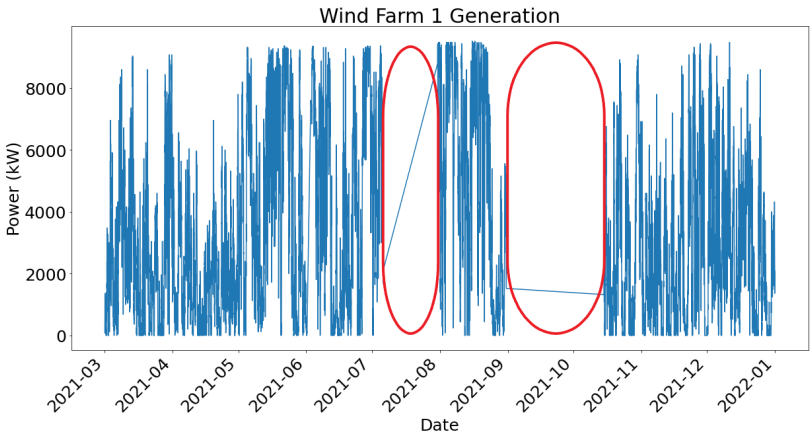


Figure 3.3: Power generation data of wind farm 1

But not only the quality of the power data is important, also looking and cleaning the weather data is necessary in order to create a robust model. Figure 3.4 shows plots of wind speed, wind direction and power generation of wind farm 9 based on IPMA data, where there are also some missing values of wind speed and wind direction around January of 2021.

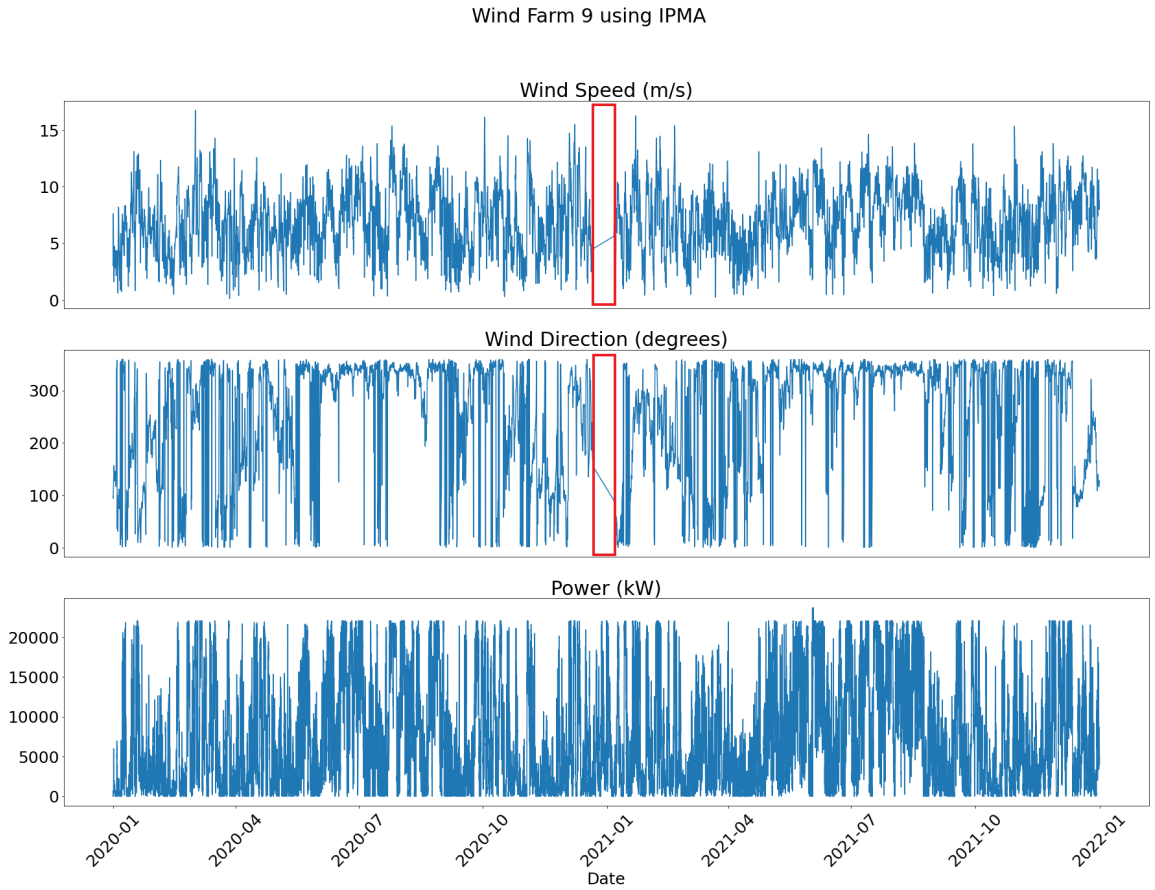


Figure 3.4: IPMA wind speed, wind direction and power generation data of wind farm 9

To deal with missing data, as it was shown in the previous examples, several strategies are applied to fill in the gaps. All the strategies are specifically based on two factors: the position (where the data is missing) and the quantity (number of consecutive values that are missing).

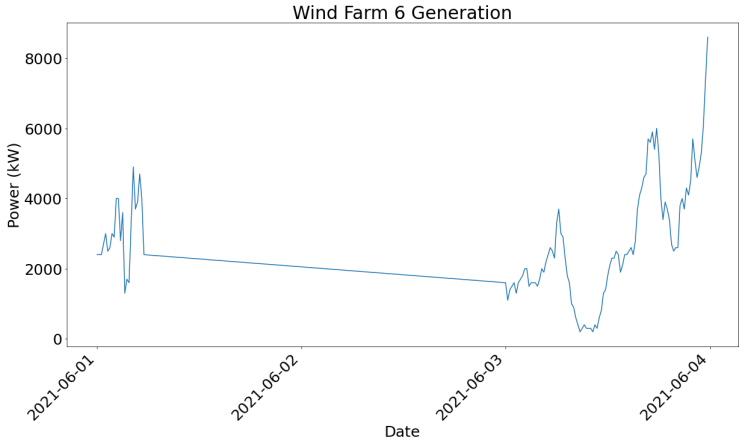
In case the missing data is located at the beginning of the dataset, instead of trying to fill the missing values, the algorithm decreases the length of the training set to the first value that is available but respecting the minimum quantity of data defined. If in the training set 50% or more of the values are missing, then no forecast is done and the training set becomes invalid.

On the other side, if the missing data is located at the end of the dataset, a calculation based on the median is used to fill in the missing values.

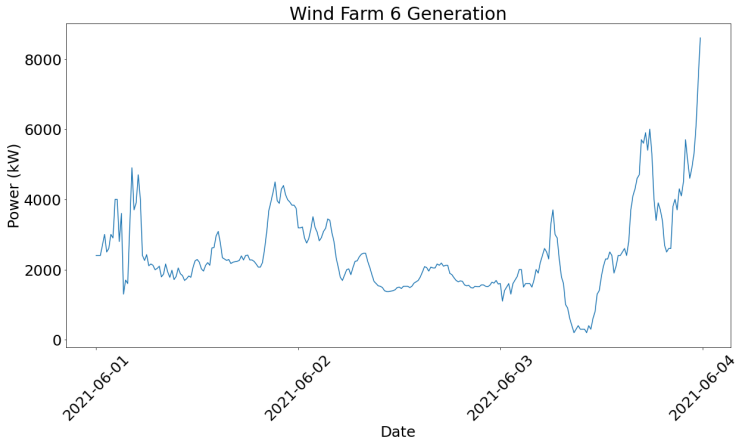
When the missing data is not located on the extremes but it is in the middle of the dataset (having available values before and after the gap), four different scenarios are considered:

- If the missing data correspond to one hour (4 data points) or less, the interpolation approach is used. Since only a small number of values are missing, a straight line between both sides gives a good approximation of the missing values.
- From one hour (4 data points) to one day (96 data points) of missing data, an approach based in adjusting the profile of the previous day is used. It considers the time where the missing data is found and also the previous day information for that specific moment, to make a normalization and adapt it to the current day.
- If the missing data goes from one day (96 data points) to one week of 5 days (480 data points), the median approach is used, but in this case the day of the week and the exact time where the data is missing are also considered. It is relevant to mention at this point that only real values contribute to the median, values created by the missing data algorithm are not taken into account in the median calculation.
- For more than one week (more than 480 data points) of missing values, the gap is not filled because creating artificial values for long periods of time may have a negative effect in the forecasting models and consequently in the results. The approach in this case, is to remove the dates that contain the large periods of missing data from the training set, as long as the minimum length defined for the training set is respected.

To show an example of how the fill missing data algorithm works, Figure 3.5 shows the power generation of wind farm 6 for three days of June of 2021. On Figure 3.5(a) a gap of missing data for almost two days can be observed, while on Figure 3.5(b) the final result after using the fill missing data algorithm is presented.



(a) Missing data



(b) Data filled

Figure 3.5: Before and after using the fill missing data algorithm example

In this case, the median plus the day of the week and the exact time approach mentioned before was used to create the missing values since the length of the gap is comprised between the range of one day and one week, as shown in Figure 3.5(a).

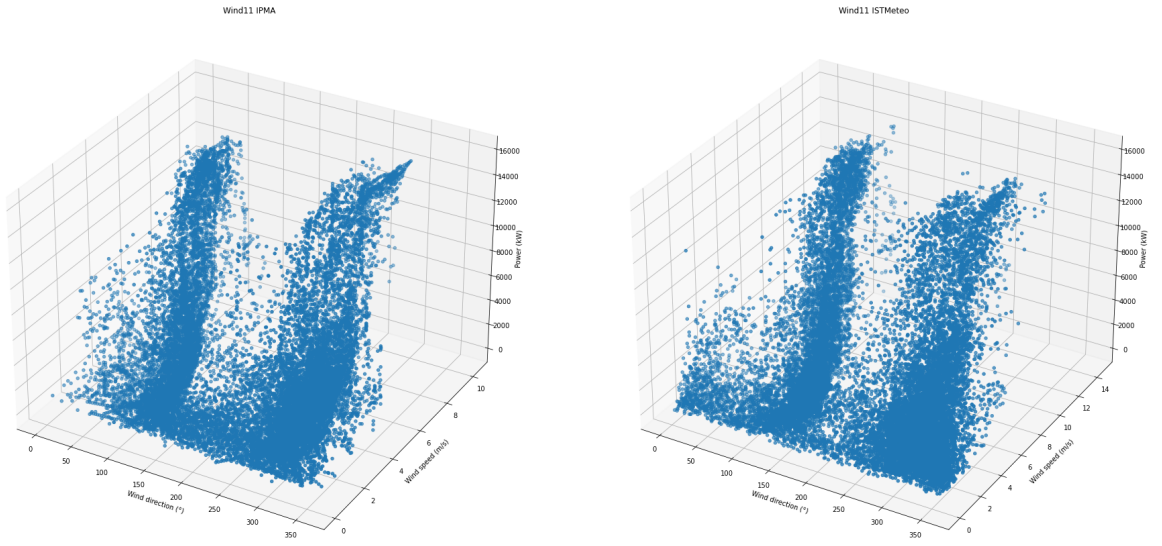
Finally, once all the data has passed through the outliers removal and the filling missing data algorithm, the clean datasets obtained after pre-processing can pass to the next stage.

The final datasets obtained for IPMA contain the information of power, temperature, radiation, wind speed and wind direction for the period 01-01-2020 to 12-31-2021, while for ISTMeteo the final datasets contain the information of power, temperature, radiation, rain, rain accumulated, wind speed and wind direction for the period 01-06-2021 to 12-31-2021.

3.2 EDA and Feature Selection

Exploratory Data Analysis (EDA) is the process where the user look at and understand the data with statistical and visualization methods. This step helps identifying patterns and problems in the dataset, as well as deciding which model or algorithm should be used in subsequent steps.

First of all, it is necessary to decide whether to use IPMA datasets or ISTMeteo datasets in the forecasting models. Figure 3.6 presents a comparison between IPMA and ISTMeteo data through the 3D plot (wind speed, wind direction and power) of wind farm 11.



(a) IPMA data **(b)** ISTMeteo data

Figure 3.6: Wind speed, wind direction and power 3D plot of wind farm 11

Both Figure 3.6(a) and Figure 3.6(b) are similar and visually it is not possible to perceive which option offers the best data quality. Hence, IPMA and ISTMeteo will be compared based on their performance to forecast 1 month of 2021 (DEC), by using 6 months of 2021 for training (JUN-NOV) and the option that offers the lower percentage of error will be selected. This time-frame of comparison is precisely chosen due to the fact that ISTMeteo available data includes just 7 months.

Now to have an idea about the data contained in the datasets, Table 3.1 and Table 3.2 present some descriptive statistics of wind farm 15 for IPMA and ISTMeteo datasets, respectively. The variables T and R stand for temperature and radiation.

Table 3.1: Descriptive statistics of wind farm 15 - IPMA dataset

	Power (kW)	T (K)	R (W/m^2)	Wind Speed (m/s)	Wind Direction ($^\circ$)
Count	70,176	70,153	70,153	70,153	70,153
Mean	7,834.07	288.58	735.84	7.01	241.51
Standard Dev	7,177.94	4.95	729.82	2.80	109.90
Min	0.00	272.94	0.00	0.14	0.02
25th Percentile	1,600.00	285.25	86.43	4.93	157.24
50th Percentile	5,610.00	288.11	524.35	6.92	284.33
75th Percentile	13,102.50	291.45	1,193.59	9.08	338.96
Max	29,705.29	310.58	2,814.88	16.32	359.98

Table 3.2: Descriptive statistics of wind farm 15 - ISTMeteo dataset

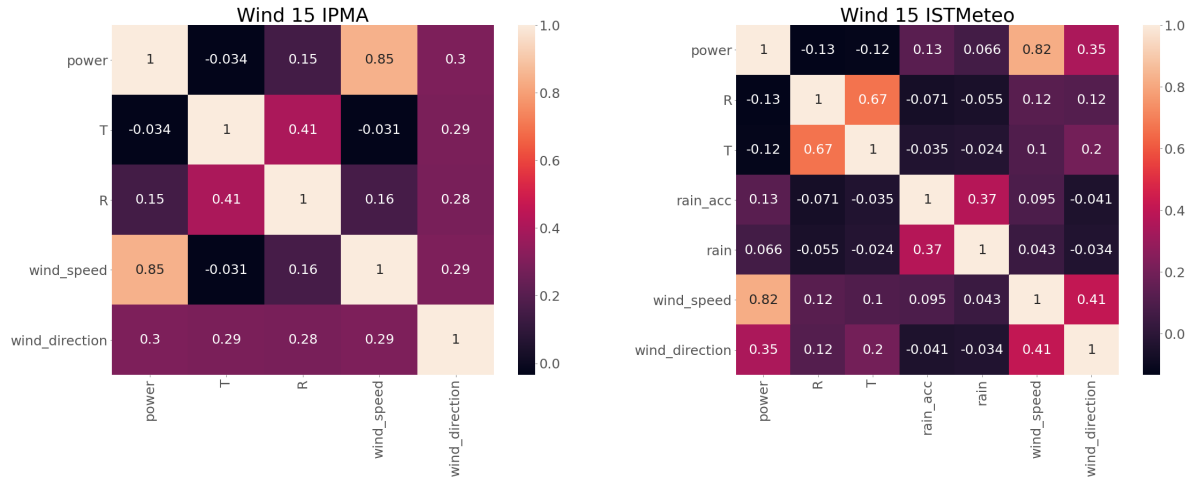
	Power (kW)	T (K)	R (W/m^2)	Rain (mm/h)	Rain Acc (mm/h)	Wind Speed (m/s)	Wind Direction ($^\circ$)
Count	20,544	20,544	20,544	20,544	20,544	20,544	20,544
Mean	8,860.76	289.46	215.95	0.01	0.23	5.73	254.72
Standard Dev.	7,265.69	4.55	313.06	0.10	1.51	2.36	113.75
Min	0.00	277.07	0.00	0.00	0.00	0.07	0.01
25th Percentile	2,370.00	286.50	0.00	0.00	0.00	3.84	170.92
50th Percentile	7,190.00	289.25	0.00	0.00	0.00	5.60	315.11
75th Percentile	14,832.50	292.27	416.24	0.00	0.00	7.53	339.59
Max	29,705.29	305.58	1,042.16	7.47	24.85	12.75	360.00

From Table 3.1 and Table 3.2 some differences between IPMA and ISTMeteo data already appear. For example, when comparing the radiation, the average value for IPMA is 735.84 W/m^2 while for IST-Meteo it is much lower, 215.95 W/m^2. Also for wind speed, the meteorological parameter that directly affects the wind energy generation, IPMA has an average value of 7.01 m/s while for ISTMeteo the average is 5.73 m/s.

After the overlook of the available data, the next step is to determine which features (input variables) are used in the forecasting models. Only a few variables in the dataset are useful for building the models and the rest of the features are either redundant or irrelevant. If we input the dataset with all these redundant or irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the models [44].

To select the appropriate features, a correlation matrix, which provides the relationship between variables is used. The correlation coefficients can fall between -1 and +1, where a high and positive correlation indicates that the variables measure the same characteristic. Thus, the features with the higher correlation with the target variable (power) are chosen and the features with negative or low correlation are discarded.

Figure 3.7 shows the correlation matrix of wind farm 15 for both IPMA data in Figure 3.7(a) and ISTMeteo data in Figure 3.7(b). The colors represent the correlation for each combination of features, in a scale from -1 to 1 but for clarity the exact value of the correlation coefficient is also shown.



(a) IPMA data

(b) ISTMeteo data

Figure 3.7: Correlation matrix of wind farm 15

Based on the correlation matrix, only wind speed and wind direction features are selected to forecast wind power generation. Wind speed presents the higher correlation as expected, followed by wind direction which, despite not having a very high correlation may be relevant. The rest of the variables are discarded because they present either negative or very low correlation. After feature selection, the dataset is prepared for the next stage.

3.3 Forecasting Models

At this point, where the data has been cleaned, the missing values were filled or handled, the outliers were removed and the feature selection has been done, the data is finally in place for the regression, ML and AI based models implementation.

In the case of time series, the forecast can be made for different time horizons (very-short, short, medium or long term forecasting) as explained in Section 2.1. Once the forecast period is defined (can be any amount of time: minutes, hours, days, months or even a year), the final dataset is divided into the following two subsets:

- Training set, data used by the model to discover and learn patterns between the features and the forecast variable, power.
- Test set, data on which the power predictions are generated. Correspond to unseen data used to evaluate the performance of the model.

To graphically observe how the data is divided into training and test, Figure 3.8 shows an example of 6 months training and 1 month forecast using wind farm 15 IPMA dataset. The training period appears in red and the test period (where the model predicts the power) in blue.

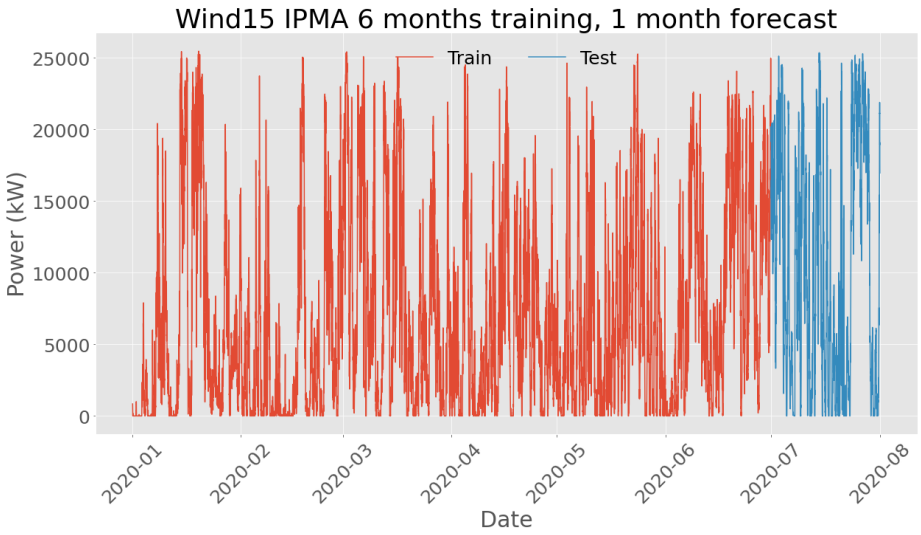


Figure 3.8: Training and test sets example

The training set is normally larger than the test set because the idea is to feed the model with as much data as possible, to learn meaningful patterns and then apply the things learned to create predictions on unseen data.

As mentioned before, eight different forecasting models are implemented to predict the power generation of 20 wind farms of Portugal; starting from persistence (to have a benchmark), passing trough regressive models, a neural network and some newer models. Specifically, the following forecasting models are tested:

- **Persistence**

  As mention in Section 2.2.1, this method assumes that the forecast value corresponds to the same value of the previous time step. In our case, persistence forecast corresponds to the power measured at the same time instant from the previous day (96 time intervals before the desired forecast time instant), considering a data resolution of 15 minutes. It can be formulated as:

$$\hat{X}(t) = X(t - 96) \tag{3.1}$$

  Where $\hat{X}(t)$ is the wind power forecast value at certain instant of time and $X(t - 96)$ is the wind power value measured 96 time intervals before.

- **Auto-Regressive (AR)**

  This model uses observations from previous time steps as an input for the regression equation, to predict the value at the next time step. In simple terms, an AR(p) model relates p past observations to the current value X_t as [45]:

$$X_t = \mu + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \tag{3.2}$$

  Where μ is the mean value, φ_i is a coefficient which reflects each past observation X_{t-i} influence on current value and ε_t is the actual stochastic perturbation.

- **Auto-Regressive with Exogenous Variable (ARX)**

  An ARX model is an auto-regressive model with exogenous inputs. It assumes a stationary and invertible process where the exogenous inputs come from an external system. Therefore, an ARX(p, n_x) model can described as [46]:

$$X_t = \mu + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{n_x} \eta_i b_{t-i} + \varepsilon_t \tag{3.3}$$

  Where η_i is the exogenous coefficient and n_x is the order of the exogenous inputs.

- **Long Short-Term Memory (LSTM) Neural Network**

  LSTM is one of many types of RNN. Since RNN cannot store long time memory, LSTM proved to be very useful in forecasting cases with long time data based on 'memory line'. In a LSTM the memorization of earlier stages is performed trough gates [47].

  Every LSTM node consists of a set of cells responsible of storing passed data streams. The upper line in each cell links the models as transport line handing over data from the past to the present ones and the independency of cells helps the model to filter aggregate values from a cell to another.

  At the end, the sigmoidal neural network layer composing the gates, drive the cell to an optimal value by disposing or letting data pass through. Each sigmoid layer has a binary value (0 or 1), with 0 meaning to let nothing pass through and 1 meaning to let everything pass through [47]. Figure 3.9 shows the composition of LSTM nodes.

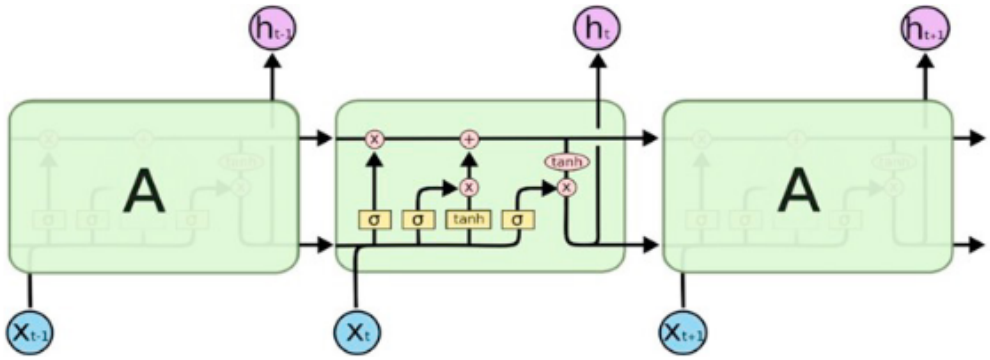


Figure 3.9: LSTM neural network structure [47]

  To develop the LSTM model in Python, the library tf.keras.layers.LSTM[1] was used.

- **Decision Trees (DT)**

  DT are a common way of representing the decision-making process through a branching, tree-like structure. They are made up of different nodes. The root node is the start of the decision tree, which is usually the whole dataset within ML. Leaf nodes are the endpoint of a branch, or the final output of a series of decisions. The features of the data are internal nodes and the outcome is the leaf node [48]. Figure 3.10 presents the basic structure of a decision tree.

  To develop the DT model in Python, the library sklearn.tree.DecisionTreeRegressor[2] was used.

[1] https://keras.io/api/layers/recurrent_layers/lstm/
[2] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

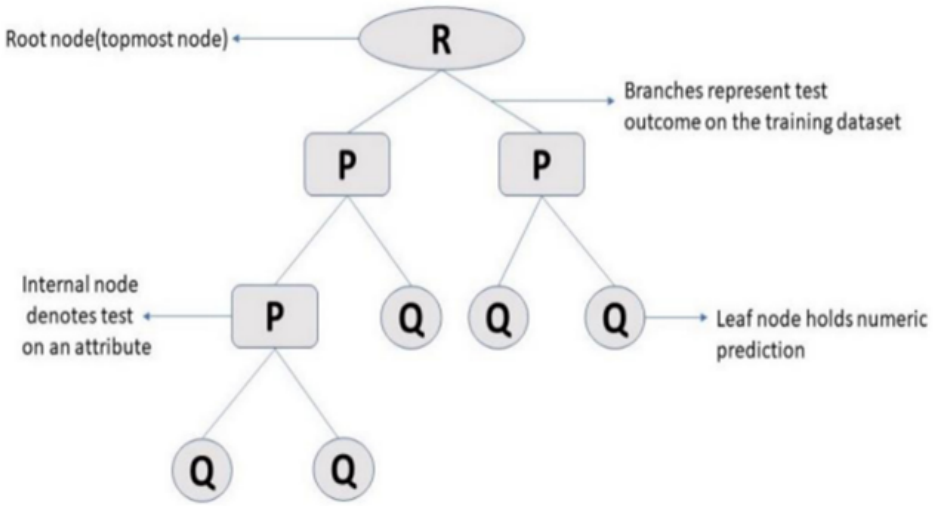


Figure 3.10: DT structure [49]

- **Random Forest (RF)**

  RF is a method that combines several decision trees and uses the majority voting of the individual trees to find the overall class. It is an ensemble learner for classification and regression that considers three steps: randomly selecting training data when making trees, choosing some subsets of features when splitting nodes and employing only a subset of all features for splitting each node in each simple decision tree. During the training of the data, each tree learns from a random sample of the data points [50]. Figure 3.11 shows the composition of 'n' number of trees, which constructs the RF algorithm.

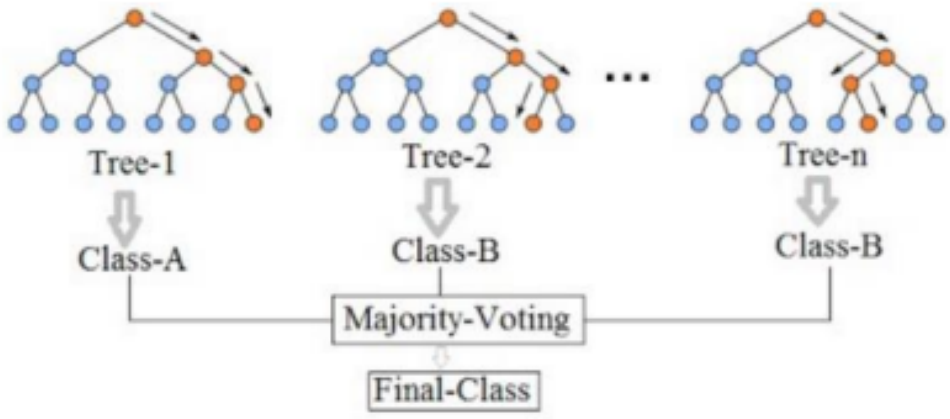


Figure 3.11: Composition of RF [1]

To develop the RF model in Python, the library sklearn.ensemble.RandomForestRegressor[3] was used.

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

- **Extreme Gradient Boosting (XGBOOST)**

  XGBOOST is one of the most efficient implementation of gradient boosted decision trees, specifically designed to optimize memory usage and exploit the hardware computing power. The main idea of boosting is to sequentially build sub-trees from an original tree such that each subsequent tree reduces the errors of the previous one. In such a way, the new sub-trees will update the previous residuals in order to reduce the error of the cost function [51].

  The process of additive learning in XGBOOST as explained by *N.Dhieb et al* [51] is presented below. First, consider a data set D expressed as follows:

$$D = \{(x_i, y_i),\ where\ x_i \in \mathbb{R}^m\ and\ y_i \in \mathbb{R}\} \tag{3.4}$$

$$|D| = n \tag{3.5}$$

  Where m is the dimension of the features x_i. y_i is the response of the sample i and n is the number of samples. The vertical bars in Equation 3.5 denotes the cardinality of the set.

  Then, the predicted value of the entry i and denoted as $\hat{y_i}$, is defined as:

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i),\ where\ f_k \in F \tag{3.6}$$

  Where f_k indicates an independent tree in the space of regression trees F and $f_k(x_i)$ refers to the predicted score given by the i-th sample and k-th tree. The objective function of the XGBOOST, denoted by ζ, is given as follows:

$$\zeta = \sum_{i=1}^{n} \ell(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{3.7}$$

  By minimizing the objective function ζ, the regression tree model functions f_k can be learned. The training loss function $\ell(y_i, \hat{y_i})$ evaluates the difference between the prediction $\hat{y_i}$ and the actual value y_i. Herein, the term Ω is used to avoid the overfitting problem by penalizing the model complexity as follows:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \left\| w \right\|^2 \tag{3.8}$$

  Where γ and λ are regularization parameters, T and w are respectively the numbers of leaves and the scores on each leaf.

A second degree Taylor series can be used to approximate the objective function. Let's define $I_j = \{i|q(x_i) = j\}$ an instance set of leaf j with $q(x)$ a fixed structure. The optimal weights w_j^* of leaf j and the corresponding optimal value can be obtained by the following equations:

$$w_j^* = -\frac{g_j}{h_j + \lambda} \tag{3.9}$$

$$\zeta^* = \frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i \in I_j} g_i)^2}{(\sum_{i \in I_j} h_i + \lambda)} + \lambda T \tag{3.10}$$

Where g_i and h_i are the first and the second gradient orders of the loss function ζ. The loss function ζ can be used as a quality score of the tree structure q. The smaller the score is, the better the model is.

As it is not possible to enumerate all the tree structures, a greedy algorithm can solve the problem by starting from a single leaf and iteratively add branches to the tree. Let's say that I_R and I_L are the instance sets of right and left nodes after split. Assuming $I = I_R \cup I_L$, the loss reduction after the split is given as:

$$\zeta_{split} = \frac{1}{2}\left[\frac{(\sum_{i \in I_L} g_i)^2}{\sum_{i \in I_L} h_i + \lambda} + \frac{(\sum_{i \in I_R} g_i)^2}{\sum_{i \in I_R} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \tag{3.11}$$

This formula is usually used in practice for evaluating the split candidates. The XGBOOST model use many simple trees and score leaf nodes during splitting. The first three terms of the equation represent respectively the score of the left, right and original leaf. In addition, the term γ is the regularization on the additional leaf and it will be used in the training process.

To develop the XGBOOST model in Python, the library xgboost.XGBRegressor[4] was used.

- **Support Vector Machine (SVM)**

  SVM is used for both regression and classification but commonly finds its application in classification purposes. SVM regression trains the model using a symmetrical loss function, which penalizes for both high and low misestimates. The aim is to find a hyperplane that differentiates the data points plotted in multi-dimensional space, where each dimension represents the different features used.

[4] https://xgboost.readthedocs.io/en/stable/python/python_api.html

The hyperplane having maximum separation distance is used to meet the request of the regression with a higher degree of accuracy. Different coordinates on the plot are obtained by mapping of the parameters under observation on the plot. It can be described with the help of mapping function formulated as [49]:

$$f(x) = \sum_{i=1}^{n} \omega\phi(X_i) + b \tag{3.12}$$

Where ω is is the weighted vector and (X_i) is the mapped regressor.

To develop the SVM model in Python, the library sklearn.svm.SVR[5] was used.

Those are the eight wind power generation forecasting methods developed and implemented in the present thesis. All models excepts Persistence and AR use wind speed and wind direction as features, as determined in the previous stage.

Once each model has been trained (using the training set) and before calculating the predictions (over the test set), a technique called k-fold Cross Validation (CV)[6] is applied to validate the effectiveness of the models. What this technique does is to divide the training set into k smaller sets (where a value of $k = 10$ is very common in the field of applied ML and then for each one of the k splits the following procedure is performed:

1. A model is fitted using $k - 1$ of the folds as training data.
2. The resulting model is validated on the remaining fold (i.e., it is used as a test set to compute a performance measure).
3. The score of the resulting model is recorded.
4. Steps 1 and 2 are repeated until every k-fold has served as test set.

Figure 3.12 shows how the procedure works. At the end the performance metric reported by k-fold cross validation is the average of the values computed in the loop, it is called average score and the closer it is to 1, the better.

[5] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html
[6] https://scikit-learn.org/stable/modules/cross_validation.html

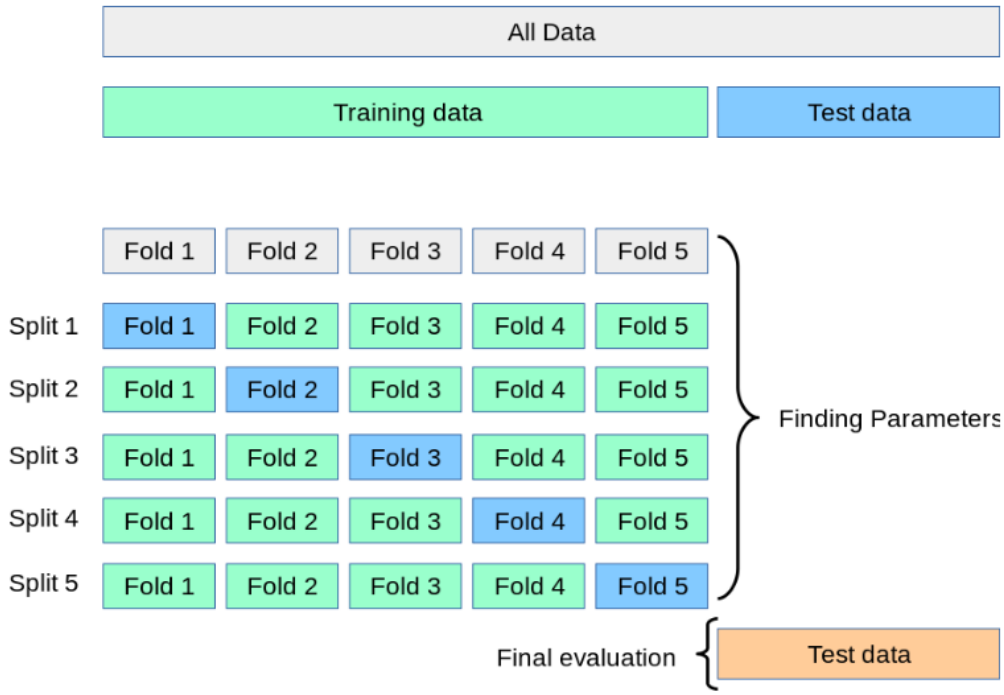


Figure 3.12: K-fold cross validation procedure [52]

After k-fold cross validation, the final model is applied on the test set and the power predictions are generated.

3.4 Post-Processing

This stage starts with the forecast results obtained for the test set in the previous stage. Its main purpose is to check the generated power predictions and to adjust the values out of range, if they exist. To do that, the algorithm checks two conditions:

- $Power\ predictions >= 0$. The predicted power cannot be negative. In case there are negative values, they are adjusted to 0.
- $Power\ predictions <= Installed\ capacity$. The predicted power cannot be higher than the installed capacity of the wind farm. In this case the maximum forecast value is limited to the installed capacity.

Table 3.3 presents the installed capacity of each wind farm and the year when they were connected to the MV distribution network of Portugal.

Table 3.3: Wind farms installed capacity

Wind Farm	Installed Capacity (kW)	Year of Connection
1	11,291	2003
2	20,850	2005
3	25,800	2008
4	2,150	2007
5	15,050	2008
6	43,000	2005
7	2,800	2004
8	650	2003
9	23,650	2009
10	6,450	2005
11	17,210	1997
12	14,000	2001
13	22,257	2004
14	22,687	2010
15	29,900	2008
16	6,500	2001
17	19,200	2006
18	2,800	2006
19	1,935	2006
20	45,150	2008

Once both conditions are verified or adjusted if necessary, the final wind power predictions are saved and a plot comparing the forecast values with the real values is generated. An example of this plot is presented in Figure 3.13, where XGBOOST method is used to forecast 1 month of 2021 (JUL) by using 6 months of training (JAN-JUN of 2021) for IPMA dataset.

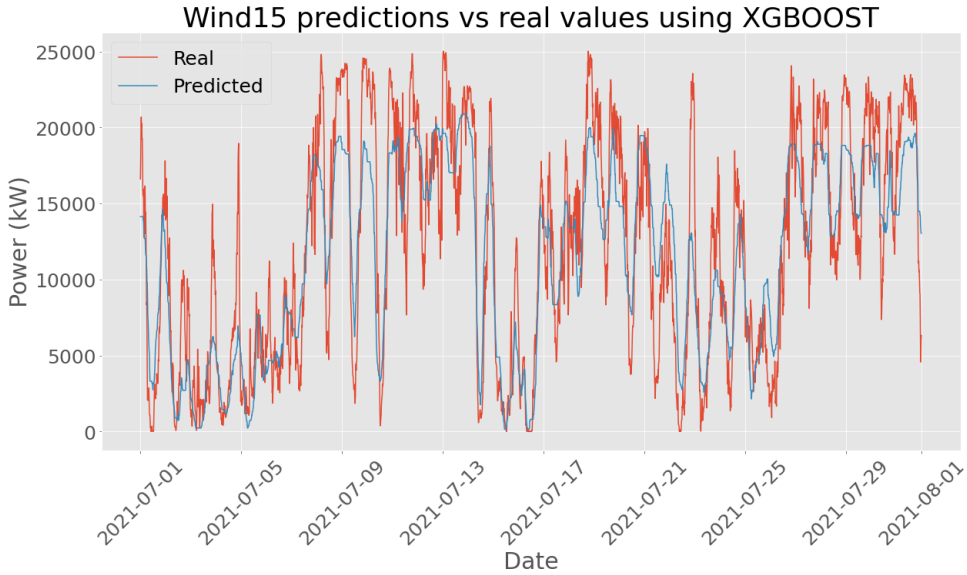


Figure 3.13: Forecast vs real values plot for 6 months training, 1 month forecast using IPMA

3.5 Validation

The accuracy is the most important factor when comparing different forecasting methods. To assess the performance of the models, several statistical metrics, that show the deviations of forecast values from measured values, are employed [28].

The most common error criteria used to evaluate and compare different forecasting methods are: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Root Mean Square Error (RMSE) and Standard Deviation Error (SDE).

The MAE is the most natural criterion, since it is simply the average of all found errors, through their absolute values to avoid error offset. The RMSE does the same, with the difference that avoids the error sign by squaring instead of using the absolute value. The MAPE divides each error by the real measured value before averaging, in order to get a percentage. However, this has its drawbacks, especially when the real value tends towards zero. [37]. The mathematical expressions used to compute each one of the mentioned errors are presented in Table 3.4.

Table 3.4: Commonly used error metrics

Error	Formula
MAE	$\frac{1}{N}\sum_{i=1}^{N}\left\|\hat{P}_i - P_i\right\|$
MAPE	$\frac{1}{N}\sum_{i=1}^{N}\frac{\left\|\hat{P}_i - P_i\right\|}{P_i} \times 100$
MSE	$\frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2$
RMSE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i)^2}$
SDE	$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{P}_i - P_i - P_{avg})^2}$

Where N is the total number of samples, $\hat{P}_i$ is the forecast value, P_i is the measured value and P_{avg} is defined as:

$$P_{avg} = \frac{1}{N} \sum_{i=1}^{N} \hat{P}_i - P_i \tag{3.13}$$

Now, after having the definitive wind power predictions and in order to compare them with the real values of power, the last stage consists in calculating the error metric that is used to analyze the performance of the models. From the different criteria mentioned in Table 3.4, the Root Mean Square (RMSE) was chosen, but with a small difference: in this case the error is normalized by dividing by the installed capacity of the wind farm.

Thus, it is called Relative Root Mean Square Error (RRMSE) and it is calculated as:

$$\text{RRMSE } (\%) = \frac{\text{RMSE}}{P_{installed}} \times 100 = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{P}_i - P_i)^2}}{P_{installed}} \times 100 \tag{3.14}$$

Where N is the total number of samples, $\hat{P}_i$ is the forecast value, P_i is the measured value and $P_{installed}$ is the installed capacity of the wind farm.

Basically, the algorithm calculates the daily RRMSE between predictions and real values of power for the test period defined and then the average of this daily error is reported (as a percentage), to have an idea of the accuracy of the forecast made.

This RRMSE metric is used as comparison point in all the calculations, results and improvement tests performed in this work. It will help to determine which datasets to use: IPMA or ISTMeteo, which forecasting model is the best option to use and to tune (the one that gives the lower percentage of error) and it will be used to compare the results achieved with the DSO results.

Again, the main objective of this thesis is to develop and implement a framework based on a robust wind power forecasting model that improves the performance of the forecast model that the DSO is currently using and to reduce the RRMSE achieved as maximum as possible.

4

Results and Discussion

Contents

This section presents the results obtained for the forecasting models developed, the comparison of the RRMSE between them and with the DSO results. It also presents the different tests and the tuning process performed to the best-suited model in order to improve the results.

4.1 Persistence and AR

The first method tested is persistence, that corresponds to the most simplest model and basically gives a point of reference for the rest of the models. Also AR is tested at this moment because both methods, persistence and AR, use only the past measurements of power to make the predictions, no meteorological data is needed and that means that the results are the same independently of which dataset, IPMA or ISTMeteo is used.

Table 4.1 presents the RRMSE of the 20 wind farms obtained for persistence and AR models, for 6 months training (JUN-NOV of 2021) and 1 month forecast (DEC of 2021). The DSO error is also presented for the same forecast period to have a point of comparison.

Table 4.1: RRMSE for Persistence and AR: 6 months training, 1 month forecast

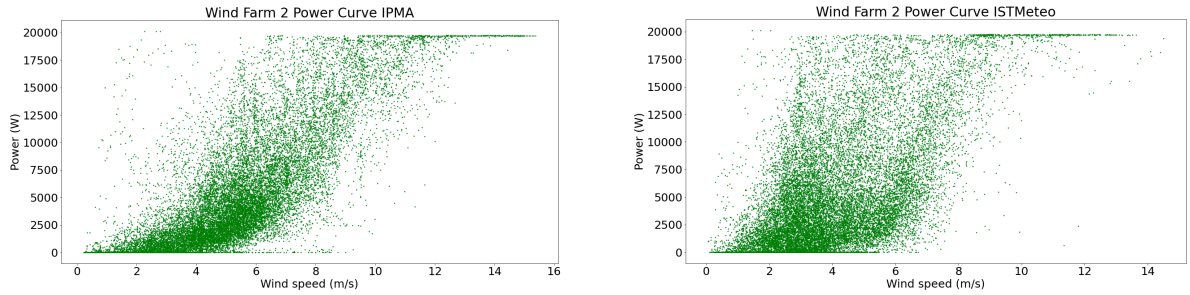
Wind Farm	Persistence (%)	AR (%)	DSO (%)
1	24.594	16.886	13.482
2	37.717	35.105	16.187
3	14.895	12.396	41.013
4	35.714	32.035	19.850
5	30.814	26.713	14.874
6	32.897	27.102	18.929
7	35.403	28.947	50.877
8	38.103	32.230	21.536
9	30.248	26.306	14.372
10	34.136	30.781	45.416
11	31.939	29.759	29.586
12	24.222	19.232	15.593
13	33.787	29.940	17.470
14	38.087	30.688	21.550
15	26.191	20.947	11.954
16	37.940	36.241	21.983
17	34.902	29.912	19.541
18	34.712	33.975	26.619
19	31.060	24.680	18.242
20	29.404	26.565	18.998
Average	31.838	27.522	22.904

The results presented in Table 4.1 show that the average error between persistence predictions and the real values of power are higher than 30%. In the case of the AR model the performance is better than persistence with an average RRMSE of 27.522%, but still when comparing with the DSO error, both methods are far above the 22.904%, that is the mark to beat.

4.2 IPMA vs ISTMeteo Comparison using ARX

Since the other models involve the usage of the two meteorological parameters selected as features, namely wind speed and wind direction, it is necessary to define which datasets IPMA or ISTMeteo will be used to run the models.

As mentioned in Section 3.2 visually or statistically is not possible to determine which datasets, IPMA or ISTMeteo offer better data quality. What is clear is that there are differences present in the data, as shown in Figure 4.1, that compares the power curve of wind farm 2 obtained for IPMA in Figure 4.1(a) and for ISTMeteo in Figure 4.1(b), respectively.



(a) IPMA data

(b) ISTMeteo data

Figure 4.1: Wind farm 2 power curve

Hence, both data are compared based on their RRMSE using an ARX model with wind speed and wind direction as exogenous variables. Both IPMA and ISTMeteo are tested under the same conditions, at the end the datasets with the best performance (lower percentage of error) are selected as the best option and are used to run all the forecasting methods.

Table 4.2 presents the results for ARX model comparing IPMA and ISTMeteo. The training and test periods defined for this comparison are the same that were used for Persistence and AR methods, training set: JUN-NOV of 2021 and test set: DEC of 2021.

Table 4.2: IPMA vs ISTMeteo RRMSE for ARX: 6 months training, 1 month forecast

Wind Farm	IPMA ARX (%)	ISTMeteo ARX (%)	DSO (%)
1	15.384	14.030	13.482
2	19.248	25.155	16.187
3	12.515	12.436	41.013
4	28.131	24.914	19.850
5	20.364	17.288	14.874
6	21.185	20.149	18.929
7	19.673	23.833	50.877
8	27.287	27.379	21.536
9	20.168	20.233	14.372
10	20.690	21.337	45.416
11	23.380	20.676	29.586
12	17.073	15.283	15.593
13	18.394	19.316	17.470
14	25.111	24.070	21.550
15	13.660	14.491	11.954
16	26.028	27.347	21.983
17	23.803	24.219	19.541
18	28.478	29.769	26.619
19	19.225	22.812	18.242
20	21.114	21.995	18.998
Average	21.046	21.337	22.904

From Table 4.2 it is determined that IPMA datasets give a lower average RRMSE than ISTMeteo and therefore is the data chosen to use as input in all the forecasting models.

Regarding the RRMSE for ARX method, it is possible to observe that the average error is already lower than the DSO results, which is positive. However, looking deeply at the individual results for each wind farm the perspective is different. DSO error is lower in almost all wind farms except in wind farm 3, wind farm 7 and wind farm 10, where the high percentage of error on those farms makes the average RRMSE of the DSO (22.904%) worse than the average error for IPMA ARX (21.046%). Based on this, the reality is that the DSO predictions are still better than the predictions obtained for the methods tested until now (Persistence, AR and ARX).

4.3 ML and AI Based Models

AR and ARX models did not outperformed the DSO results, subsequently the rest of the ML and AI based models listed in Section 3.3 are tested. The RRMSE of all of them is compared and the model with the best performance is chosen as the definitive model, if it gives a lower average RRMSE than the DSO.

Table 4.3 presents the RRMSE for Long Short-Term Memory (LSTM), Decision Trees (DT), Random Forest (RF), Extreme Gradient Boosting (XGBOOST) and Support Vector Machine (SVM) models, respectively. The training and test periods defined in all the simulations were the same as before, 6 months training (JUN-NOV of 2021) and 1 month forecast (DEC of 2021). This with the purpose of making the results comparable between them and with the persistence, AR and ARX methods tested previously.

Table 4.3: RRMSE for LSTM, DT, RF, XGBOOST and SVM: 6 months training, 1 month forecast

Wind Farm	LSTM (%)	DT (%)	RF (%)	XGBOOST (%)	SVM (%)	DSO (%)
1	23.209	19.543	12.371	12.451	19.492	13.482
2	24.413	24.596	19.952	17.637	31.561	16.187
3	8.288	15.671	11.257	10.549	10.399	41.013
4	29.888	28.721	28.698	22.649	47.410	19.850
5	22.727	21.783	14.873	13.617	29.755	14.874
6	22.555	25.147	23.205	19.273	35.377	18.929
7	29.426	32.046	21.851	21.101	20.949	50.877
8	25.833	25.484	25.398	24.427	39.240	21.536
9	26.900	23.291	17.296	17.472	24.699	14.372
10	26.700	22.602	21.286	16.374	28.953	45.416
11	21.877	25.785	22.783	21.412	34.463	29.586
12	19.562	23.656	17.673	17.509	23.907	15.593
13	21.859	17.593	18.195	12.947	26.973	17.470
14	26.925	29.963	26.919	28.127	36.026	21.550
15	22.833	17.059	12.828	11.651	15.297	11.954
16	29.310	25.150	22.071	20.628	37.585	21.983
17	27.285	26.611	25.206	21.862	45.642	19.541
18	24.323	31.979	28.080	26.764	29.098	26.619
19	21.575	27.072	18.103	17.219	24.802	18.242
20	27.717	23.865	19.042	18.595	34.802	18.998
Average	24.160	24.381	20.354	18.613	29.822	22.904

From the results obtained in Table 4.3 just two methods, RF (20.354%) and XGBOOST (18.613%) outperformed the DSO results (22.904%). Since XGBOOST has the lower RRMSE, it is chosen as the method to be focus on and to be improved, in order to reduce the percentage of error even more.

At this point, one of the main objectives of the thesis has been achieved, considering that the developed and implemented XGBOOST model can forecast wind power generation with greater accuracy than the DSO system. Now, some tests and improvements to the algorithm are developed and executed with the idea of improving the results as much as possible.

4.4 XGBOOST Adjusting Training and Test Periods

The first test consists on adjusting the training and test periods, to compare the RRMSE of the XGBOOST model under different time horizons. Until now, just a training period of 6 months (JUN-NOV of 2021) and a forecast of 1 month (DEC of 2021) has been tested, so the idea is to make different combinations of training and test sets with the available data.

Since IPMA data was chosen as the datasets to use, there are two years of information that can be divided as training and test periods in several ways. The idea is to start with the same period of time for both training and test sets and then start increasing the training period, while decreasing the forecast horizon; this with the purpose to observe if the model improves the results for longer training periods and shorter test periods. In the same way, long test periods (6 months and 1 year) are tested using the rest of the data available for training, to observe if an improvement in the accuracy can be achieved.

The following eight combinations of training and test periods, using the available 2020 and 2021 data were defined:

- *Combination 1:* 6 months training (JAN-JUN of 2021) and 6 months forecast (JUL-DEC of 2021).
- *Combination 2:* 7 months training (JAN-JUL of 2021) and 5 months forecast (AUG-DEC of 2021).
- *Combination 3:* 8 months training (JAN-AUG of 2021) and 4 months forecast (SEP-DEC of 2021).
- *Combination 4:* 9 months training (JAN-SEP of 2021) and 3 months forecast (OCT-DEC of 2021).
- *Combination 5:* 10 months training (JAN-OCT of 2021) and 2 months forecast (NOV-DEC of 2021).
- *Combination 6:* 11 months training (JAN-NOV of 2021) and 1 month forecast (DEC of 2021).
- *Combination 7:* 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021).
- *Combination 8:* 1 year training (JAN-DEC of 2020) and 1 year forecast (JAN-DEC of 2021).

Table 4.4 to Table 4.7 present the average RRMSE obtained for each combination and the respective DSO error calculated for the same forecast period. To have a fair comparison between the different combinations, regardless of the number of months to forecast, the RRMSE of the same month (DEC of 2021) was analyzed independently of the combination and the same results were obtained.

Table 4.4: RRMSE for XGBOOST Combination 1 and Combination 2

Wind Farm	Combination 1 (%)	DSO (%)	Combination 2 (%)	DSO (%)
1	17.165	11.805	12.365	11.672
2	13.442	13.984	13.484	13.854
3	6.608	21.061	6.831	22.651
4	16.724	14.863	19.819	15.578
5	12.106	10.526	11.838	10.797
6	13.730	11.704	15.590	12.379
7	22.395	42.329	19.029	40.460
8	20.415	14.050	23.409	14.932
9	21.307	11.740	18.295	11.887
10	13.647	27.324	13.728	27.867
11	15.704	19.468	17.318	20.556
12	10.532	9.822	11.641	10.322
13	15.623	15.242	15.589	15.170
14	15.983	15.433	17.259	15.915
15	14.844	10.464	10.729	10.272
16	18.088	16.616	18.478	16.846
17	16.288	14.998	16.966	15.397
18	16.650	16.290	18.842	16.977
19	13.928	13.110	13.682	12.801
20	15.117	14.183	17.164	14.394
Average	15.515	16.251	15.603	16.536

Table 4.5: RRMSE for XGBOOST Combination 3 and Combination 4

Wind Farm	Combination 3 (%)	DSO (%)	Combination 4 (%)	DSO (%)
1	11.738	11.602	11.738	11.602
2	14.058	14.801	14.193	15.148
3	7.478	26.038	8.206	28.588
4	20.532	17.421	20.733	17.595
5	11.569	11.783	12.322	12.207
6	15.177	13.768	16.188	14.626
7	15.109	38.924	15.562	40.388
8	24.396	17.038	24.447	17.897
9	15.434	12.035	15.086	12.181
10	13.770	30.352	14.288	32.443
11	19.153	22.338	19.993	23.498
12	13.159	11.384	14.128	12.613
13	15.555	15.845	15.082	16.127
14	20.958	17.201	20.601	18.243
15	10.616	10.341	10.654	10.752
16	18.869	17.651	19.826	18.729
17	17.670	16.707	18.852	17.418
18	18.740	18.292	19.525	18.816
19	13.458	13.131	14.010	13.554
20	16.912	15.148	16.824	15.803
Average	15.717	17.590	16.113	18.411

Table 4.6: RRMSE for XGBOOST Combination 5 and Combination 6

Wind Farm	Combination 5 (%)	DSO (%)	Combination 6 (%)	DSO (%)
1	12.079	12.026	12.655	13.482
2	15.766	16.847	16.265	16.775
3	9.327	32.118	10.469	41.013
4	21.093	19.319	20.879	19.850
5	14.014	13.214	13.645	14.874
6	17.915	16.396	18.388	18.929
7	17.243	43.920	17.348	50.877
8	25.475	19.438	33.674	21.536
9	16.463	13.200	17.956	14.372
10	15.418	37.894	15.500	45.416
11	23.374	25.471	25.495	29.586
12	15.975	14.656	17.007	15.593
13	15.306	17.810	13.068	17.470
14	22.438	20.670	23.539	21.550
15	11.401	11.538	11.949	11.954
16	20.756	19.458	21.573	21.983
17	20.098	17.872	18.651	19.541
18	21.479	20.925	26.063	26.619
19	15.679	15.599	17.438	18.242
20	17.812	17.038	19.554	18.998
Average	17.456	20.270	18.556	22.933

Table 4.7: RRMSE for XGBOOST Combination 7 and Combination 8

Wind Farm	Combination 7 (%)	DSO (%)	Combination 8 (%)	DSO (%)
1	11.247	10.193	11.814	10.820
2	13.775	13.443	13.947	13.717
3	8.558	26.794	8.432	23.886
4	17.922	17.293	17.344	16.061
5	11.947	11.221	11.721	10.869
6	12.886	12.321	12.550	12.002
7	14.374	39.519	15.452	40.944
8	15.996	15.510	15.438	14.769
9	14.989	11.981	15.047	11.859
10	15.495	30.680	14.891	28.978
11	14.400	19.459	15.095	19.463
12	10.595	10.538	10.797	10.176
13	15.012	14.034	15.420	14.719
14	16.386	15.931	16.536	15.680
15	10.804	10.308	11.104	10.387
16	19.087	18.055	18.229	17.325
17	15.345	15.129	15.289	15.063
18	17.286	16.313	17.383	16.301
19	13.685	12.689	14.173	12.903
20	15.350	15.140	15.248	14.655
Average	14.257	16.827	14.296	16.529

The results obtained from Table 4.4 to Table 4.7 can be summarized in a graphical way through Figure 4.2, that presents the average RRMSE of the 20 wind farms for each combination, achieved by the XGBOOST model and by the DSO.

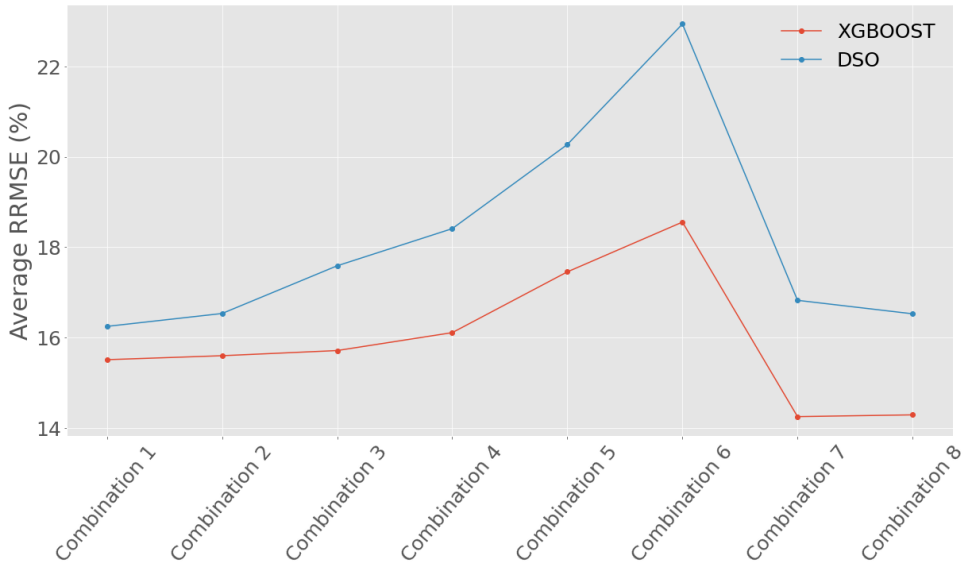


Figure 4.2: Average RRMSE for each combination

Figure 4.2 shows that first, the error of the XGBOOST model developed is always lower than the error of the actual system used by the DSO for any combination of training and test sets. Second, the XGBOOST model more accurately forecasts long periods of time like 6 months (Combination 7) or 1 complete year (Combination 8) instead of short periods of time like 1 month (Combination 6) or 2 months (Combination 5), that present the higher percentages of error. Third, the best combination found corresponds to Combination number 7: 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021) with an average RRMSE of 14.257%. From now on these training and test periods are used in all tests.

4.5 XGBOOST Hyperparameter Tuning

Hyperparameter tuning or hyperparameter optimization, is the process of determining the right combination of hyperparameters that maximizes a ML or AI model performance. It works by running multiple trials of different combinations of hyperparameters in a single training process. Once the process ends, it gives the set of hyperparameter values that are best suited for the model to give the most optimal result [53].

The hyperparameters of XGBOOST that are tuned are the following [54]:

- **max_depth:** Maximum depth per tree. A deeper tree might increase the performance, but also the complexity and chances to overfit. The value must be an integer greater than 0. Default is 6.
- **learning_rate:** Determines the step size at each iteration while the model optimizes toward its objective. A low learning rate makes computation slower, and requires more rounds to achieve the same reduction in residual error as a model with a high learning rate. The value must be between 0 and 1. Default is 0.3.
- **n_estimators:** The number of trees in the ensemble. Equivalent to the number of boosting rounds. The value must be an integer greater than 0. Default is 100.
- **colsample_bytree:** Represents the fraction of columns to be randomly sampled for each tree. It might improve overfitting. The value must be between 0 and 1. Default is 1.
- **subsample:** Represents the fraction of observations to be sampled for each tree. Lower values prevent overfitting, but might lead to underfitting. The value must be between 0 and 1. Default is 1.
- **min_child_weight:** Defines the minimum sum of weights of all observations required in a child. It is used to control overfitting. The larger it is, the more conservative the algorithm will be. The value must be an integer greater than 0. Default is 1.

To find the best combination of hyperparameters for the XGBOOST model, Random Search optimization algorithm[1] is used. It consists in a large range of hyperparameters values, which are randomly iterated a specific number of times over combinations of the values defined. The number of iterations and the metric used to evaluate the performance of the cross-validated model are specified.

To run Random Search, the ranges of values for each hyperparameter were defined as follows:

- *max_depth:* Integer in the range $1 - 6$.
- *learning_rate:* Number in the range $0 - 0.2$.
- *n_estimators:* Integer in the range $100 - 1000$.
- *colsample_bytree:* Number in the range $0.5 - 1$.
- *subsample:* Number in the range $0.5 - 1$.
- *min_child_weight:* Integer in the range $1 - 12$.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

The number of iterations defined for Random Search was 50 and the Mean Square Error (MSE) is the metric used to evaluate the performance for each combination of hyperparameters. Since Random Search requires to run 50 times the XGBOOST model and that is done for all the 20 wind farms, the computation time is very high and it takes a long time to get the results. Therefore, this process is done only once.

Table 4.8 presents the best combination of hyperparameters obtained for each wind farm after running Random Search and the average values of each hyperparameter, when considering the 20 wind farms all together.

Table 4.8: Best XGBOOST hyperparameters for each wind farm

Wind Farm	max_depth	learning_rate	n_estimators	colsample_bytree	subsample	min_child_weight
1	2	0.050	200	0.7	0.7	10
2	2	0.050	500	1.0	0.7	10
3	2	0.001	385	1.0	1.0	5
4	3	0.030	200	1.0	0.7	5
5	3	0.030	200	1.0	1.0	10
6	3	0.030	500	1.0	0.5	10
7	2	0.017	610	0.7	1.0	5
8	3	0.050	200	1.0	0.5	5
9	2	0.050	500	1.0	0.7	10
10	2	0.005	715	1.0	1.0	3
11	3	0.022	345	1.0	0.7	10
12	2	0.050	200	1.0	0.5	3
13	2	0.050	100	1.0	1.0	10
14	3	0.100	100	0.7	1.0	10
15	2	0.025	502	1.0	1.0	5
16	2	0.048	181	1.0	0.7	5
17	3	0.054	208	1.0	1.0	5
18	2	0.046	217	0.7	0.7	3
19	2	0.050	500	1.0	0.7	10
20	2	0.050	500	1.0	0.7	10
Average	2	0.04	343	0.9	0.8	7

Considering the obtained results, two tests, one using the best combination of hyperparameters for each wind farm and the other using the same average values of hyperparameters for all wind farms are performed. The idea is to compare the best RRMSE achieved so far, with the RRMSE obtained after the hyperparameter optimization. The results are presented in Table 4.9, using 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021), that was the best combination found in Section 4.4 and are the training and test periods that used from now on.

Table 4.9: RRMSE for XGBOOST after hyperparameter tuning: 1 year training, 6 months forecast

Wind Farm	Best results until now (%)	Using best combination of hyperparameters (%)	Using average values of hyperparameters (%)	DSO (%)
1	11.247	10.321	10.338	10.193
2	13.775	12.835	12.860	13.443
3	8.558	7.586	11.194	26.794
4	17.922	17.070	17.497	17.293
5	11.947	11.066	11.174	11.221
6	12.886	11.770	12.125	12.321
7	14.374	13.547	13.567	39.519
8	15.996	14.452	14.756	15.510
9	14.989	14.279	14.254	11.981
10	15.495	14.246	14.404	30.680
11	14.400	13.450	13.526	19.459
12	10.595	9.838	9.909	10.538
13	15.012	14.076	14.175	14.034
14	16.386	15.358	15.459	15.931
15	10.804	9.729	9.796	10.308
16	19.087	16.147	16.237	18.055
17	15.345	14.352	14.732	15.129
18	17.286	16.302	16.335	16.313
19	13.685	12.660	12.726	12.689
20	15.350	14.525	14.555	15.140
Average	14.257	13.180	13.481	16.827

From Table 4.9 it is possible to observe that the average RRMSE was reduced from 14.257% to 13.180% after the hyperparameter tuning done specifically for each wind farm, meaning an improvement of 7.55%. However, since the computation time required to run Random Search to find the best hyperparameters is around 12 hours per wind farm, the RRMSE was also computed using the average values of hyperparameters instead of the specific combination found for every wind farm. In this case the average RRMSE achieved was 13.481%, that is not far from the 13.180% obtained before. In both cases a considerable reduction of the error was achieved.

Hence, after the comparison between the two tests performed, it was decided that for future forecasts just the average combination of hyperparameters ($max\_depth = 2,\ learning\_rate = 0.04,\ n\_estimators = 343,\ colsample\_bytree = 0.9,\ subsample = 0.8,\ min\_child\_weight = 7$) will be used to run the XGBOOST model independently of the wind farm. This, considering that the DSO has 200 wind farms connected to the MV distribution network of Portugal and running Random Search for each one is not worth the computation time required for the little extra improvement obtained when calculating the best combination of hyperparameters specific for every wind farm.

4.6 XGBOOST Trying New Features

The next tests carried out with the aim of reducing the RRMSE of the model even more, consist of creating and testing two new features. One feature is related with the wind speed of previous days, while the other one is directly related to the error.

4.6.1 Wind Speed of Previous Days as a Feature

Creating new features from the existing data available, is sometimes a good technique used in ML to improve results. In this case, since wind speed is the feature that better correlates wind power generation, the idea is to incorporate not only the wind speed information of the forecast period (as the XGBOOST model normally do), but also include the wind speed information of 1 day, 2 days and even 3 days ahead the forecast period, as new features.

Table 4.10 presents the results for the wind speed of previous days features and the comparison of the RRMSE with the best results and with the DSO, for 1 year training and 6 months forecast.

Table 4.10: RRMSE for XGBOOST trying wind speed of previous days as a feature

Wind Farm	Best results (%)	Wind speed -1 (%)	Wind speed -1 and -2 (%)	Wind speed -1, -2 and -3 (%)	DSO (%)
1	10.338	10.770	10.746	10.755	10.193
2	12.860	13.181	13.144	13.128	13.443
3	11.194	7.681	7.670	7.711	26.794
4	17.497	17.516	17.898	17.603	17.293
5	11.174	11.360	11.189	11.266	11.221
6	12.125	12.519	12.607	12.649	12.321
7	13.567	13.974	13.986	14.010	39.519
8	14.756	15.417	15.615	15.677	15.510
9	14.254	14.493	14.556	14.508	11.981
10	14.404	14.844	14.897	14.924	30.680
11	13.526	13.832	14.120	14.004	19.459
12	9.909	10.027	10.065	10.218	10.538
13	14.175	14.780	14.717	14.726	14.034
14	15.459	15.637	15.647	16.122	15.931
15	9.796	10.150	10.189	10.233	10.308
16	16.237	19.327	19.302	19.242	18.055
17	14.732	15.083	15.217	15.261	15.129
18	16.335	16.567	16.672	16.714	16.313
19	12.726	13.120	13.111	13.265	12.689
20	14.555	15.219	15.014	15.347	15.140
Average	13.481	13.775	13.818	13.868	16.827

As shown in Table 4.10, three different simulations were performed by creating new wind speed features in the data. In the first simulation just the wind speed of the previous day (wind speed -1) was included, in the second simulation the wind speed of the previous two days (wind speed -1 and -2) were included and in the third simulation the wind speed of the previous three days (wind speed -1, -2 and -3) were included, before doing the forecast.

The results of this test are not as expected, because including the wind speed of previous days as a feature did not reduce the RRMSE of the model, on the contrary the error passed from 13.481% to 13.775%, 13.818% and 13.868% when using the wind speed information of 1 day, 2 days and 3 days ahead, respectively. Since no improvement in the XGBOOST model was achieved with this test, the wind speed of previous days as a feature is discarded.

4.6.2 Error as a Feature

The other feature created correspond to the error between predictions and real values for the training period. The idea is to forecast one part of the training set, then calculate the error of that forecast (using the actual values), include that error in the data as a new feature and finally re-run the XGBOOST model doing the normal forecast that has been used in the last sections but using the new information. The following procedure explains step by step how the process was done :

1. XGBOOST model with 6 months of training (JAN-JUN of 2020) was run to forecast the other 6 months of the same year (JUL-DEC of 2020).

2. The error for those 6 months (JUL-DEC of 2020) was computed as the difference between predictions and real values.

3. A new feature called 'Error' was included in the training data with the values obtained in the previous step.

4. XGBOOST model was re-run using JUL-DEC of 2020 data for training, which include now the Error feature and then the usual 6 months of 2021 (JAN-JUL) were forecast to make the results comparable with previous tests.

Table 4.11 presents the RRMSE obtained by trying the error of the training period as a new feature and the respective comparison with the best results and with the DSO.

Table 4.11: RRMSE for XGBOOST trying the error as a feature

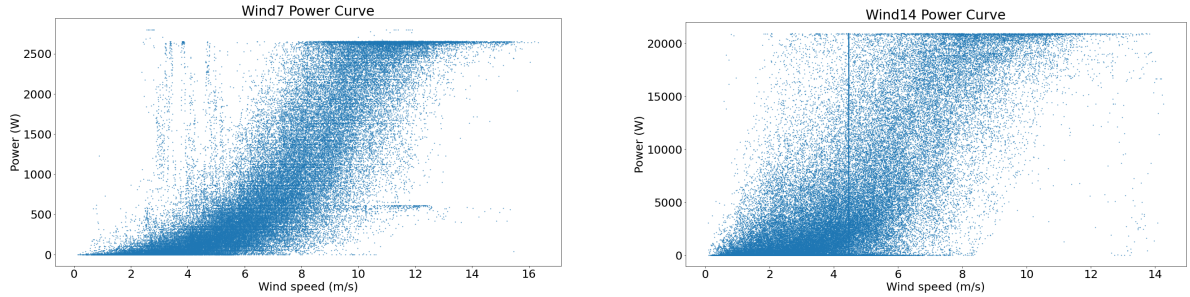
Wind Farm	Best Results (%)	Error Feature (%)	DSO (%)
1	10.338	10.685	10.193
2	12.860	13.906	13.443
3	11.194	7.950	26.794
4	17.497	18.222	17.293
5	11.174	11.286	11.221
6	12.125	12.116	12.321
7	13.567	17.283	39.519
8	14.756	14.699	15.510
9	14.254	19.567	11.981
10	14.404	15.120	30.680
11	13.526	14.006	19.459
12	9.909	9.731	10.538
13	14.175	15.328	14.034
14	15.459	15.476	15.931
15	9.796	10.839	10.308
16	16.237	19.207	18.055
17	14.732	14.740	15.129
18	16.335	16.729	16.313
19	12.726	13.183	12.689
20	14.555	14.991	15.140
Average	13.481	14.253	16.827

The results of this test are also not satisfactory, as shown in Table 4.11. By using the error of the training period as a feature, the RRMSE of the XGBOOST model increased from 13.481% to 14.253%. No improvement was achieved, therefore this new feature is also discarded.

4.7 XGBOOST Filtering the Power Curve

Since creating new features did not improve the XGBOOST model results, another approach based on filtering the training data is tested. The idea is to apply a filter to remove data that does not adjust to the theoretical power curves (Power vs Wind Speed) of the wind farms. By removing these 'outliers' from the training set, the model should have better data quality to learn from and the results might improve.

Wind farm 7 and wind farm 14 are used as example, to show what the power curve filtering algorithm does. These two wind farms are chosen, because the first one offers a more defined power curve, while the second one presents more dispersed data. Figure 4.3 presents the initial power curves of wind farm 7 in Figure 4.3(a) and wind farm 14 in Figure 4.3(b).



(a) Wind Farm 7 **(b)** Wind Farm 14

Figure 4.3: Initial power curves

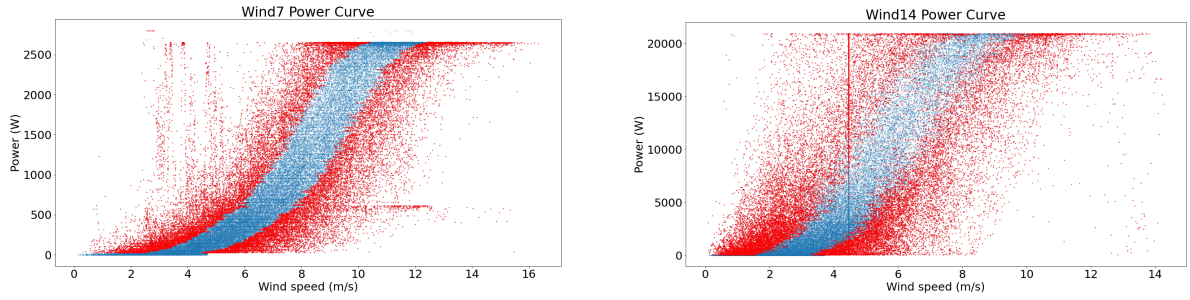
The graphs presented in Figure 4.3 were obtained by plotting the wind speed data from IPMA and the power generation measurements of each wind farm for the period $2020 - 2021$. From the graphs it is possible to observe that the power curves are not a continuous line like in the theory, but at least the set of points follows the same shape.

Thus, this wind speed and wind power data pass trough the filtering algorithm (Open OA library) that flags the 'outliers' of the power curve based on two conditions:

- Values with *low wind speed* and *high power*.
- Values with *high wind speed* and *low power*.

The algorithm works by dividing the data in bins, with a start and end point and specifying a bin width. The criteria for flagging is based on the standard deviation from the median of the bin center.

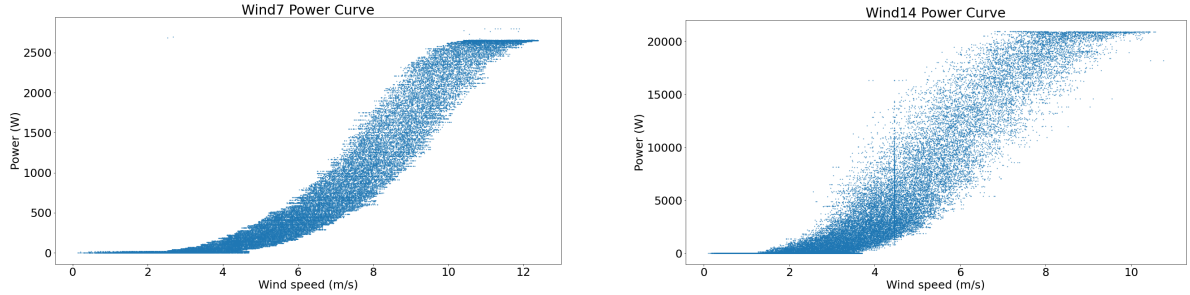
Figure 4.4 presents the power curves of wind farm 7 in Figure 4.4(a) and wind farm 14 in Figure 4.4(b), when applying the filtering. In red appears the 'outliers' that will be removed.



(a) Wind Farm 7 **(b)** Wind Farm 14

Figure 4.4: Applying the filtering to the power curves

After removing those 'outliers', the filtered power curves are obtained. Figure 4.5 presents the final power curves of wind farm 7 in Figure 4.5(a) and wind farm 14 in Figure 4.5(b) after the filtering algorithm.



(a) Wind Farm 7 **(b)** Wind Farm 14

Figure 4.5: Final power curves after filtering

The filtered power curves in Figure 4.5 now present a more similar shape to the behavior of a wind generator under normal operation. The next step is to run the XGBOOST model using just the data that passed the filtering, to see if the results improve or not. Table 4.12 presents the RRMSE for 1 year training and 6 months forecast, obtained when applying the filtering algorithm to the power curve.

Table 4.12: RRMSE for XGBOOST filtering the power curve

Wind Farm	Best Results (%)	Filtering power curve(%)	DSO (%)
1	10.338	10.882	10.193
2	12.860	13.578	13.443
3	11.194	7.552	26.794
4	17.497	20.594	17.293
5	11.174	12.199	11.221
6	12.125	14.648	12.321
7	13.567	14.334	39.519
8	15.756	16.522	15.510
9	14.254	15.655	11.981
10	14.404	15.502	30.680
11	13.526	14.173	19.459
12	9.909	10.513	10.538
13	14.175	15.576	14.034
14	15.459	16.718	15.931
15	9.796	10.267	10.308
16	18.237	19.620	18.055
17	14.732	16.165	15.129
18	16.335	17.350	16.313
19	12.726	13.054	12.689
20	14.555	15.205	15.140
Average	13.481	14.505	16.827

Unfortunately, the results shown in Table 4.12 are not as expected. The RRMSE obtained after filtering the power curve increased from 13.481% to 14.505%, when compared with the best results achieved. One possible explanation of this situation is that the test data (forecast period) also contains values that do not adjust to the power curve of the wind farm, and that are part of one of the two conditions mentioned before, but those values cannot be removed from the test data because then the forecasting would be incomplete.

Therefore, the model has been trained with 'good' quality data that adjust to the theory but in reality it has to forecast the power of the test set and its 'outliers', showing that the available data may not be the most optimal. Trying to further reduce the error might have reached the limit when using the meteorological data available.

4.8 XGBOOST with Backtesting

Backtesting is a term used in modeling that refers to testing a predictive model on historical data. It involves moving backward in time, step-by-step, in as many stages as it is necessary. Hence, it is a special type of cross-validation applied to previous periods [55].

To have a graphical understanding of how backtesting works in time series, Figure 4.6 shows an example of a time series backtesting diagram with an initial training size of 10 observations, a prediction horizon of 3 steps, and retraining at each iteration.

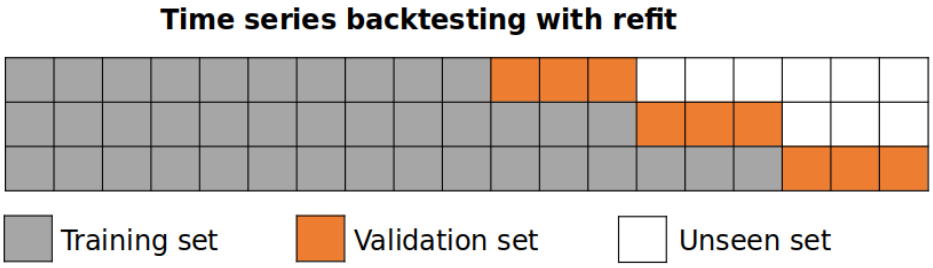


Figure 4.6: Time series backtesting example [55]

The purpose of this test is then to apply the backtesting with refit and increasing training size strategy inside the XGBOOST model, to see if the RRMSE can be reduced. To do that, the model is trained each time before making a new prediction, then that prediction is included in the training set and the process is repeated until all the predictions are made. That means that the model uses all the data available so far, while the training set increases sequentially, maintaining the temporal order of the data.

The initial training set in our case corresponds to 1 year of data (JAN-DEC of 2020), the prediction horizon correspond to 1 day (meaning that the model is trained in each iteration to forecast each day separately) and the retraining is done until the 6 months (JAN-JUN of 2021) that correspond to the forecast period are predicted. Table 4.13 presents the RRMSE achieved when using backtesting strategy implemented inside the XGBOOST model.

Table 4.13: RRMSE for XGBOOST using backtesting strategy

Wind Farm	Best Results (%)	Backtesting (%)	DSO (%)
1	10.338	10.053	10.193
2	12.860	12.568	13.443
3	11.194	9.212	26.794
4	17.497	16.475	17.293
5	11.174	10.683	11.221
6	12.125	-	12.321
7	13.567	13.247	39.519
8	14.756	14.109	15.510
9	14.254	13.740	11.981
10	14.404	14.031	30.680
11	13.526	12.901	19.459
12	9.909	9.355	10.538
13	14.175	13.983	14.034
14	15.459	14.809	15.931
15	9.796	9.575	10.308
16	16.237	17.668	18.055
17	14.732	14.153	15.129
18	16.335	15.899	16.313
19	12.726	12.547	12.689
20	14.555	13.832	15.140
Average	13.481	13.097	16.827

As shown in Table 4.13, the results obtained with backtesting are better than the best RRMSE achieved until now. There is a little improvement of 2.8%, since the error was reduced from 13.481% to 13.097%. However, when considering the computation time that backtesting requires, which is in average 10 hours per wind farm, the small reduction of the error makes not worth to implement this strategy into the model. For the DSO the main point is that the model is able to do the forecast in a short computing time because they have 200 wind farms connected to the MV distribution network of Portugal . The implemented XGBOOST model takes between $20 - 30$ seconds per wind farm to run, and with backtesting it takes 1500 times more. Since the accuracy of the forecast with backtesting does not represent a significant improvement, the inclusion of backtesting is discarded.

4.9 Stacking

Stacking is the process of using different ML and AI models one after another, where the predictions from each model are added as new features. It is done in layers, and there can be arbitrarily many layers, dependent on exactly how many models are trained, along with the best combination of these models. At the end, the final dataset combining the initial features plus the predictions created after each layer are feed into a last model. The last model is called a meta-learner, and its purpose is to generalize all the features from each layer into the final predictions [56].

Figure 4.7 shows the general structure of the stacking process. It is composed by two levels, in Level 1 M number of models are stacked in layers one by one and in Level 2 the best model is used as meta-learner to make the final predictions.

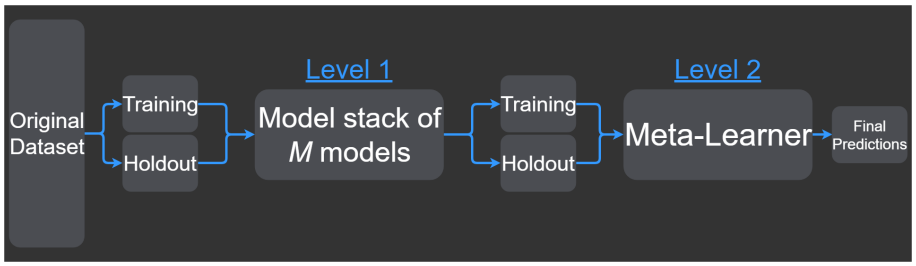


Figure 4.7: Stacking process structure [56]

Figure 4.8 presents the diagram of the stacking process implemented in this case. In Level 1, six layers were defined using the following models: Random Forest (RF), Light Gradient Boosting Machine (LGBM), Extreme Gradient Boosting (XGBOOST), Ridge, Lasso and Support Vector Machine (SVM). Then, in Level 2 the XGBOOST model was used as meta-learner to obtain the final predictions.

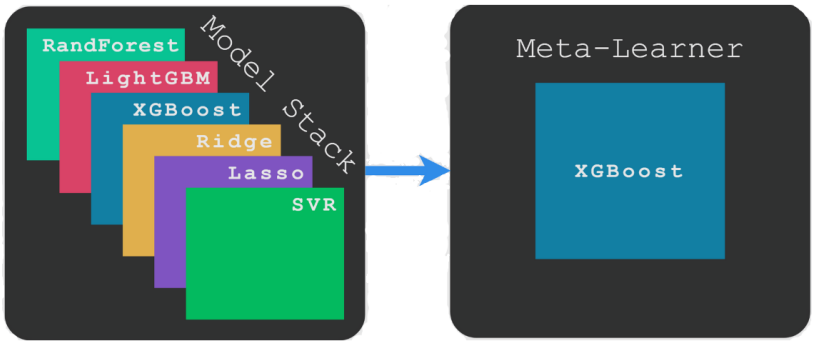


Figure 4.8: Stacking process implemented [56]

Table 4.14 presents the RRMSE obtained using the stacking approach with RF, LGBM, XGBOOST, Ridge, Lasso and SVM layers and XGBOOST meta-learner, for the same 1 year training and 6 months forecast of the last tests.

Table 4.14: RRMSE for stacking approach

Wind Farm	Best Results (%)	Stacking (%)	DSO (%)
1	10.338	10.358	10.193
2	12.860	12.856	13.443
3	11.194	8.793	26.794
4	17.497	17.685	17.293
5	11.174	11.136	11.221
6	12.125	12.225	12.321
7	13.567	13.589	39.519
8	14.756	14.619	15.510
9	14.254	14.077	11.981
10	14.404	14.731	30.680
11	13.526	13.569	19.459
12	9.909	9.974	10.538
13	14.175	14.585	14.034
14	15.459	15.394	15.931
15	9.796	9.853	10.308
16	16.237	16.288	18.055
17	14.732	14.912	15.129
18	16.335	16.134	16.313
19	12.726	12.689	12.689
20	14.555	14.375	15.140
Average	13.481	13.392	16.827

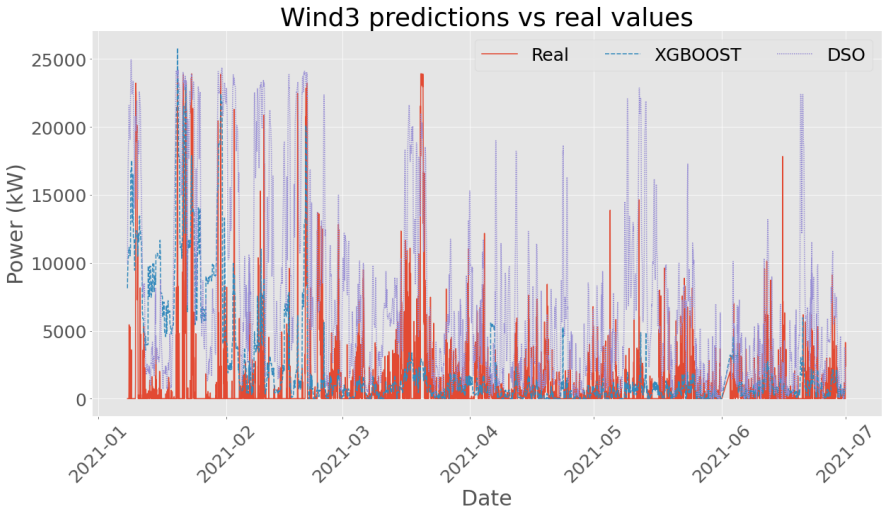
In this case, by using stacking the RRMSE passed from 13.481% to 13.392%, equivalent to a 0.66% of improvement. Regarding the computation time required by this approach, for each wind farm it takes on average 15 minutes to run, that is 40 times more than the normal XGBOOST (that takes between $20 - 30$ seconds to run). Therefore, even when the RRMSE results are better when using stacking, the little reduction of the error is not worth the extra computation time and this approach is discarded.

4.10 Best Results and RRMSE Analysis

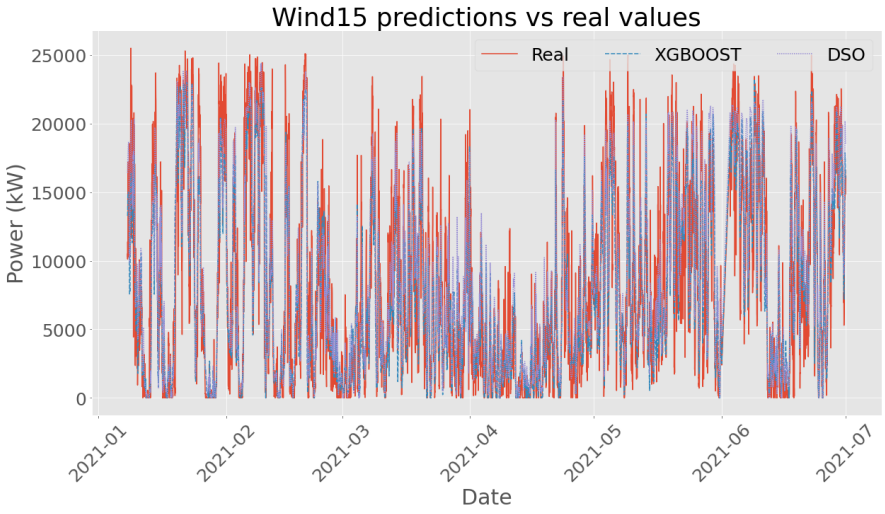
After all the tests performed, the best average RRMSE achieved for the implemented XGBOOST model corresponds to 13.481%, neglecting backtesting and stacking that achieved a lower RRMSE but were discarded due to the computation time required. The training and forecast periods of the best results correspond to 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021).

Now a comparison analysis of the best results and the RRMSE distribution of two wind farms is presented. Wind farm 15 is chosen since it has the lower RRMSE from all wind farms: 9.796%. The other wind farm chosen is wind farm 3 because it presents a big difference between the RRMSE of XGBOOST and the DSO, 11.194% and 26.794% respectively.

Figure 4.9 shows the comparison between XGBOOST predictions, DSO predictions and the real values of power for the six months forecast. Figure 4.9(a) corresponds to wind farm 3 and Figure 4.9(b) corresponds to wind farm 15.



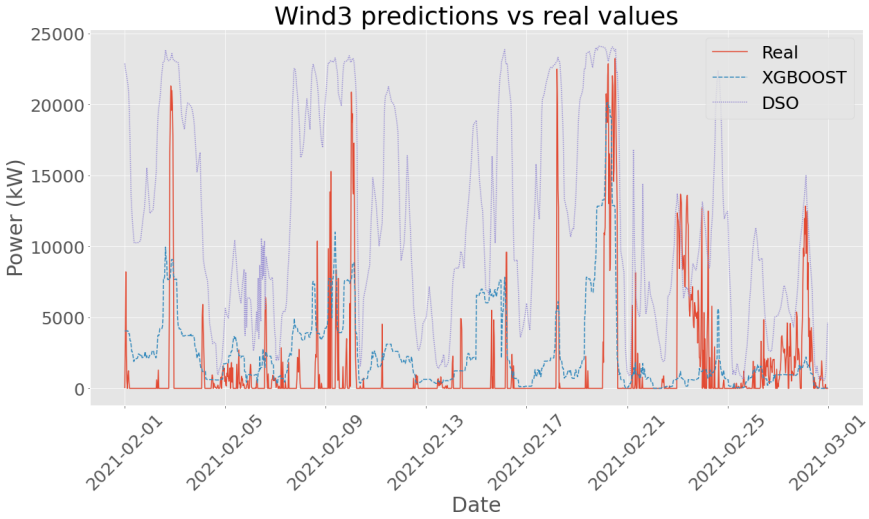
(a) Wind Farm 3



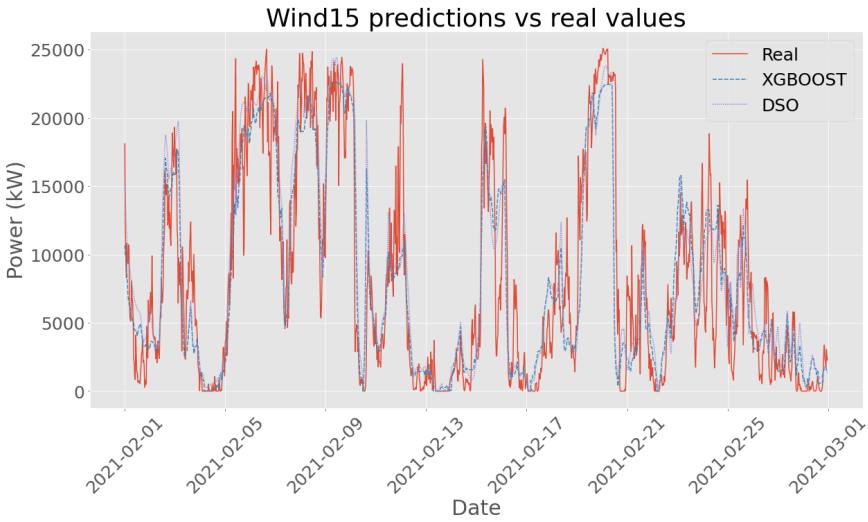
(b) Wind Farm 15

Figure 4.9: Forecast vs real values for JAN-JUN of 2021

Looking into more detail, Figure 4.10 presents the same comparison between XGBOOST predictions, DSO predictions and the real values of power for wind farm 3 and wind farm 15, but focusing now in just one month of the forecast period, specifically February.



(a) Wind Farm 3

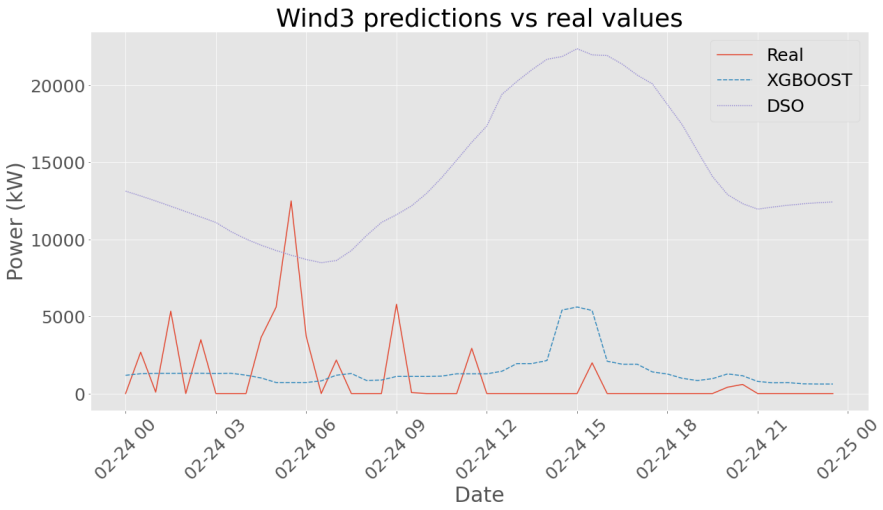


(b) Wind Farm 15

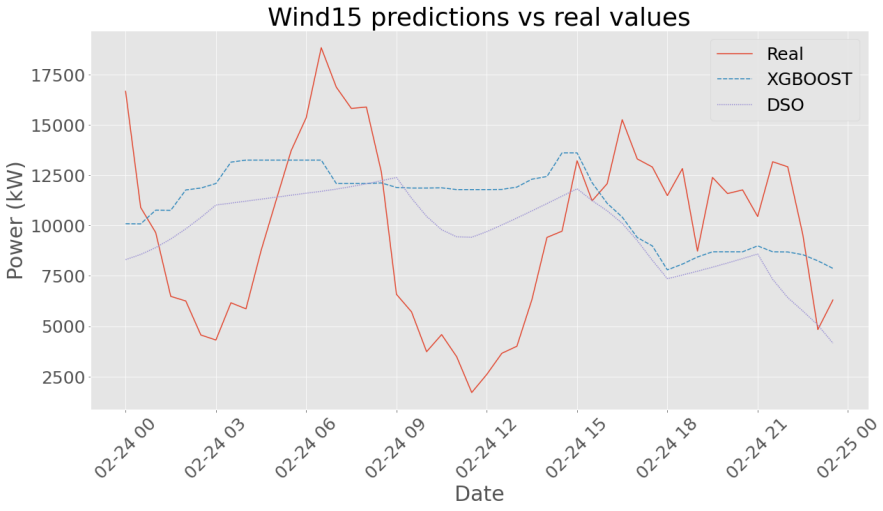
Figure 4.10: Forecast vs real values for FEB of 2021

From Figure 4.10 it is already possible to observe that some of the predicted values of power are above the actual values but others are below, so there is no clear trend but in general the results of the XGBOOST model present a shape and a behavior similar to the real measurements. For wind farm 3 in Figure 4.10(a), it is possible to notice that the DSO predictions are well above the actual values, which explains why the DSO error is so big for this wind farm.

Finally, Figure 4.11 presents the same comparison but for a specific day of February, the 24th to be more precise. This allows to observe the actual differences between XGBOOST predictions (in blue), DSO predictions (in purple) and the real values of power (in red). Figure 4.11(a) corresponds to wind farm 3 and Figure 4.11(b) corresponds to wind farm 15.



(a) Wind Farm 3



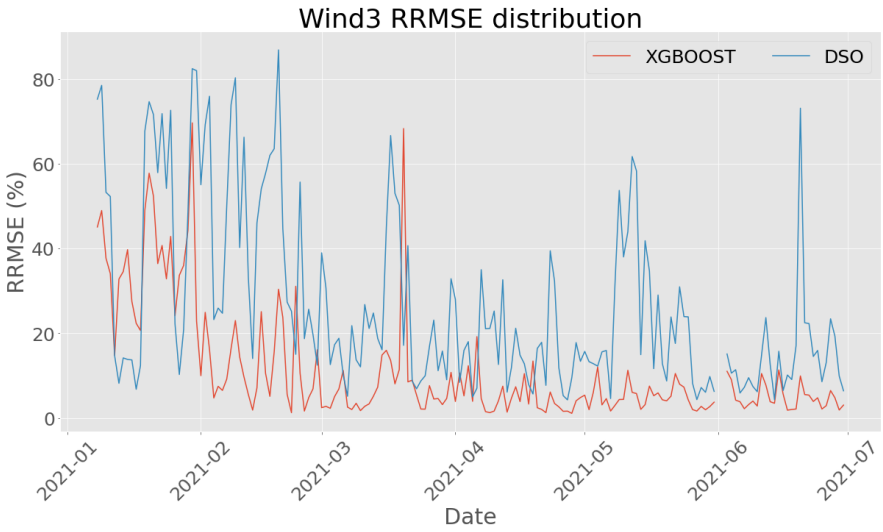
(b) Wind Farm 15

Figure 4.11: Forecast vs real values for the 24th of February of 2021

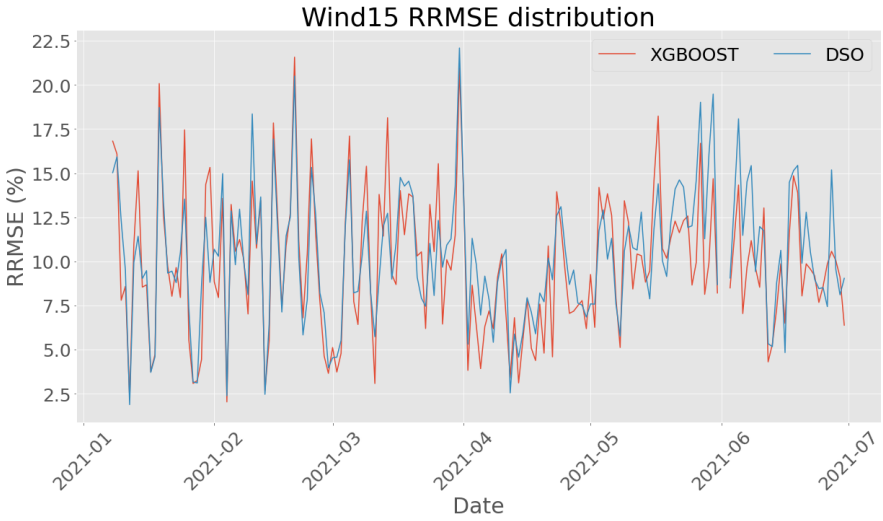
What Figure 4.11 shows is that the XGBOOST model and the forecasting system used by the DSO cannot capture all the power fluctuations of wind for small periods of time like hours. This is normal as consequence of the variability and volatility of wind speed, added to the fact that the meteorological data

available may not be optimal. However, what matters is that in the big picture the XGBOOST predictions are accurate enough and give a good approximation of the wind power generation. More importantly, when comparing XGBOOST with the DSO, not only the results of the model outperformed the DSO performance, also the fitting of XGBOOST seems to be more accurate.

Now looking at the error metric between predictions and real values, Figure 4.12 presents the RRMSE distribution for the six months forecast. Figure 4.12(a) corresponds to wind farm 3 and Figure 4.12(b) corresponds to wind farm 15.



(a) Wind Farm 3

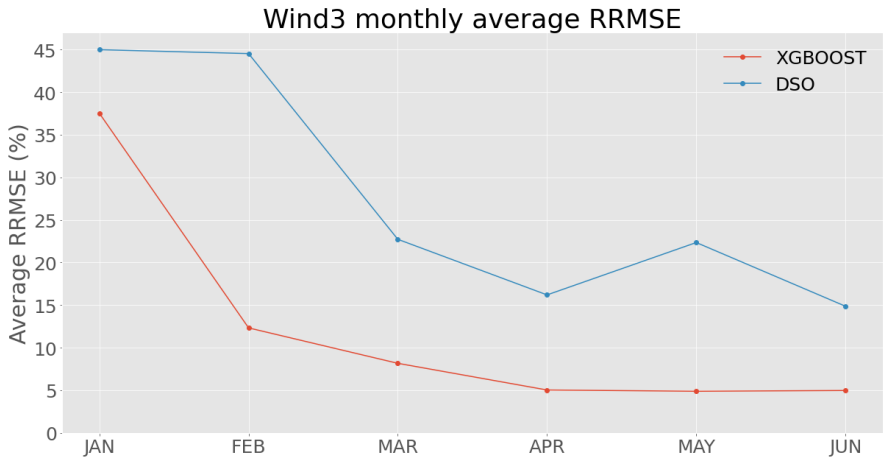


(b) Wind Farm 15

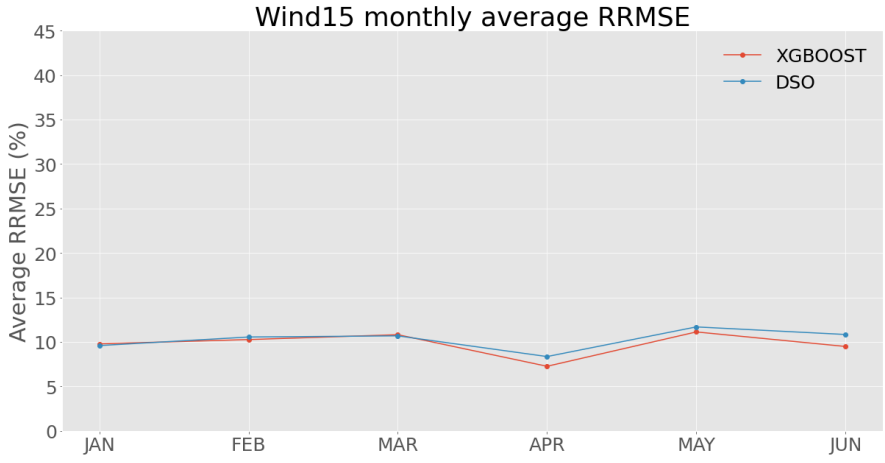
Figure 4.12: RRMSE distribution for the 6 months forecast

Figure 4.12 shows that the RRMSE also fluctuates a lot. There is no tendency because the error can go up or down inside each month. For wind farm 3 in Figure 4.12(a), the DSO error is higher than XGBOOST in almost all points; while for wind farm 15 in Figure 4.12(b) there is no big difference between XGBOOST and the DSO, both curves present a similar behavior.

Finally, Figure 4.13 presents the monthly average RRMSE for the 6 months forecast. Figure 4.13(a) corresponds to wind farm 3 and Figure 4.13(b) corresponds to wind farm 15.



(a) Wind Farm 3



(b) Wind Farm 15

Figure 4.13: Monthly average RRMSE for the 6 months forecast

In the case of Figure 4.13(a) corresponding to wind farm 3, it is possible to observe an important reduction of the error, specifically there is a 58% of improvement when comparing XGBOOST results with the DSO results. In all 6 months the average RRMSE of XGBOOST is lower than the DSO, which means that the forecast values obtained with our model are closer to the real values. In general, XGBOOST results are a good approximation; the error remains lower than 15% in almost all months, with the only exception of January, where the average RRMSE is relatively high.

In the case of Figure 4.13(b) corresponding to wind farm 15, the average RRMSE is basically the same between XGBOOST and the DSO, with a little improvement in the months of April, May and June when doing the forecast with the XGBOOST model. For wind farm 15 the error is stable along all 6 months and it represents the best result obtained from all wind farms, since the average RRMSE achieved was 9.8%, when using XGBOOST.

5

Conclusions and Future Work

Contents

5.1 Conclusions

In this work eight different forecasting models namely, Persistence, Auto-Regressive (AR), Auto-Regressive with Exogenous Variable (ARX), Long Short-Term Memory (LSTM) neural network, Decision Trees (DT), Random Forest (RF), Extreme Gradient Boosting (XGBOOST) and Support Vector Machine (SVM) were developed and tested to predict the power generation of 20 wind farms connected to the secondary substations of the MV distribution network of Portugal mainland.

The historical wind power generation data used was provided by the Portuguese DSO and corresponds to seven years of data, from 2015 to 2021. The DSO also included their own predictions for the period 2020-2021 obtained through their actual forecasting system. The meteorological data was obtained from two sources, IPMA (Instituto Português do Mar e da Atmosfera) and ISTMeteo (the meteorological investigation group of IST). IPMA data contains two years of information, 2020 and 2021; while ISTMeteo data contains seven months of information, from June to December of 2021.

To define which meteorological data to use, IPMA and ISTMeteo datasets were tested under the same forecast conditions, 6 months training (JUN-NOV of 2021) and 1 month forecast (DEC of 2021), using ARX method. The test determined that IPMA datasets offer better data quality and higher correlation with wind power generation than ISTMeteo, since IPMA presents a lower RRMSE (21.046%) than ISTMeteo (21.337%). Consequently, IPMA datasets were chosen to run all the forecasting models.

Regarding the methodology implemented in this thesis, it consisted of a framework with five stages: *Pre-Processing*, where the initial data was cleaned and the missing values were handled, *EDA and Feature Selection*, where the feature selection was carried out, *Forecasting Models*, where the final data was divided into training and testing and the predictions for each model were produced, *Post-Processing*, where the forecast results were checked, adjusted (if necessary) and saved and *Validation*, where the error metric (RRMSE) was calculated and reported to evaluate the performance of the models.

After comparing the eight models between them and with the DSO predictions, the results showed that for 6 months training (JUN-NOV of 2021) and 1 month forecast (DEC of 2021), XGBOOST obtained the best performance with a RRMSE of 18.613%, followed by RF with a RRMSE of 20.354% and ARX with a RRMSE of 21.046%. The rest of the models obtained an error that is higher than the error of the DSO predictions for the same period, which corresponds to a RRMSE of 22.904%. Specifically, LSTM neural network, DT, AR, SVM and Persistence obtained respectively a RRMSE of 24.160%, 24.381%, 27.522%, 29.822% and 31.838%.

With XGBOOST as the best-suited forecasting model for the wind farms analyzed, some tests and improvements were performed to this method in order to reduce the error as much as possible. It was found that the best combination of training and test periods based on the two years of information available for IPMA, corresponds to 1 year of training (JAN-DEC of 2020) and 6 months of forecast (JAN-JUN of 2021). When using this specific combination the average RRMSE gets reduced to 14.257%.

A hyperparameter tuning of XGBOOST using Random Search optimization was carried out to improve the previous result. The best combination of hyperparameters were found for each wind farm and the average RRMSE got reduced to 13.180%. However, since the computation time to run Random Search (around 12 hours) is very high, it was decided to use the average values of the hyperparameters independently of the wind farm. Those values correspond to a $max\_depth = 2$, $learning\_rate = 0.04$, $n\_estimators = 343$, $colsample\_bytree = 0.9$, $subsample = 0.8$ and $min\_child\_weight = 7$. With this combination of hyperparameters the RRMSE achieved is 13.481%, that is not so far from the value obtained using the best combination of hyperparameters, and therefore this approach should be used for future forecasts or with new wind farms.

Other improvements that lowered the best RRMSE (13.481%) of the developed XGBOOST model were achieved using backtesting and stacking approaches. In the case of backtesting the RRMSE got reduced to 13.097%, while for stacking the RRMSE got reduced to 13.392%. Nevertheless, both processes require a longer computation time, 10 hours per wind farm for backtesting and 15 minutes per wind farm for stacking, than the normal XGBOOST model which takes only between 20 to 30 seconds per wind farm to run. Since one of the most important characteristics of a forecasting model is to make predictions in an efficient way, meaning rapidly and with accuracy, it was concluded that the small reduction of the error achieved with this strategies is not worth the large computation time needed and consequently, backtesting and stacking were discarded.

After all, using the proposed XGBOOST model for 1 year training (JAN-DEC of 2020) and 6 months forecast (JAN-JUN of 2021), the best average RRMSE achieved for the 20 wind farms studied, corresponds to 13.481%; after discarding Random Search, backtesting and stacking of course. The results successfully fulfilled the main goal of this thesis, that was to improve the performance of the actual DSO forecasting system, which for the same period of analysis presents a RRMSE of 16.827%. With the XGBOOST model developed an improvement of 20% is achieved. The framework is scalable, computationally efficient and can be used for future wind power forecasting, if the DSO want to obtain predictions with higher accuracy.

5.2 Limitations

The limitations encountered during the development and implementation of the models are mainly related with the fluctuating nature of wind. Since wind is an intermittent source and wind speed and wind direction change very quickly, WPGF is a challenging task. The variations on wind conditions could explain somehow why the lowest percentage of error achieved with the XGBOOST model developed was 13.481% and why after doing a lot of tests or trying different approaches, none or little improvement of the error was obtained, but with a high computational cost that at the end makes it non-viable.

Another limitation may have been the data used. It was shown through the power curves of the wind farms that the meteorological data available may not be optimal, because there is a lot of dispersion and for some wind farms the behavior or shape of the power curve differs significantly from the theory. The main reason of this is that we do not have exactly the weather data in each wind farm, we have just the NWP of IPMA and ISTMeteo in an area of $14\ km^2$, hence, it is possible that with better meteorological data quality or with the specific weather data of the wind farms the results might improve.

Moreover, for ISTMeteo there was limited meteorological data available, just 7 months of information, which also limited the period of training and forecast to that time. And although for IPMA there was more data available, 2 years of information, it is still a reduced amount when compared to the 7 years of wind power data provided by the DSO.

5.3 Future Work

After the completion of this thesis, some ideas that appear as an opportunity for future improvements are: explore other strategies to fill the missing data or to remove the outliers during the pre-processing, test the implemented models with other wind farms (new or in different locations) and most importantly, try other datasets or other sources where to get the meteorological data, because that was the main limitation in this case.

As a future research, the XGBOOST model developed should be tested to forecast the power generation of other renewable technologies connected to the distribution network, such as Solar Photovoltaic (PV) or hydropower, and the performance of those technologies must be compared with the results obtained in this thesis for wind. Finally, new forecasting models for wind power generation should be developed and tested, considering not only their performance in terms of error metrics but also bearing in mind the computation time required to run those models, which is as determinant factor as the accuracy itself.

Bibliography

[1] K. L. Jørgensen and H. R. Shaker, "Wind power forecasting using machine learning: State of the art, trends and challenges," in *2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE)*. IEEE, 2020, pp. 44–50.

[2] S. Micheli, "Policy strategy cooperation in the 2030 climate and energy policy framework," *Atlantic Economic Journal*, vol. 48, no. 2, pp. 265–267, 2020. [Online]. Available: http://dx.doi.org/10.1.1/jpb001

[3] M. S. Javed, T. Ma, J. Jurasz, and M. Y. Amin, "Solar and wind power generation systems with pumped hydro storage: Review and future perspectives," *Renewable Energy*, vol. 148, pp. 176–192, 2020.

[4] G. W. E. Council, "GWEC — Global Wind Report 2021," *Global Wind Energy Council: Brussels, Belgium*, 2021.

[5] Energias endógenas de Portugal, "Wind farms in portugal," Institute of Mechanical Engineering and Industrial Management (INEGI), Portuguese Renewable Energy Association (APREN), Tech. Rep., 2018.

[6] W. E. Newsroom. Clean electricity made in portugal will create jobs and lower consumers' energy bills. [Online]. Available: https://windeurope.org/newsroom/news/clean-electricity-made-in-portugal-will-create-jobs-and-lower-consumers-energy-bills/

[7] J. W. Taylor, P. E. McSharry, and R. Buizza, "Wind power density forecasting using ensemble predictions and time series models," *IEEE Transactions on Energy conversion*, vol. 24, no. 3, pp. 775–782, 2009.

[8] P. Wilczek, "Connecting the dots: distribution grid investments to power the energy transition," in *11th Solar & Storage Power System Integration Workshop (SIW 2021)*, vol. 2021. IET, 2021, pp. 1–18.

[9] S. A. Vargas, G. R. T. Esteves, P. M. Maçaira, B. Q. Bastos, F. L. C. Oliveira, and R. C. Souza, "Wind power generation: A review and a research agenda," *Journal of Cleaner Production*, vol. 218, pp. 850–870, 2019.

[10] B. Ernst, B. Oakleaf, M. L. Ahlstrom, M. Lange, C. Moehrlen, B. Lange, U. Focken, and K. Rohrig, "Predicting the wind," *IEEE power and energy magazine*, vol. 5, no. 6, pp. 78–89, 2007.

[11] C. Monteiro, R. Bessa, V. Miranda, A. Botterud, J. Wang, and G. Conzelmann, "Wind power forecasting: State-of-the-art 2009." Argonne National Lab.(ANL), Argonne, IL (United States), Tech. Rep., 2009.

[12] A. M. Foley, P. G. Leahy, A. Marvuglia, and E. J. McKeogh, "Current methods and advances in forecasting of wind power generation," *Renewable energy*, vol. 37, no. 1, pp. 1–8, 2012.

[13] S. S. Soman, H. Zareipour, O. Malik, and P. Mandal, "A review of wind power and wind speed forecasting methods with different time horizons," in *North American Power Symposium 2010*. IEEE, 2010, pp. 1–8.

[14] Y. Wu and J. Hong, "A literature review of wind forecasting technology in the world," in *2007 IEEE Lausanne Power Tech*. IEEE, 2007, pp. 504–509.

[15] D. R. Chandra, M. S. Kumari, and M. Sydulu, "A detailed literature review on wind forecasting," in *2013 International Conference on Power, Energy and Control (ICPEC)*. IEEE, 2013, pp. 630–634.

[16] S. Dutta, Y. Li, A. Venkataraman, L. M. Costa, T. Jiang, R. Plana, P. Tordjman, F. H. Choo, C. F. Foo, and H. B. Puttgen, "Load and renewable energy forecasting for a microgrid using persistence technique," *Energy Procedia*, vol. 143, pp. 617–622, 2017.

[17] W. Y. Cheng, Y. Liu, Y. Liu, Y. Zhang, W. P. Mahoney, and T. T. Warner, "The impact of model physics on numerical wind forecasts," *Renewable Energy*, vol. 55, pp. 347–356, 2013.

[18] J. Jung and R. P. Broadwater, "Current status and future advances for wind speed and power forecasting," *Renewable and Sustainable Energy Reviews*, vol. 31, pp. 762–777, 2014.

[19] A. Yamaguchi, T. Ishihara, and Y. Fujino, "An assessment of offshore wind energy potential using mesoscale model and gis," in *Proceedings of European Wind Energy Conference*, 2004.

[20] F. Cassola and M. Burlando, "Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output," *Applied energy*, vol. 99, pp. 154–166, 2012.

[21] C. Gallego, P. Pinson, H. Madsen, A. Costa, and A. Cuerva, "Influence of local wind speed and direction on wind power dynamics–application to offshore very short-term forecasting," *Applied Energy*, vol. 88, no. 11, pp. 4087–4096, 2011.

[22] M. J. Duran, D. Cros, and J. Riquelme, “Short-term wind power forecast based on arx models,” *Journal of Energy Engineering*, vol. 133, no. 3, pp. 172–180, 2007.

[23] J. L. Torres, A. Garcia, M. De Blas, and A. De Francisco, “Forecast of hourly average wind speed with arma models in navarre (spain),” *Solar energy*, vol. 79, no. 1, pp. 65–77, 2005.

[24] M. Milligan, M. Schwartz, and Y.-h. Wan, “Statistical wind power forecasting models: Results for us wind farms,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2003.

[25] E. Yatiyana, S. Rajakaruna, and A. Ghosh, “Wind speed and direction forecasting for wind power generation using arima model,” in *2017 Australasian Universities Power Engineering Conference (AUPEC)*. IEEE, 2017, pp. 1–6.

[26] R. G. Kavasseri and K. Seetharaman, “Day-ahead wind speed forecasting using f-arima models,” *Renewable Energy*, vol. 34, no. 5, pp. 1388–1393, 2009.

[27] E. Machado, T. Pinto, V. Guedes, and H. Morais, “Electrical load demand forecasting using feed-forward neural networks,” *Energies*, vol. 14, no. 22, p. 7644, 2021.

[28] S. Hanifi, X. Liu, Z. Lin, and S. Lotfian, “A critical review of wind power forecasting methods—past, present and future,” *Energies*, vol. 13, no. 15, p. 3764, 2020.

[29] A. More and M. Deo, “Forecasting wind with neural networks,” *Marine structures*, vol. 16, no. 1, pp. 35–49, 2003.

[30] J. d. S. Catalão, H. M. I. Pousinho, and V. M. F. Mendes, “Short-term wind power forecasting in portugal by neural networks and wavelet transform,” *Renewable energy*, vol. 36, no. 4, pp. 1245–1251, 2011.

[31] M. C. Mabel and E. Fernandez, “Analysis of wind power generation and prediction using ann: A case study,” *Renewable energy*, vol. 33, no. 5, pp. 986–992, 2008.

[32] S. Khazaei, M. Ehsan, S. Soleymani, and H. Mohammadnezhad-Shourkaei, “A high-accuracy hybrid method for short-term wind power forecasting,” *Energy*, vol. 238, p. 122020, 2022.

[33] X. Qin, C. Jiang, and J. Wang, “Online clustering for wind speed forecasting based on combination of rbf neural network and persistence method,” in *2011 Chinese Control and Decision Conference (CCDC)*. IEEE, 2011, pp. 2798–2802.

[34] J. Shi, J. Guo, and S. Zheng, “Evaluation of hybrid forecasting approaches for wind speed and power generation time series,” *Renewable and Sustainable Energy Reviews*, vol. 16, no. 5, pp. 3471–3480, 2012.

[35] Q. Phan, Y. Wu, and Q. Phan, "A comparative analysis of xgboost and temporal convolutional network models for wind power forecasting," in *2020 International Symposium on Computer, Consumer and Control (IS3C)*. IEEE, 2020, pp. 416–419.

[36] Q. Chen and K. Folly, "Wind power forecasting," *IFAC-PapersOnLine*, vol. 51, no. 28, pp. 414–419, 2018.

[37] A. Lahouar and J. B. H. Slama, "Hour-ahead wind power forecast based on random forests," *Renewable energy*, vol. 109, pp. 529–541, 2017.

[38] S. Salcedo-Sanz, E. G. Ortiz-Garcı, Á. M. Pérez-Bellido, A. Portilla-Figueras, L. Prieto *et al.*, "Short term wind speed prediction based on evolutionary support vector regression algorithms," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4052–4057, 2011.

[39] H. Zheng and Y. Wu, "A xgboost model with weather similarity analysis and feature engineering for short-term wind power forecasting," *Applied Sciences*, vol. 9, no. 15, p. 3019, 2019.

[40] F. Castellanos and N. James, "Average hourly wind speed forecasting with anfis," in *11th Americas Conference on Wind Engineering*, 2009, pp. 26–29.

[41] Y. Kassa, J. Zhang, D. Zheng, and D. Wei, "Short term wind power prediction using anfis," in *2016 IEEE international conference on power and renewable energy (ICPRE)*. IEEE, 2016, pp. 388–393.

[42] L. Fugon, J. Juban, and G. Kariniotakis, "Data mining for wind power forecasting," in *European Wind Energy Conference & Exhibition EWEC 2008*. EWEC, 2008, pp. 6–pages.

[43] M. A. Mohandes, T. O. Halawani, S. Rehman, and A. A. Hussain, "Support vector machines for wind speed prediction," *Renewable energy*, vol. 29, no. 6, pp. 939–947, 2004.

[44] Feature selection techniques in machine learning. [Online]. Available: https://www.javatpoint.com/feature-selection-techniques-in-machine-learning

[45] G. Santamaría-Bonfil, A. Reyes-Ballesteros, and C. Gershenson, "Wind speed forecasting for wind farms: A method based on support vector regression," *Renewable Energy*, vol. 85, pp. 790–809, 2016.

[46] H. N. Akouemo and R. J. Povinelli, "Data improving in time series using arx and ann models," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3352–3359, 2017.

[47] A. Moghar and M. Hamiche, "Stock market prediction using lstm recurrent neural network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, 2020.

[48] Seldon, “Decision trees in machine learning explained,” Nov 2021. [Online]. Available: https://www.seldon.io/decision-trees-in-machine-learning

[49] A. Chaudhary, A. Sharma, A. Kumar, K. Dikshit, and N. Kumar, “Short term wind power forecasting using machine learning techniques,” *Journal of Statistics and Management Systems*, vol. 23, no. 1, pp. 145–156, 2020.

[50] A. Ahmadi, M. Nabipour, B. Mohammadi-Ivatloo, A. M. Amani, S. Rho, and M. J. Piran, “Long-term wind power forecasting using tree-based learning algorithms,” *IEEE Access*, vol. 8, pp. 151 511–151 522, 2020.

[51] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, “Extreme gradient boosting machine learning algorithm for safe auto insurance operations,” in *2019 IEEE international conference on vehicular electronics and safety (ICVES)*. IEEE, 2019, pp. 1–5.

[52] 3.1. cross-validation: Evaluating estimator performance. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

[53] S. ES and A. Bajaj, “Hyperparameter tuning in python: A complete guide,” Mar 2022. [Online]. Available: https://neptune.ai/blog/hyperparameter-tuning-in-python-complete-guide

[54] D. Martins, “Xgboost: A complete guide to fine-tune and optimize your model,” Dec 2021. [Online]. Available: https://towardsdatascience.com/xgboost-fine-tune-and-optimize-your-model-23d996fab663

[55] J. Amat and J. Ortiz, “Skforecast: Time series forecasting with python and scikit-learn,” Feb 2021. [Online]. Available: https://www.cienciadedatos.net/documentos/py27-time-series-forecasting-python-scikitlearn.html

[56] C. Hansen, “Model stacking explained and python code,” Jan 2020. [Online]. Available: https://mlfromscratch.com/model-stacking-explained/