**Instituto Superior de Engenharia de Lisboa**
**Área Departamental de Matemática**

# Machine Learning in Wind Power Forecast

Diogo Camilo de Carvalho

**Licenciatura em Matemática Aplicada à Tecnologia e à Empresa**

**Júri:**

Prof. Gonçalo Marques

**Orientadores:**

Prof. Carlos Geraldes

Prof. Carlos Leandro

Eng.º Rui Pestana

2021

# Resumo

A atenção nas energias renováveis tem vindo a aumentar com o decorrer dos anos, nomeadamente, no capítulo da energia eólica. Com isto, a necessidade de se fazer previsão a curto e médio prazo da produção de energia neste setor tem aumentado, de modo, a melhorar a segurança e o sucesso económico das redes de energia. Historicamente, métodos numéricos de previsão meteorológicos (NWP) são utilizados, porém, devido às características de instabilidade e volatilidade nos sistemas de energia eólica, é difícil obter bons resultados, com estes métodos. Assim, o objetivo deste trabalho, passa por explorar metodologias que possam ser escalados ao *Big Data* para a previsões de curto (6 horas) e médio prazo (3 dias) de energia eólica.

Os resultados mostram que o sistema desenvolvido é uma ferramenta útil e fiável. As previsões de energia eólica podem ser significativamente melhoradas com um pré-processamento que tire vantagem das características dos dados utilizados. Assim, é possível obter um erro inferior a 10% para previsões a curto prazo e um erro entre os 10 e 15% para previsões a médio prazo para os dados de 2019, demonstrando a precisão do sistema.

Em termos gerais, pode-se concluir que as previsões obtidas estão ao mesmo nível dos modelos estado da arte utilizados pela REN, dando assim um contributo positivo à REN na previsão da potência de energia eólica.

# Palavras Chave

previsão, previsão energia eólica, processamento de dados, aprendizagem automática, modelos estatísticos, , Gradient Boosting Machines, XGBoost, LightGBM, ARIMAX, GAMAR

# Abstract

Renewable energies have recently gained considerable attention, in particular, the interest in wind energy is rapidly growing globally. With this, the importance of accurate short-term wind power production forecasting for improving the security and economic success of the power grid is growing as well. Historically, Numerical Weather Prediction (NWP) methods were used, however, the characteristics of instability and volatility on wind energy systems make it very difficult to obtain good results using such methods. To address those issues, the aim of the present work is to explore statistical methods that support Big Data to make short-range (6 h) and medium-term (3 days) predictions.

The results show that the developed system can be a useful and reliable tool. Wind-power forecasts can be significantly improved with feature engineering that takes advantage of the characteristics of the data that is being considered. Thus, it is possible to obtain an error of below 10% for short-term forecasts and an error between 10 and 15% for medium-term forecasts for the 2019 data, demonstrating the precision of the system.

In general terms, it can be concluded that the forecasts obtained are at the same level as the state-of-the-art models used by REN, thus giving a positive contribution to REN in forecasting wind energy power.

# Keywords

forecasts, wind-power forecasts, feature engineering, machine learning, statistical models, Gradient Boosting Machines, XGBoost, LightGBM, ARIMAX, GAMAR

# Contents

**List of Figures** viii

**List of Tables** ix

**Acronyms** x

**1 Introduction** 1

1.1 Motivation . . . 1

1.2 Work overview . . . 2

**2 State of the art** 4

2.1 Physical or Power curve approach . . . 4

2.2 Statistical approach . . . 5

2.3 Combination approach . . . 6

**3 NWP Data** 7

3.1 NWP input data . . . 7

3.2 Exploratory Data Analysis . . . 9

**4 Feature Engineering** 13

4.1 Feature Selection . . . 13

4.2 Layer's Interactions . . . 17

4.3 Lag Features . . . 18

**5 Prediction** 21

5.1 Performance Indicators . . . 21

5.2 Methodology . . . 21

**6 Models** 24

6.1 Persistence . . . 24

6.2 Gradient Boosting Machine . . . 25

6.2.1 Results . . . 26

6.3 Support Vector Machine . . . 27
6.3.1 Data Preprocessing . . . 27
6.3.2 Results . . . 28
6.4 ARIMAX . . . 29
6.4.1 Data Preprocessing . . . 30
6.4.1.A Standardize Data . . . 30
6.4.1.B Principal Components Analysis . . . 30
6.4.2 Stationarity . . . 31
6.4.3 Parameters . . . 34
6.4.4 Residuals . . . 34
6.4.5 Results . . . 36
6.5 GAMAR . . . 37
6.5.1 Residuals . . . 38
6.5.2 Results . . . 40

**7 Ensemble** **42**
7.1 Weighted ensemble . . . 42
7.1.1 Results . . . 43
7.2 Stacking . . . 44
7.2.1 Principal Components Analysis . . . 45
7.2.2 Results . . . 45

**8 Conclusion** **47**
8.1 Discussion of results . . . 47
8.2 Future work . . . 48

**Bibliography** **48**

**A Hyperparameter tunning** **53**

# List of Figures


1.1 Project architecture . . . . . 2

2.1 Typical power curve . . . . . 5

3.1 Representation of the initial domain of 15x15km (left) and a 3x3km downscaled grid (right) . . . . . 7
3.2 Daily/monthly average of power generated . . . . . 9
3.3 Pearson correlation matrix . . . . . 10
3.4 Interaction between power generated, wind speeds at every height. . . . . 11
3.5 Wind rose with power generated at every wind direction at 170m (left) and sea level (right). . . . . 12

4.1 CRISP-DM representation of the feature selection process . . . . . 14
4.2 Feature importance by XGBoost model . . . . . 15
4.3 Partial dependence plot, at a 95% confidence level, for wind speed at 100m (left), and mean sea level pressure (right) . . . . . 16
4.4 Backwards and forwards lag features . . . . . 19

5.1 Representation of a 1-step-ahead forecast . . . . . 22
5.2 Representation of a 2-step-ahead forecast, with extrapolation . . . . . 22
5.3 Model error with and without extrapolation . . . . . 23
5.4 Demonstration of the results of combining the models . . . . . 23

6.1 Persistence error over time . . . . . 25
6.2 XGBoost and LightGBM error over time . . . . . 26
6.3 XGBoost and LightGBM error over time, zoomed in . . . . . 26
6.4 Support Vector Machine error over time . . . . . 28
6.5 Support Vector Machine error over time, zoomed in . . . . . 28
6.6 Scree plot and Cumulative Scree plot . . . . . 30

6.7 Power Generated . . . . . 31
6.8 Autocorrelation plot . . . . . 31
6.9 Partial Autocorrelation plot . . . . . 32
6.10 Differentiated series . . . . . 32
6.11 Autocorrelation plot of the differentiated series . . . . . 33
6.12 Partial Autocorrelation plot of the differentiated series . . . . . 33
6.13 ARIMAX residues representation . . . . . 36
6.14 ARIMAX error over time . . . . . 37
6.15 GAMAR residues representation . . . . . 40
6.16 GAMAR error over time . . . . . 41
6.17 GAMAR error over time, zoomed in . . . . . 41

7.1 Representation of weighted mean ensemble . . . . . 43
7.2 Weighted ensemble error over time . . . . . 43
7.3 Representation of stacking ensemble . . . . . 44
7.4 Scree plot and Cumulative Scree plot for model's predictions . . . . . 45
7.5 Stacking ensemble error over time . . . . . 46

A.1 Cross validation with K=5, where the red shapes represent the training set and the blue shapes represent the validation set. . . . . 53

# List of Tables

3.1 Atmospheric features initially considered . . . . . 8

4.1 Features removed and their impact on the error . . . . . 16

6.1 Identification table for $p$ and $q$. parameters . . . . . 34

A.1 Hyperparameters for each model in a short-term regime . . . . . 54
A.2 Hyperparameters for each model in a medium-term regime . . . . . 55

# Acronyms

**ACF** Autocorrelation Function.

**ANN** Artificial Neural Network.

**AR** Autoregressive.

**ARIMA** Autoregressive Integrated Moving Average.

**ARIMAX** Autoregressive Integrated Moving Average with Exogenous Variables.

**EDA** Exploratory Data Analysis.

**GAM** Generalized Additive Model.

**GAMAR** Generalized Additive Model with an Autoregressive component.

**LightGBM** Light Gradient Boosting Machine.

**MA** Moving Average.

**MM5** Fifth-generation Mesoscale Model.

**NRMSE** Normalised Root Mean Squared Error.

**NWP** Numeric Weather Prediction.

**PACF** Partial Autocorrelation Function.

**PCA** Principal Components Analysis.

**SVM** Support Vector Machine.

**XGBoost** Extreme Gradient Boosting.

# Chapter 1

# Introduction

In this chapter, we explain the main reasons that led to the accomplishment of this work. Furthermore, an initial overview of the proposed solution structure is shown.

## 1.1 Motivation

The worldwide significant increase of renewable energies is not only due to their important environmental advantages, but also due to their advantages in increasing a country's energetic independence while promoting domestic economic growth. Most industrialized countries, including Portugal, have adopted policies to increase the usage of renewable energy in order to comply with international environmental agreements, such as The Paris agreement. In such context, Portugal already has over 100 wind farms all across the continent, which produces over 25% of all the consumed energy [1].

One of the most essential tasks in power system operations and control is short and medium-term forecasting. The short and medium-term forecasting of electric power production at wind farms is essential because it allows power production schedules at conventional power plants to be established and power reserves to be determined. Thus, accurate forecasts of electric power production at wind farms play a vital role in keeping the costs competitive by reducing the need for wind curtailments and thereby increasing revenue in electricity market operations [2]. However, the random and unstable characteristics of the wind make it difficult to forecast the power generated. Hence, extensive efforts have been devoted to the developments and improvements of wind speed and power forecasting by numerous energy and environment related research centers and universities [3].

This work will focus on 12 parks, chosen by an expert, that represents the entire Portuguese

system. However, the main goal is to implement a system able to make short and medium-term for one of the parks, called Park-Number1, so that it is possible to replicate the results in every other park.

## 1.2 Work overview

The thesis starts with an overview of the three state of the art methods: the physical, the statistical, and the combined approaches, shown in Chapter 2.

Subsequently, it is explained the work done, which can be divided into five steps: NWP Data, Feature Engineering, Methodology, Models, and Ensemble, as shown in Figure 1.1. Firstly, in Chapter 3, it is explained the way of obtaining, which are not observations but forecasts, and an initial exploratory analysis is made. Afterward, in Chapter 4, we show all the transformations done to our dataset, in order to improve the model quality. Right after, in Chapter 5, it is explained the methodology used to obtain the forecasts, with the models shown in Chapter 6. Finally, a way to aggregate the various forecasts is discussed, in Chapter 7.

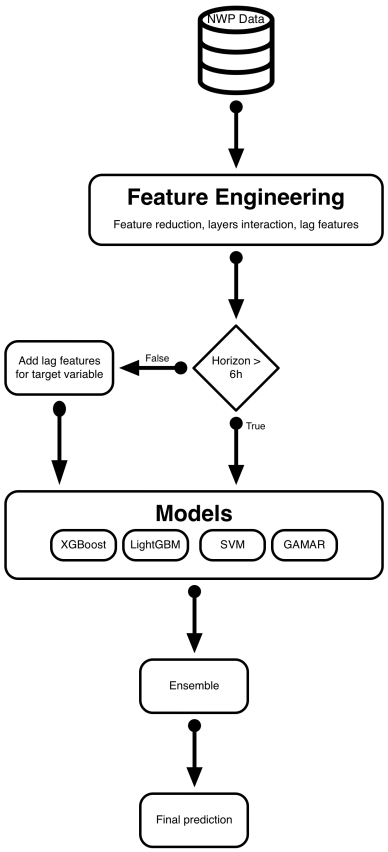


**Figure 1.1:** Project architecture

Lastly, a discussion of the obtained results is made, as well as, the main achievements and limitations of the work architecture are pointed out. It is also suggested possible developments to the solution architecture.

# Chapter 2

# State of the art

Research in the area of forecasting power generated by wind farms has been devoted to the development of effective and reliable tools, and many different approaches have been proposed. These techniques are often based on wind speed forecasts from NWP models, just like the MM5 model, and, eventually, previous measurements are used. The two mainstream approaches are the physical or power curve approach and the statistical approach. In some cases, a combined approach is used, in an attempt, to integrate the advantages of both approaches.

## 2.1 Physical or Power curve approach

The physical approach takes advantage only of the wind speed prediction of NWP models. The wind speed is transformed into power generated based on each type of aerogenerator power curve, represented in Figure 2.1. The power curve of each aerogenerator is a function that is obtained in idealized conditions and relates wind speed to the expected power generated. In the aerogenerator power curve, there are three key points on the velocity scale [4]:

- Cut-in speed: the minimum wind speed at which the machine will deliver useful power,
- Rated wind speed: the wind speed at which the rated power (generally the maximum power output of the electrical generator) is reached.
- Cut-out speed: the maximum wind speed at which the turbine is allowed to deliver power (usually limited by engineering design and safety constraints).

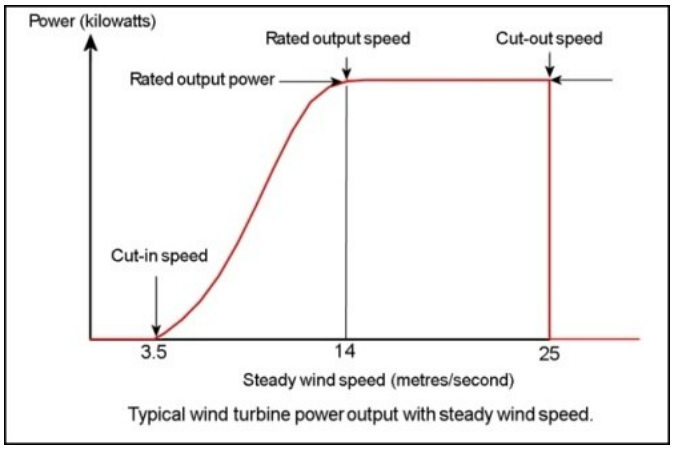


**Figure 2.1:** Typical power curve

This approach tends to yield good results in medium to long-term wind power prediction. However, the relationship between wind speed observed at a location in the farm and the aggregate wind generation from the farm is far more complicated than a simple transformation based on the turbine power curve. This is because, the power curve does not take into consideration a multitude of factors, just like the wind speed and direction variations, turbulence, air temperature, and density that affect the aerogenerator power generation.

Also, because the wind is not spatially homogeneous, and each aerogenerator is sited in different topographic and land conditions, make it so that each aerogenerator works at their rate, which is not considered when combining the power outputs of the power curve.

Therefore, the applicability of the prior efforts is rather limited, particularly when the farm has a large number of turbines distributed over an extended geographical area.

## 2.2 Statistical approach

The alternative approach for wind power forecasting is based on purely statistical modeling. Statistical models in their pure form try to establish a relation between a set of explanatory variables, and wind power measures. The statistical models can be divided into two types: the time series based models and the non time series based models.

Time series based models are models that try to establish a relationship between previous observations and the next value. For this, an autoregressive component is commonly used. Several types of time-series models may be considered, including autoregressive model (AR), moving average model (MA), autoregressive moving average model (ARMA), and autoregressive integrated moving average model (ARIMA). Pure time series models exhibit good

performance for small horizons, this is, up to 6-10 hours ahead [5].

On the other hand, non time series based models don't require previously observed data to make predictions but use the NWP forecasts and further meteorological variables to establish a relationship with the power generated. However, it is common to also use the previously observed data in the model. Several types of models are used, but the more common ones are Artificial Neural Network (ANN) and Support Vector Machines (SVM) [6].

## 2.3 Combination approach

The basic idea of the combination model is to combine different strategies or models, retaining the advantages of each approach. The desire is to improve the forecast accuracy, but combining forecasts does not always perform better than the best individual forecasts. However, in some cases, it is viewed as less risky to combine forecasts than to select just an individual forecast [7].

The most widely accepted procedure for the model combination is to assign a weighting coefficient to each method, such that the sum of weights is equal to one. A common weighting strategy is to give a higher weight for statistical models when the prediction horizon is small, and higher weight for models with the physical approach when for medium-term predictions.

# Chapter 3

# NWP Data

## 3.1 NWP input data

The data used for this project came from Numeric Weather Prediction model forecasts for the Portuguese Park-Number1, the MM5 model. The MM5 model, short for Fifth-generation Mesoscale Model, is a mesoscale model that has the ability to describe the behavior and evolution of air masses and treat explicitly the inherent phenomena of atmospheric turbulence as well as other types of nonlinear atmospheric phenomena. For these predictions, the model has into consideration the roughness of the terrain as well as previous weather data. The model starts by calculating the values to a 15x15km grid, and afterward, the results are downscaled to a 3x3km grid, as shown in Figure 3.1.

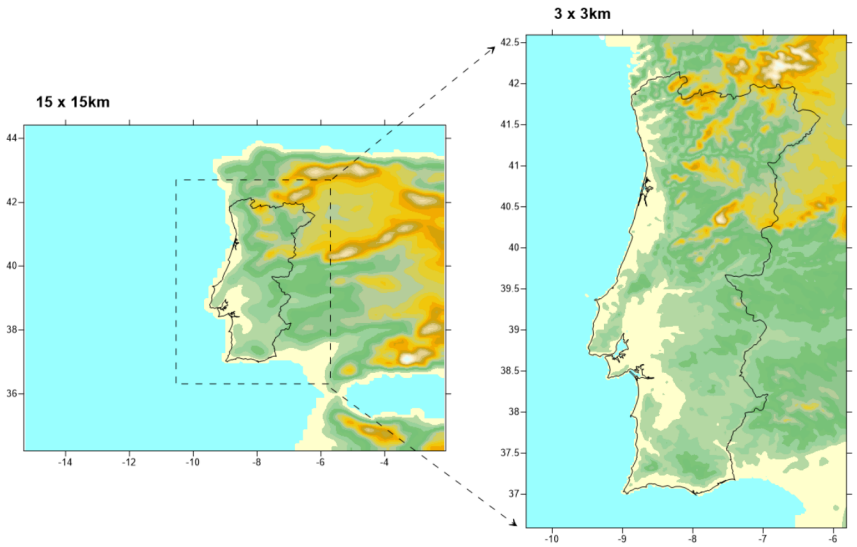


**Figure 3.1:** Representation of the initial domain of 15x15km (left) and a 3x3km downscaled grid (right)

After the downscaling to a 3x3km grid, spatial interpolation is done to get the values for the exact coordinates of the eolic park.

The model predictions are made on a daily basis starting at 00:00 UTC, with a time horizon of 76 hours, for the years 2018 and 2019. The sample frequency corresponds to 15 minutes.

Lastly, these predictions result in a data set with the following 30 atmospheric parameters that are used to predict the power generated at a given time:

| Name | Abbreviation | Units |
|---|---|---|
| Wind Speed 170m height | ws170m | $m/s$ |
| Wind Speed 100m height | ws100m | $m/s$ |
| Wind Speed 30m height | ws30m | $m/s$ |
| Wind Speed 10m height | ws10m | $m/s$ |
| Wind Speed Sea Surface height | wssea | $m/s$ |
| Direction 170m height | d170m | º |
| Direction 100m height | d100m | º |
| Direction 30m height | d130m | º |
| Direction 10m height | d10m | º |
| Direction Sea Surface height | dsea | º |
| Temperature 170m height | t170m | $ºC$ |
| Temperature 100m height | t100m | $ºC$ |
| Temperature 30m height | t30m | $ºC$ |
| Temperature 10m height | t10m | $ºC$ |
| Temperature 2m height | t2m | $ºC$ |
| Relative Humidity 2m height | rh2m | $\%$ |
| Momentum Flux | momf | $kg/(ms^2)$ |
| Surface Pressure | psurf | $hpa$ |
| Mean Sea Level Pressure | mslp | $hpa$ |
| Cloud Fraction Total | clf | 0 to 1 - 0 no clouds; 1 full cloudy |
| Low Cloud Fraction | clo | 0 to 1 - 0 no low clouds; 1 full low cloudy |
| Middle Cloud Fraction | cmi | 0 to 1 - 0 no middle clouds; 1 full middle cloudy |
| High Cloud Fraction | chi | 0 to 1 - 0 no high clouds; 1 full high cloudy |
| Surface Heat Flux | shf | $W/m^2$ |
| Latent Heat Flux | lhf | $W/m^2$ |
| Incoming Short Wave Radiation | swr | $W/m^2$ |
| Total Precipitation | raint | $mm$ |
| Convective Precipitation | rainc | $mm$ |
| Non Convective Precipitation | rainnc | $mm$ |

**Table 3.1:** Atmospheric features initially considered

## 3.2 Exploratory Data Analysis

Exploratory data analysis (EDA) is used to analyze and investigate datasets and summarize their main characteristics, often employing data visualization methods. EDA helps to discover patterns, spot anomalies, or any other problem so that it can be solved later.

We will start the analysis by representing the daily and monthly average of the wind power generated over both years:

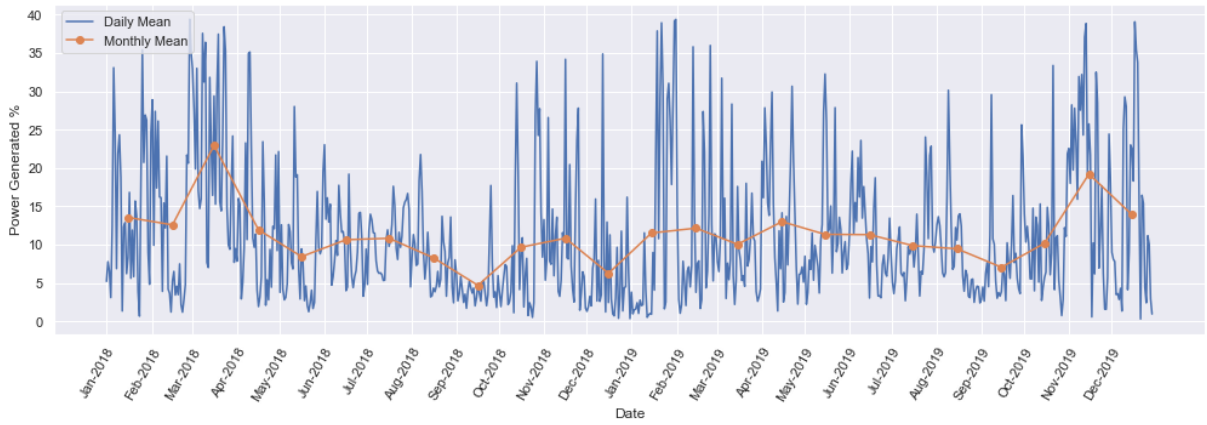


**Figure 3.2:** Daily/monthly average of power generated

As it is possible to see, the data seems to have an annual seasonality, where January, February, March, November, and December are the ones where more wind power is generated, whereas, in May, June, July, August, and September, the opposite happens. Thus, we conclude that the 2018 data appears to be representative of the 2019 data, with the exception of March of 2018, which was an atypical month, in the sense that the wind power production was much higher than the average.

After identifying some insights about wind energy production over time, we can now investigate the features available in our data set. Since we have many attributes, then it is not practical to be representing all of them, and we will only represent the correlation matrix. The correlation matrix contains Pearson's correlation coefficient for all pairs of variables. This coefficient measures the linear relationship between each pair of variables, and its value ranges from -1 to 1, where -1 shows a perfect negative correlation, while a correlation of 1.0 represents a perfect positive correlation. If the absolute value of the coefficient is greater than 0.7 then the variables have a strong linear correlation.

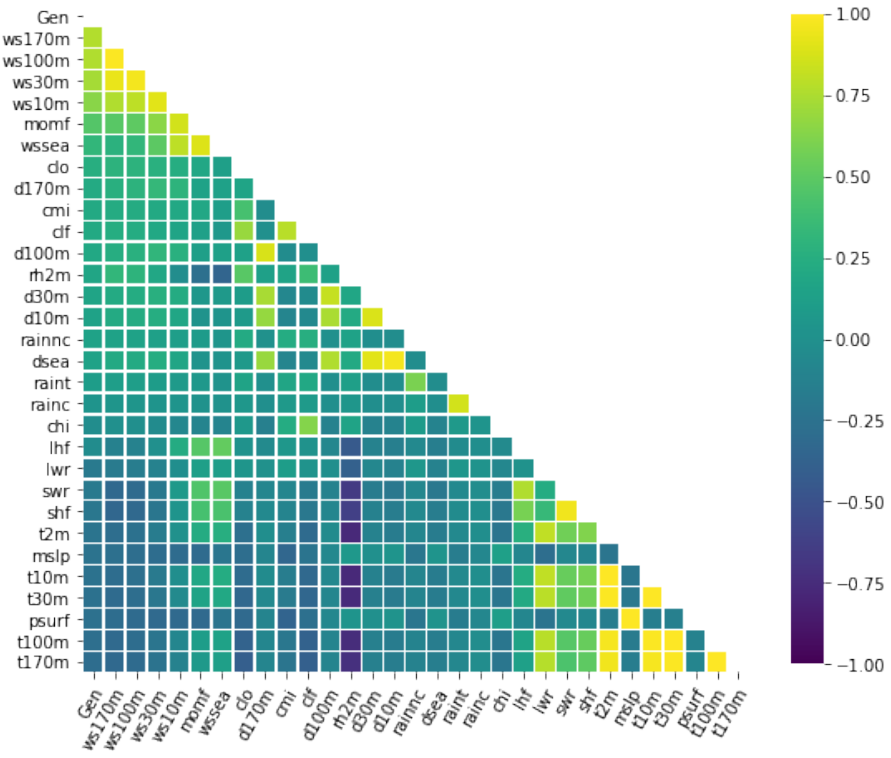


**Figure 3.3:** Pearson correlation matrix

By analyzing Figure 3.3, more specifically the first column, it is possible to conclude that the power generated has a higher linear correlation with the wind speeds at all the different altitudes than with any other variable. However, the power generated is only strongly correlated with the wind speed at 170, 100, and 30 meters, with a correlation coefficient of 0.77, 0.76, 0.73, respectively.

In this same figure, it is also possible to detect a potential problem, in which some of the explanatory attributes have a very high correlation. This may indicate that the two features explain the same information, so it would be advantageous to only keep one of them. Some examples of these attributes are the temperature at 10 meters height with the temperature at 2 meters height that have a correlation coefficient of 0.996 or the wind speed at 170 meters height and at 100 meters height that have a correlation coefficient of 0.992. Figure 3.4 shows a different way of reaching the same conclusions, that is, that the relationship between the wind speed at 170 meters high and 100 meters high is linearly strong.

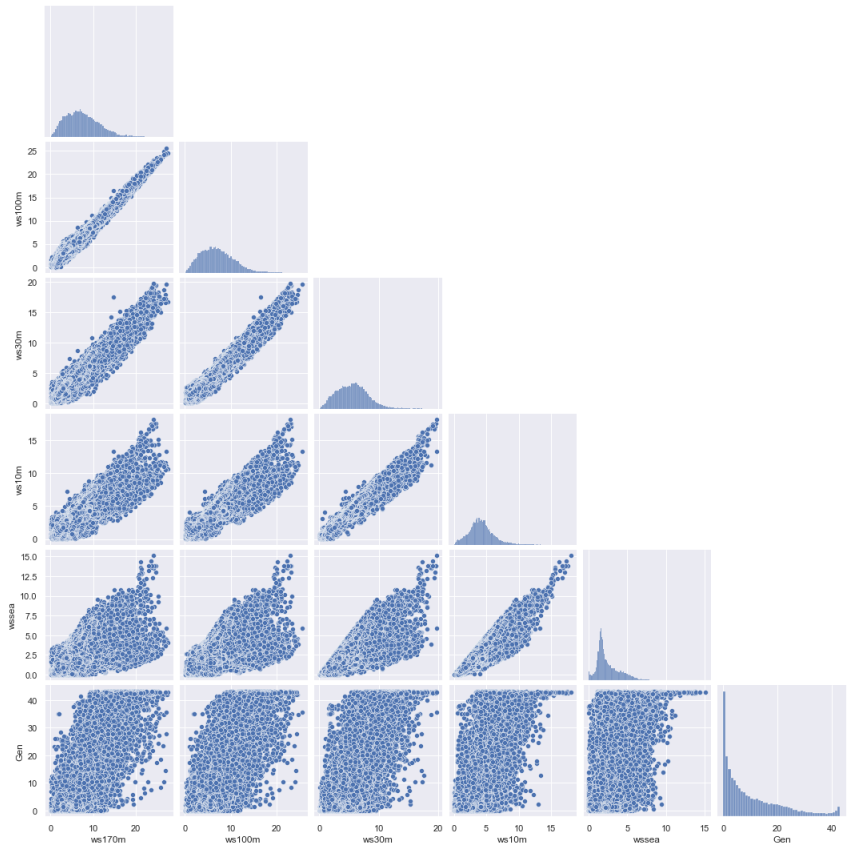


**Figure 3.4:** Interaction between power generated, wind speeds at every height.

Also, on the bottom line of Figure 3.4, it is possible to see that the relationship between wind speed and expected power generated is not as smooth as the Power curve approach indicates, shown in Chapter 2.1.

Even if an attribute and the target variable have a low correlation coefficient, this does not imply that the attribute should not be considered. A low correlation coefficient only means that the relationship between the variables is not linear, an example of this is the wind direction. The correlation between the wind direction and the energy produced is between 0.254 and 0.165, depending on the height. However, when observing the wind rose graph between the direction of the wind and the energy produced, Figure 3.5, it is possible to conclude that this feature manages to contribute to the explanation of the power generated. This is, if the wind comes from the northwest, then there is a greater chance that the wind farm will produce a greater amount of energy.

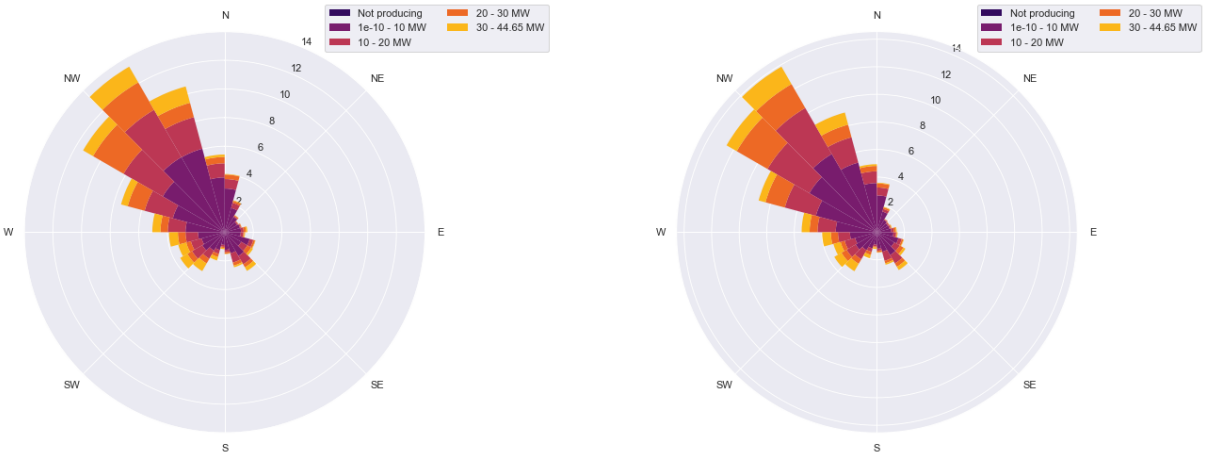


**Figure 3.5:** Wind rose with power generated at every wind direction at 170m (left) and sea level (right).

# Chapter 4

# Feature Engineering

Feature engineering is a process of transforming already existing features into new ones in order to have a better representation of the problem and resulting in a decrease of the model error. This step is considered the most important step in every machine learning problem, to the point of professor Andrew Y. Ng saying: ”Applied machine learning is basically feature engineering.” [8]. With this in mind, this section will contain the feature engineering done to our dataset, from removing unnecessary features to adding new ones that represent the interaction between altitudes and, lastly, adding features to deal with the autocorrelation problem.

## 4.1 Feature Selection

Feature selection refers to techniques that help us select the best feature to use, in order to reduce the number of input variables in a dataset. More input features often make a predictive modeling task more challenging, so our goal is to remove features that don’t contribute any information to our model or even harm the model performance. This can be a very iterative process since it is necessary to train the models, find out which variables contribute least to the forecast in order to try to remove them, make sure that the error does not increase, and repeat the process, as represented by Figure 4.1.

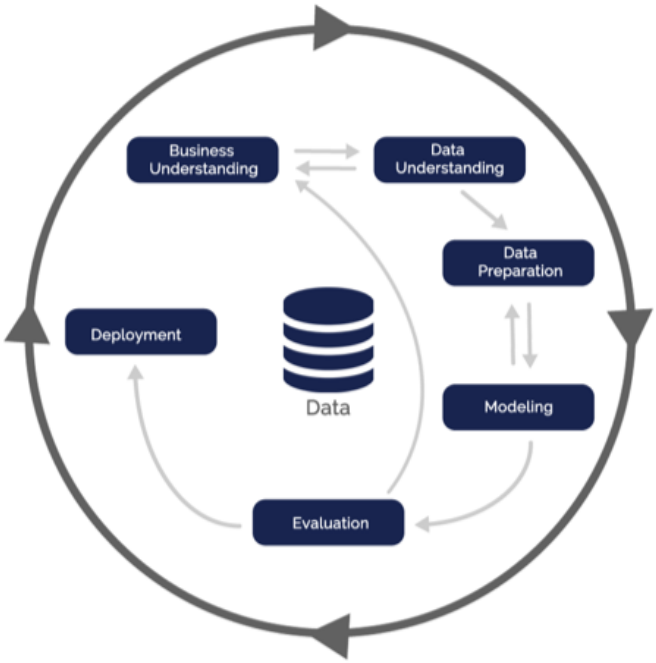


**Figure 4.1:** CRISP-DM representation of the feature selection process

The most difficult step of this process is to detect which variable contributes the least to the forecast. And so, two different techniques are used, based on two different models, the XGBoost and GAM, explained in chapters 6.2 and 6.5, respectively.

Starting with XGBoost, this model is based on the combination of several decision trees into one predictor. So it is possible to obtain a rank of each feature with the number of times the feature was considered by the model, as shown in Figure 4.2.

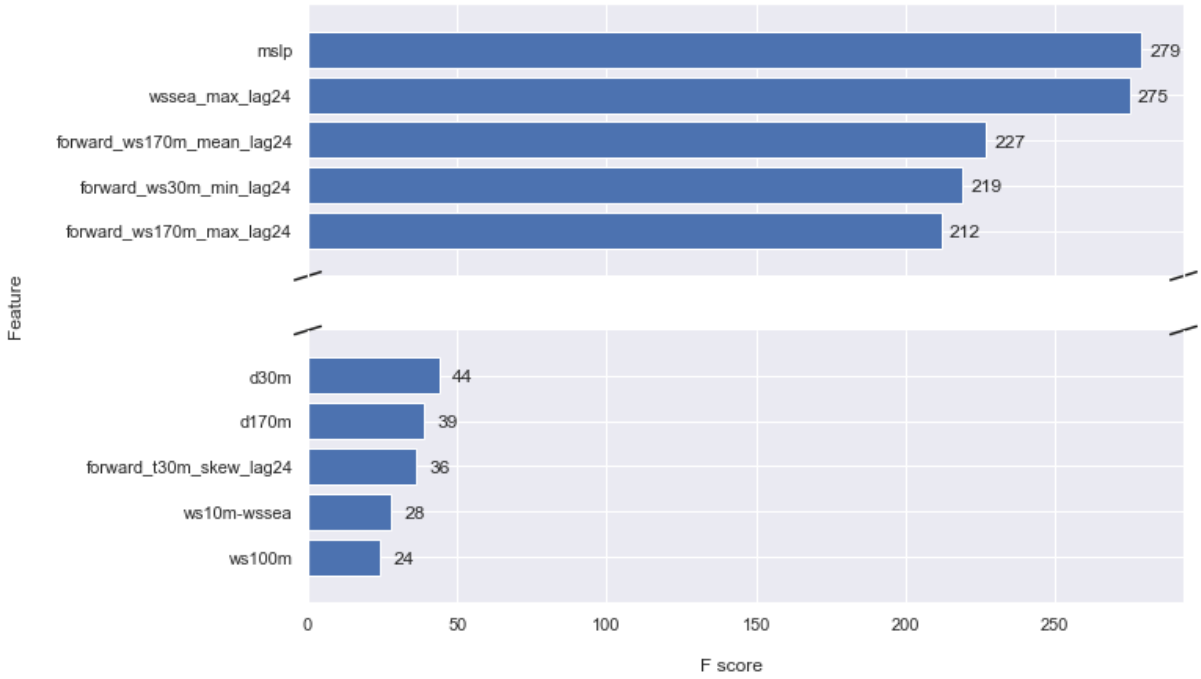


**Figure 4.2:** Feature importance by XGBoost model

In this iteration, the most important feature was the ”mslp”, which stands for mean sea level pressure, because it was used the most amount of times, 279 times out of 440 estimators. On the other hand, the least important attribute is the ”ws100m”, which stands for wind speed at 100 meters, since it was used the least amount of times, only 24 out of 440 estimators. Thus, the attribute to try to remove is the least used one, and in this iteration of the process, it is the wind speed at 100 meters.

A second technique is used, to confirm if the results obtained by the previous technique hold true for different types of models. This second technique is based on a different model, it is based on the generalized additive model (GAM). GAM's are an extension of general linear models, then to find out if a variable is significant we can analyze the partial dependence plot of each attribute. In Figure 4.3, it is represented the partial dependence plot, with a 95% confidence interval, for the least important attribute, the wind speed at 100 meters, and the most relevant attribute, the mean sea level pressure, according to XGBoost.

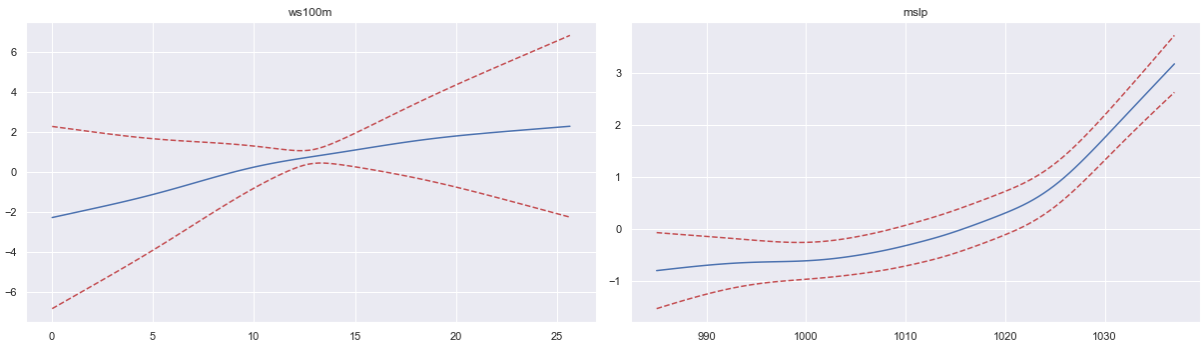


**Figure 4.3:** Partial dependence plot, at a 95% confidence level, for wind speed at 100m (left), and mean sea level pressure (right)

As it is possible to observe in the partial dependence plot for the mean sea level pressure, the bands do not contain zero in the majority of the interval, indicating that the variable is significant. Unlike the wind speed at 100 meters, as its bands always contain zero, indicating that the wind speed at 100 meters is not significant at a 95% confidence level.

So, since both techniques indicate that the wind speed at 100 meters is not significant, we conclude that the wind speed at 100 meters should be the next feature removed.

Despite both models agreeing on the feature to remove, it is still important to analyze the effect of removing said feature on the error. If the error increases with the removal of the attribute, then it is not worth removing the variable and it should be maintained in the dataset, despite what the models say.

From repeating this process multiple times, it was possible to conclude that several variables presented in our dataset do not contribute to prediction, as initially speculated in the exploratory data analysis. These variables were:

| Feature removed | Mean error reduction |
|---|---|
| Features that represent precipitation | 0.057 |
| Features that represent cloud coverage | 0.036 |
| Wave radiation | 0.020 |
| Heat flux | 0.022 |
| Surface pressure | 0.042 |
| Wind speed, direction and temperature at 100 meters | 0.077 |
| Wind speed, direction and temperature at 10 meters | 0.025 |

**Table 4.1:** Features removed and their impact on the error

The results obtained by the models seem to make sense, since precipitation, cloud coverage, wave radiation, and heat flux are all fundamental variables for forecasting solar energy and not wind energy, hence their removal. On the other hand, the information at 100 and 10 meters is important to the problem, and it is highly correlated with the power generated. However, this information is also strongly correlated with the information at 170 meters and sea level, respectively, as shown in the exploratory data analysis, Chapter 3.2, making them unnecessary to the models.

In total, just by removing these attributes, our models improved, on average, 0.279% on the error.

We also tried to add new attributes to the database, but the inclusion of these did not cause the error to decrease, so they were not added. These attributes were:

- Air density at every level
- Wind speed change at every level
- Wind direction change at every level
- Temperature change at every level

## 4.2 Layer's Interactions

The interaction between consecutive altitudes may be helpful to correct any outlier that exists at a specific altitude, and so to model the turbulence. Thus, the layer's interaction will be calculated for the wind speed, wind direction, and temperature at every consecutive pair of altitudes.

To calculate the wind speed interaction between consecutive layers ($\psi_{i,j}$), we used an approximation of the Shear velocity. Shear velocity, also called friction velocity, is a form by which a shear stress may be re-written in units of velocity. It is useful as a method in fluid mechanics to compare true velocities [9]. The approximation considered is given by:

$$\psi_{i,j} = \sqrt{\frac{|\text{WS}_i - \text{WS}_j|}{\rho} * dist_{i,j}}, \tag{4.1}$$

where $\text{WS}_i$ is the wind speed at the altitude $i$, $\rho$ is the air density, and $dist_{i,j}$ is the distance between altitudes.

For the wind direction, the interaction between consecutive layers, $\Phi_{i,j}$, is given by the difference between the two values, bearing in mind that these attributes are not linear but

circular, and so an extra step is necessary, this is:

$$\phi_{i,j} = |Ang_i - Ang_j| \tag{4.2}$$

$$\Phi_{i,j} = \begin{cases} 360 - \phi_{i,j} \quad if \quad \phi_{i,j} \geqslant 180 \\ \\ \phi_{i,j} \quad if \quad \phi_{i,j} < 180 \end{cases}, \tag{4.3}$$

where $Ang_i$ represents the wind direction at altitude $i$, in degrees.

Lastly, the interaction between the temperature of consecutive layers, $\tau_{i,j}$, is given by the difference between both, this is:

$$\tau_{i,j} = T_i - T_j \tag{4.4}$$

where $T_i$ represents the temperature at the altitude $i$.

## 4.3 Lag Features

One of the great difficulties in any longitudinal problem is the high autocorrelation of the variables. This is a crucial problem to solve as it is normal that past observations have a great influence on the future, therefore, they should be considered in any form in the prediction.

Some models, such as ARIMA and variants, have built-in methodologies to solve this problem, usually through an autoregressive component. However, many other models, such as boosting models and Support Vector Machines, are not prepared for such a problem. Hence we will try to add new attributes to deal with the problem so that we can use any model that is not prepared for this type of situation.

A common strategy to solve this problem is to try to replicate an autoregressive component. However, instead of taking into account all past values, as is usually done by an autoregressive component, we will consider a window with a statistical summary of the interval. This type of feature is usually called lag features. However, since our database contains predictions from the NWP model, then we can also create lag features based on the weather forecasts for better results, as is represented in Figure 4.4.

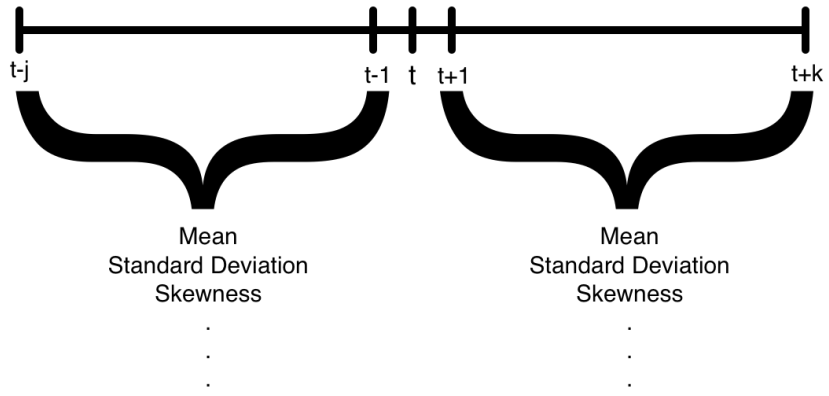


**Figure 4.4:** Backwards and forwards lag features

For this problem, we came to the conclusion that the wind speed at 170 meters, 30 meters, and sea level, the temperature at 170 meters, and 30 meters should have lag features. The statistical methods used are the methods normally used to describe any observation, this is the mean, the standard deviation, the maximum and minimum value, the skewness, and the robust coefficient of variation. The size of the window, that gave us the best results was 24 observations for the backward and forward lag features, this is a window of 6 hours.

It was also created lag features for the direction features, this is for the wind direction at 170 meters and wind direction at 30 meters. The statistical methods used to describe the interval were the same as the wind speed and temperature with exception of the coefficient of variation being replaced by the phase of the angular momentum [10]. That is, the full list of the statistical methods considered is: the mean, the standard deviation, the maximum and minimum value, the skewness, and the phase of the angular momentum. It is important to note that, when calculating all these attributes, it is necessary to take into account, that we are dealing with circular data and not with linear data [11]. As previously, the size chosen for both windows were of 24 observations, this is windows of 6 hours.

Lastly, we can also create lag features for the power generated. However, unlike the lag features for wind speed, temperature, and wind direction, the size of the window used will be much smaller: it will be of 2 observations (window of half an hour) and a second window of 4 observations (window of an hour). The justification for choosing such small windows is the fact that, as predictions are made, it will be necessary to calculate the windows with the predicted values and very large windows would lead to great uncertainty. Since the size of each window is so small, then it is not feasible to use certain statistical methods, such as standard

deviation and skewness. For this reason, only one statistical method is used in the description of these intervals, which is the mean. Thus, these attributes are a closer representation of a conventional autoregressive component.

It is important to reinforce that for the power generated, we can only create lag features based on previous observations, it is also important to say that these features are only created when the prediction horizon is smaller than 6 hours, as will be explained in Chapter 5.2.

# Chapter 5

# Prediction

This chapter contains a description of the performance indicators used in the comparison of the various models tested. In addition, it also describes the methodology used when making the forecasts, which was already suggested in the previous chapter.

## 5.1 Performance Indicators

To further verify the effectiveness and performance of the prediction method proposed for wind power prediction, a single error analysis criteria is introduced to evaluate the model performance over a three days period. The error analysis criteria considered is the Normalized Root Mean Squared Error (NRMSE), that calculated the following way:

$$\text{NRMSE} = \frac{1}{P_{farm}} \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{5.1}$$

where $y_i$ are the actual values, $\hat{y}_i$ are the predicted values, and $P_{farm}$ is the installed capacity of the wind farm.

## 5.2 Methodology

As it was described in Chapter 4, before making a prediction at a time t there are multiple previous steps required. More precisely, in Chapter 4.3, we referred that it is necessary to

have the four previous target observations to make the power generated lag features. For a one-step-ahead prediction, this seems not to be a problem, as shown in Figure 5.1:

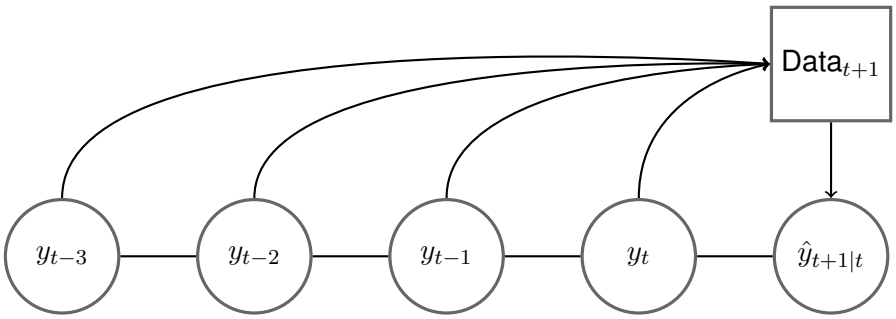


**Figure 5.1:** Representation of a 1-step-ahead forecast

However, it is impossible to have all the observations whenever we want to make n-steps-ahead predictions, where $n > 1$. In this case, instead of using the real observations of the power generated, we will extrapolate the model results and use the model predictions instead, as it is exemplified in Figure 5.2 for a 2-steps-ahead prediction.

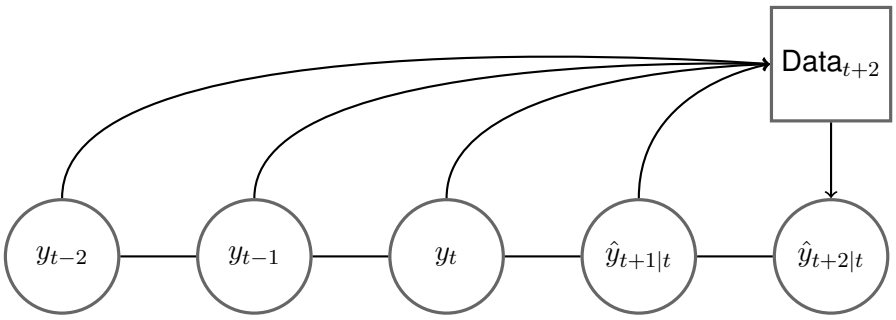


**Figure 5.2:** Representation of a 2-step-ahead forecast, with extrapolation

Even so, this process can cause problems, since the error that the model commits will accumulate as n increases. Thus, in order to conclude whether having the lag features for the power generated is beneficial, we analyzed the error of a model with these attributes and without, as shown in Figure 5.3:

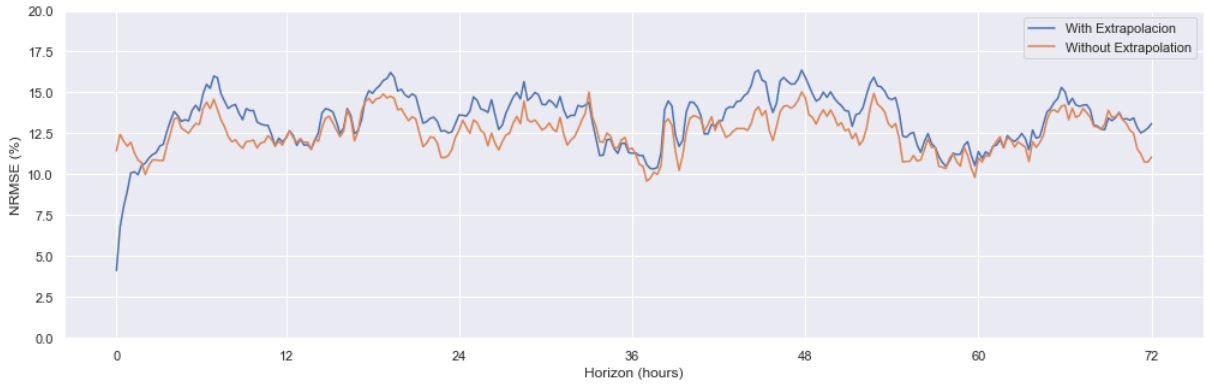


**Figure 5.3:** Model error with and without extrapolation

As it is possible to see, the model that uses the power generated lag features, and by consequence, extrapolates the results to make predictions has better results for short-term predictions, however, the results end up being worse than the model without these features when comparing in a medium-term regime. It is important to notice that this behavior occurred with every algorithm tested, and so we conclude that the usage of lag features for the target variable should only be done if the forecast horizon is smaller than 6 hours.

In summary, every method tested has two versions: one for short term predictions, this is when the prediction horizon smaller than 6 hours, where the model extrapolates the results, and a second version for medium-term predictions when the goal is to make predictions after 6 hours, in which case the model does not use the lag features for the target variable, which is the power generated. As shown in Figure 5.4.

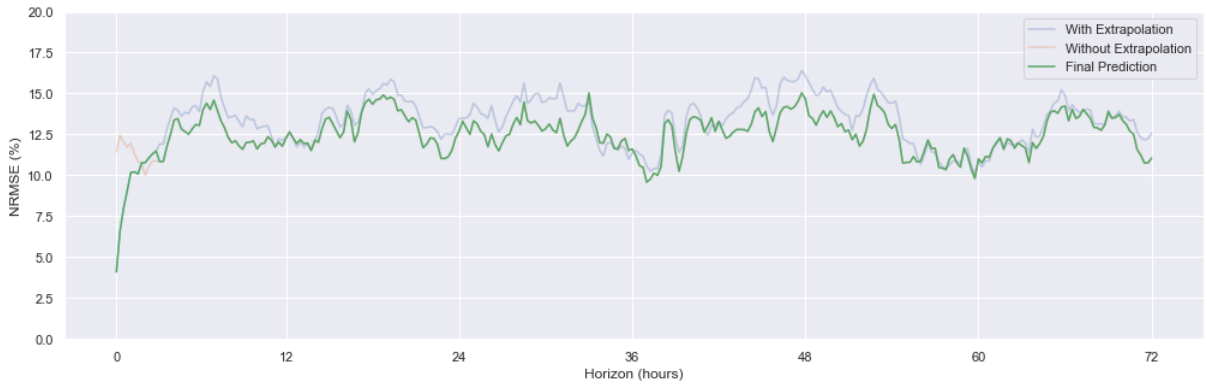


**Figure 5.4:** Demonstration of the results of combining the models

# Chapter 6

# Models

This chapter contains a description of the various algorithms tested during the project. In addition, an analysis of the results of each algorithm is also made. Lastly, it is explained why only four algorithms are considered in the next phase, the ensemble.

## 6.1 Persistence

The persistence method is a well-known method in wind power forecasting for its simplicity. This approach uses the last known value as the forecast for every future point, this is:

$$\hat{y}_{t+k|t} = y_t, \qquad k \geqslant 1 \tag{6.1}$$

The big advantage of this method is its simplicity while giving good results in short-term prediction. On the other hand, this method can not be used for predictions with a horizon longer than 6 hours, as it performs poorly, shown by Figure 6.1:

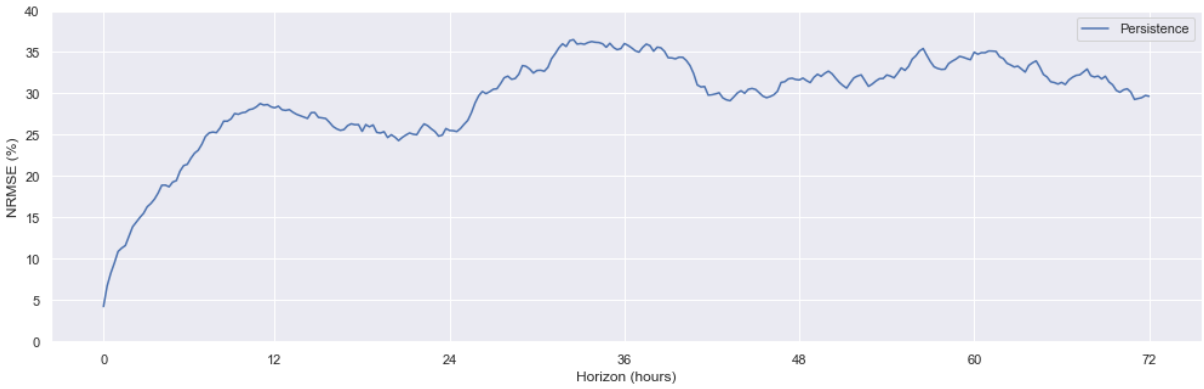


**Figure 6.1:** Persistence error over time

It is commonly said that if a model has worse performance than persistence then it should not be considered in any application.

## 6.2 Gradient Boosting Machine

Gradient boosting machine is a powerful technique for building predictive models. It selects a loss as the objective function and uses the additive model of many weak learners, where each new model in the sequence tries to fix up where the previous one made the biggest mistake while minimizing the loss. The final prediction is a weighted sum of the predictions made by every model:

$$\hat{F}(x) = \sum_{i=1}^{M} \gamma_i h_i(x) \tag{6.2}$$

where $\gamma_i$ is the weight to the $h_i$ model. Usually, the base model is a simple regression three that has very few splits. The parameters of each regression tree are tuned by a gradient descent algorithm, hence the name of the algorithm. For a more detailed illustration of the gradient boosting algorithm, please refer to [12].

Besides the fact that this type of algorithm can learn non-linear relations between the features and the target variable while being interpretable, as seen in Chapter 4.1, the big advantage of this type of algorithm is the fact that it is very robust to highly correlated features, which is essential in this problem. While the only disadvantage of the algorithm is the fact that it can overfit the training data very easily. So a good hyperparameter tunning is important to ensure that this is not happening. The way that the hyperparameters were tunned is shown in the appendix A.

For this project, we used two implementations of this algorithm: Extreme Gradient Boosting (XGBoost) [13] and Light Gradient Boosting Machine (LightGBM) [14], where they only differ in the tree growth procedure.

### 6.2.1 Results

In this section, we will compare the error of the models and the persistence to see if it is worth continuing the usage of this algorithm. According to the following figure:

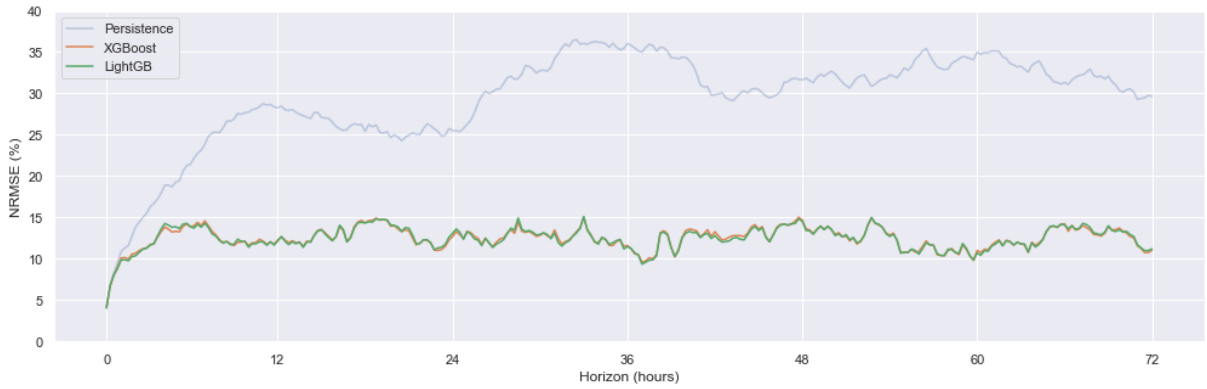


**Figure 6.2:** XGBoost and LightGBM error over time

we can conclude that both models perform significantly better than the persistence in a medium-term prediction, having an average error of 12.603% for XGBoost and 12.565%.

If we zoom in, shown by Figure 6.3, we can see that our models also perform better in short-term predictions, however, the results don't have as big of a discrepancy as they have in the medium-term predictions.

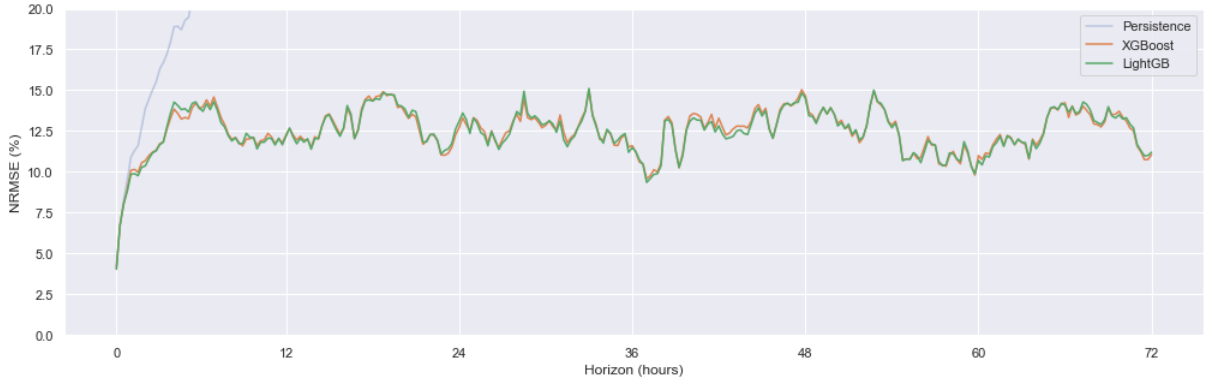


**Figure 6.3:** XGBoost and LightGBM error over time, zoomed in

## 6.3 Support Vector Machine

Support Vector Machine (SVM) is a statistical machine learning model initially developed for binary classification, however, it is now used in every supervised problem. The goal of the support vector machine in regression is to fit as many instances as possible on the decision boundary while limiting margin violations, this is:

$$\min_{\alpha,\alpha^*} \quad \frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i-\alpha_i^*)K(x_i,x_j)(\alpha_j-\alpha_j^*)+\varepsilon\sum_{i=1}^{N}(\alpha_i-\alpha_i^*)-\sum_{i=1}^{N}y_i(\alpha_i-\alpha_i^*), \tag{6.3}$$

$$s.t. \quad \sum_{i=1}^{N}(\alpha_i-\alpha_i^*)=0, \tag{6.4}$$

$$0\leq\alpha_i,\alpha_i^*\leq C, \quad i=1,...,N \tag{6.5}$$

where $\varepsilon$ is a hyperparameter that controls the margin of tolerance, $C$ is a hyperparameter that controls the regularization, and $K(x_i,x_j)$ is the kernel function [15].

SVM's are commonly used, due to their ability to estimate the optimal hyperplane for the data while having the ability to handle high-dimensional data. However, one of the difficulties in applying SVM's is in the choice of the kernel function. The kernel function is an important hyperparameter for the SVM since it models the interaction between the features and the target variable. There are multiple known kernel functions, but for this project, we only used two: linear kernel and RBF kernel. A linear kernel, as the name suggests, models a linear interaction between the features and the target variable, with this we notice that this kernel gave better results for short-term predictions. A linear kernel is given by:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\mathbf{x}_i^T\mathbf{x}_j \tag{6.6}$$

On the other side, for medium-term predictions we got a better result using RBF kernel, also named as Gaussian kernel, that is given by:

$$K(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\gamma||\mathbf{x}_i-\mathbf{x}_j||^2). \tag{6.7}$$

### 6.3.1 Data Preprocessing

SVM assumes that the data it works with is in a standard range. So the normalization of feature vectors prior to feeding them to the SVM is a very important step. We opted to use a standard scalar for every feature, which consists of subtracting the mean and dividing by the standard

deviation. The normalization for the value of order $i$ in the attribute $j$ feature vector is:

$$Z_{ij} = \frac{x_{ij} - \overline{x}_j}{s_{x_j}} \tag{6.8}$$

### 6.3.2 Results

As previously, we will compare the model results with the persistence to conclude if the model should be considered in the ensemble phase. Since we already have the results of XGBoost and LightGBM we will also compare the results with those.

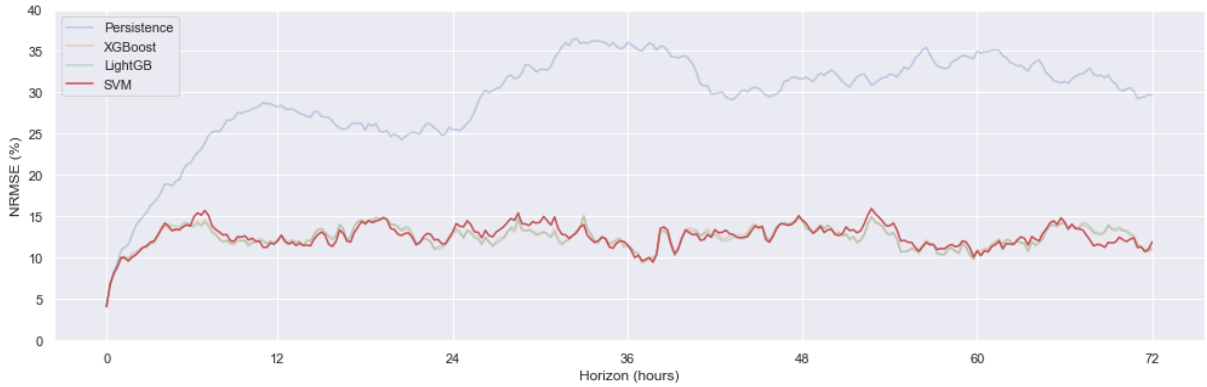


**Figure 6.4:** Support Vector Machine error over time

As we can see, by Figure 6.4, the SVM results can't be distinguished from the persistence in a short-term regime, however, the model outperforms the persistence afterward. The model is able to get an improvement over 20% at some points, just like the boosting algorithms did.

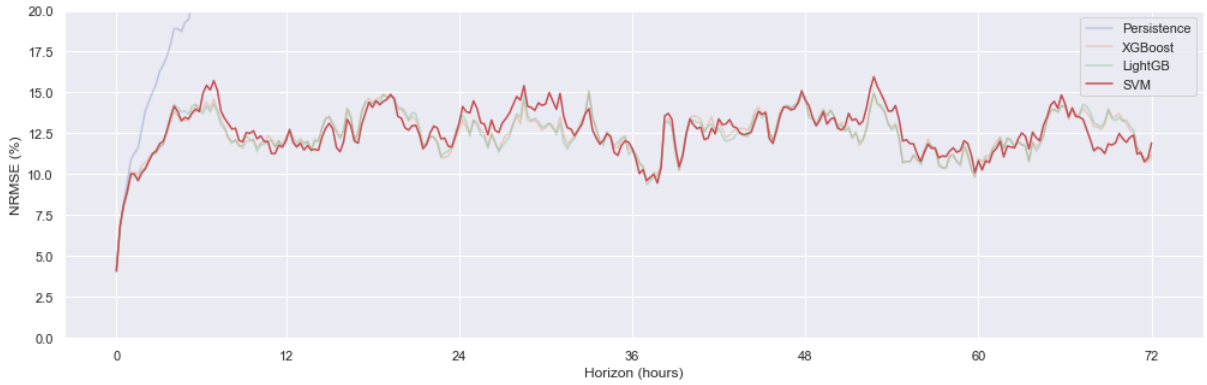


**Figure 6.5:** Support Vector Machine error over time, zoomed in

If we zoom in, as shown in Figure 6.5, it is possible to observe that the SVM competes with the Gradient Boosting Machine algorithm for the best model, giving a good sign for an

ensemble strategy. The average error for the model is 12.726%.

## 6.4 ARIMAX

One of the biggest difficulties in this problem is to deal with the high autocorrelation of the variables, where, previously, to solve this problem, we created the lag features, as explained in Chapter 4.3. However, we will now try a more common approach, the use of a model with an autoregressive component, which is the ARIMAX model.

ARIMAX is an acronym for *AutoRegressive Integrated Moving Average with Exogenous Variables* with is a combination of the standard ARIMA model and linear regression with the exogenous variables.

The ARIMA model can be divided into three parts: the autoregressive process of order $p$, AR($p$), which delineates the relationship of the target variable with the linear combination of its past values.

$$y_t = \phi_1 y_{t-1} + ... + \phi_p y_{t-p}, \tag{6.9}$$

an integrated part, I($d$), to difference the series $d$ times to make it stationary:

$$y_t^{'} = (1 - L)^d y_t, \tag{6.10}$$

and the moving average process of order $q$, MA($q$), where the variable of interest $y_t$ is model by the current and several past errors.

$$y_t = \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q}. \tag{6.11}$$

Combining the three processes we get the ARIMA model of order $(p, d, q)$:

$$(1 - L)^d y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t. \tag{6.12}$$

Finally, by adding a linear regression of the exogenous variables, we get the complete ARIMAX$(p, d, q)$ model:

$$(1 - L)^d y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \sum_{i=1}^{p} \beta_i X_i + \varepsilon_t. \tag{6.13}$$

where $c$ is a constant.

### 6.4.1 Data Preprocessing

One of the assumptions considered when adding a linear component to the model was that the variations are not correlated. However, as it was verified in the exploratory data analysis, Chapter 3.2, and besides all our efforts with the feature selection, Chapter 4.1, this assumption is not being completely fulfilled. To solve this problem a Principal Components Analysis (PCA) is going to be applied to the exogenous variables.

#### 6.4.1.A Standardize Data

Since the features are not all in the same unit of measure, and the variances of the features are of a different magnitude, then it is necessary to standardize the data before applying the principal component analysis. To standardize the data we subtract the mean $\overline{x}_j$ and divide by the standard deviation $s_{x_j}$. The normalization for the value of order $i$ in the attribute $j$ feature vector is:

$$Z_{ij} = \frac{x_{ij} - \overline{x}_j}{s_{x_j}} \tag{6.14}$$

#### 6.4.1.B Principal Components Analysis

After standardizing the data, we need to make the choice of the number of components with which we will carry out the study. To make this decision two parameters will be considered: the contribution of each component, with the Scree Plot, and the cumulative contribution of the components.

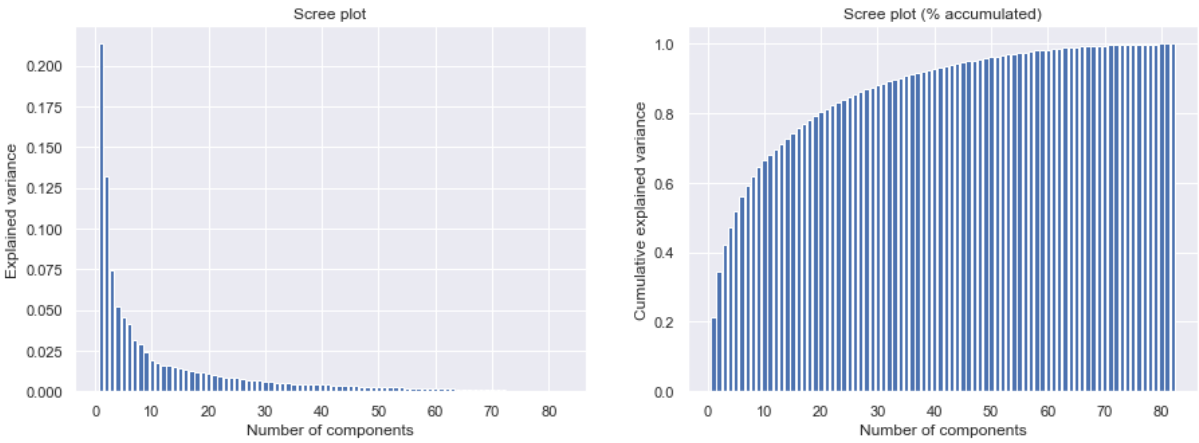


**Figure 6.6:** Scree plot and Cumulative Scree plot

Thus, analyzing the scree plot, the left graph of Figure 6.6, we observed that the bars become close to parallel with the x-axis and so with zero slope from the 30 component. From the cumulative scree plot, the right graph of the same figure, we conclude that 30 components explain more than 80% of the data variance, 88.6% to be precise, which is the minimum required to consider that the PCA is being well applied. With this, we conclude that we must choose only 30 components from the 103.

### 6.4.2 Stationarity

The first stage of ARIMA model building is to identify whether the variable, which is being forecasted, is stationary in time series or not. By stationary we mean, a time series with the properties of not depending on the time in which it is observed, and so having a constant mean and a constant variance.

To identify if the wind power production is stationary we will look at the time series plot but also at the Autocorrelation Plot and Partial Autocorrelation Plot.

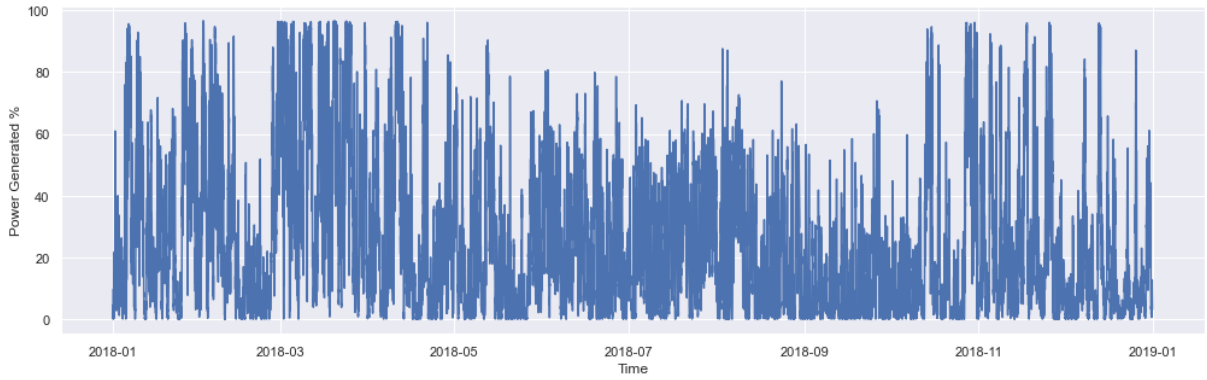


**Figure 6.7:** Power Generated

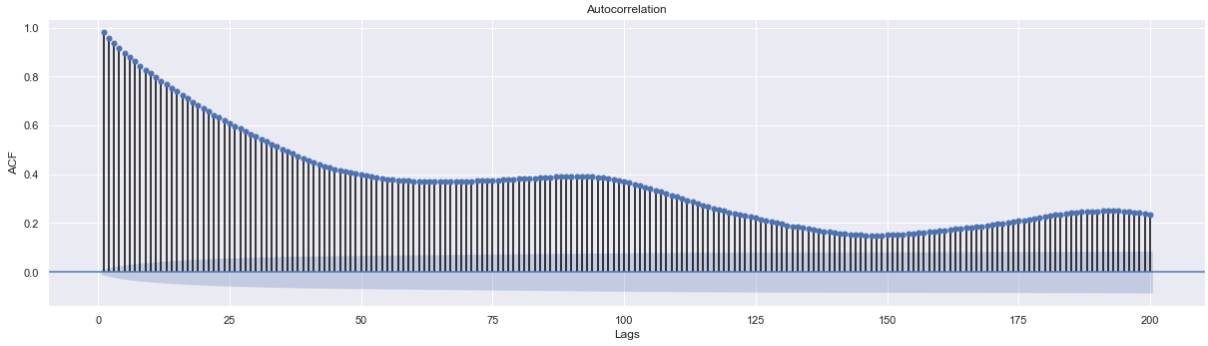


**Figure 6.8:** Autocorrelation plot

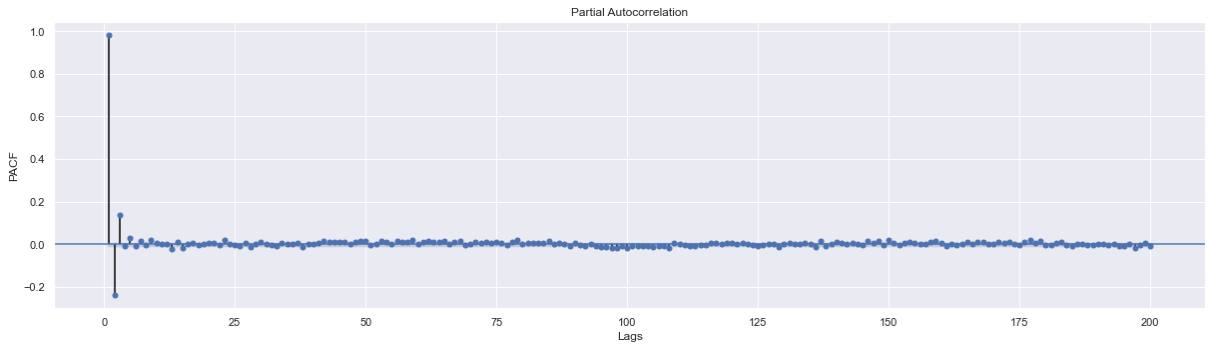


**Figure 6.9:** Partial Autocorrelation plot

As it is possible to see in Figure 6.8, the slow decay of the ACF plot indicates that the series is not stationary, which is confirmed by looking at the time series in itself. In Figure 6.7, is represented the wind power generated, and as it is possible to observe the variance is not constant. For example, the variance of February is much smaller than the variance of March, which seems to be the month with the highest variance.

There are multiple possible solutions to this problem, however, the one that it was decided to apply is differencing. With this, we conclude that the $d$ parameter, represented in Equation 6.10, has to be equal to 1, and the series now looks the following:

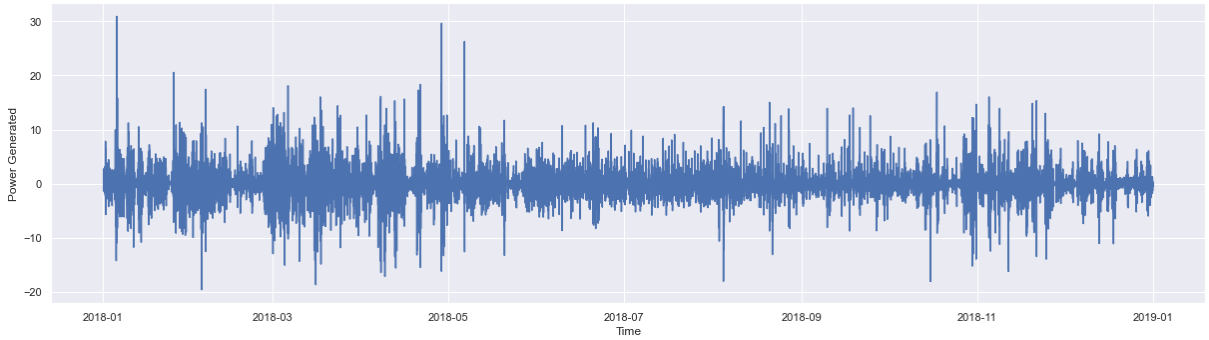


**Figure 6.10:** Differentiated series

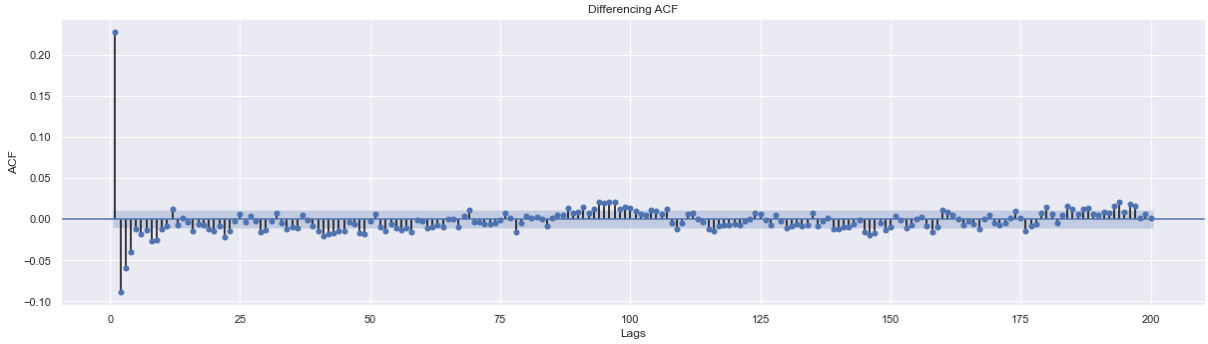


**Figure 6.11:** Autocorrelation plot of the differentiated series

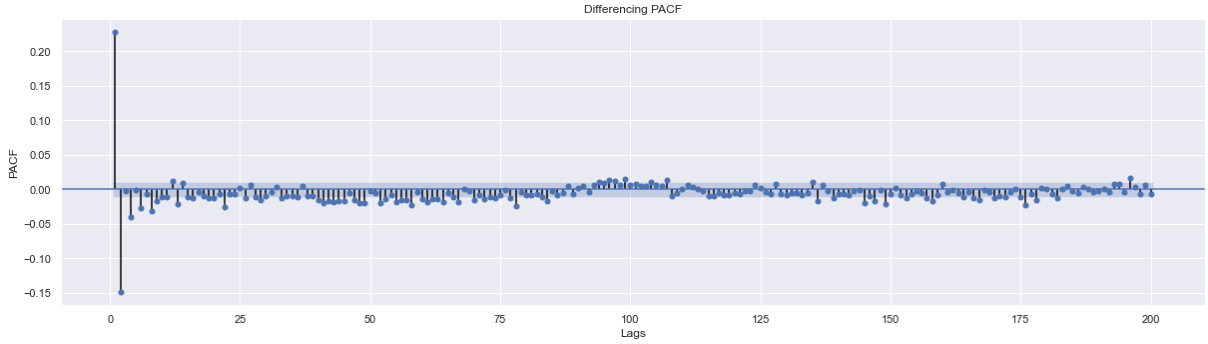


**Figure 6.12:** Partial Autocorrelation plot of the differentiated series

As it is possible to observe by the ACF plot, Figure 6.11, the problem seems to be solved. This is confirmed, by looking at the differentiated series, Figure 6.10, that seems to have a constant mean and constant variance, only with a few outliers.

To confirm that the series is stationary a statistical hypothesis test is going to be applied. The test in consideration is the Dickey-Fuller test which is a unit root test that aims to verify that the series is stationary and has the following hypotheses:

$$H_0 : \text{Series is not stationary} \quad VS \quad H_1 : \text{Series is stationary},$$

with this, our goal is to be able to reject the null hypotheses. Applying the test to our differentiated series we get:

$$\text{p-value} = 5.173 * 10^{-25} \approx 0$$

Since p-value $\approx 0 \leqslant 0.05 = \alpha$ then we reject the null hypotheses, $H_0$, at a significant level of 5%. We also conclude that there is enough evidence that the series is stationary, and it is

possible to continue the ARIMAX modeling.

### 6.4.3 Parameters

Since we already have a stationary series, then we can proceed to choose the most suitable parameters for the autoregressive and moving average components. To identify these parameters we will take into account the ACF plot, the PACF plot, and the following table:

| Process | ACF Plot | PACF Plot |
|---|---|---|
| AR($p$) | Slowly falls to zero | Sharp decline to zero at lag $p+1$ |
| MA($q$) | Sharp decline to zero at lag $q+1$ | Slowly falls to zero |
| ARMA($p, q$) | Slowly falls to zero | Slowly falls to zero |

**Table 6.1:** Identification table for $p$ and $q$. parameters

Looking at the ACF plot, Figure 6.11, it is possible to see the slow decay as the lag increases, while on the PACF plot, Figure 6.12, there is a sharper decline as the lags increase. This indicates that an autoregressive process is necessary, this implies that the $p$ parameter, represented in Equation 6.9, needs to be at least larger than 5. On the other hand, a moving average component does not seem to be necessary, and so the parameter $1$, represented in Equation 6.11, is most likely going to be small, if not equal to 0. With this in mind, it was tested a multiple set of combinations of $p$ and $1$, to see which would minimize the error, as described in the appendix A. It was concluded, that the best value for $p$ would be 15 and the best value for $q$ would be 1, and so we get our final model, an ARIMAX(15, 1, 1).

### 6.4.4 Residuals

Before looking at the model results in our test data, the 2019 data, it is necessary to conclude if the model is well adjusted to the data. To be able to conclude that, it is necessary to make an analysis of the residuals, to see if the model assumptions are not being broken. For the ARIMAX model to be well fitted it is required that the residuals are a pure random process, that is, white noise, in other words, the residual need to be independent and identically distributed with standard normal distribution:

$$\varepsilon_i \sim \text{i.i.d with N(0,1).}$$

The normality of the residues is verified by observing graphs such as the QQ plot and the histogram with the estimated density function. It is possible to see, in the top right corner of the Figure 6.13, that the estimated density of the residuals is very far apart from the density function of a Standard Normal distribution. This is confirmed by the QQ plot, which is represented in the bottom left corner of the same figure, and as it is possible to observe, both ends of the graph are very far apart from the expected quantiles. With this, we conclude that the normality assumption is not being met. However, the normality assumption is the least important assumption when dealing with a large database, just like ours. So, even though this assumption is not being met, we can carry out the study of the residual and check if the residuals are independent and identically distributed.

To verify if the residues are independent and identically distributed, it will be taken into consideration the ACF plot and the Ljung-Box test. Looking firstly at the ACF plot, which is represented in the bottom right of the Figure 6.13, we can see that the lag correlation is very small, indicating that the assumption is being verified. To confirm that everything is correct a statistical hypothesis test is going to be applied. The test in consideration is the Ljung-Box test which tests whether a series of observations over time is random and independent, with the following hypotheses:

$H_0$ : The time series data at each lag are i.i.d., that is, the correlations between the population are zero

$$VS$$

$H_1$ : The time series data are not i.i.d. and possess serial correlation.

with this, our goal is to not have enough evidence to reject the null hypotheses. The test statistic used is the following:

$$Q = n(n+2)\sum_{k=1}^{m}\frac{\hat{r}_k^2}{n-k}, \tag{6.15}$$

where $\hat{r}_k$ is the accumulated sample autocorrelations and m the time lag in consideration. Applying the test to the model residuals we get:

$$\chi_0^2 = 8.252 \qquad \text{p-value} = 0.989$$

Since p-value $= 0.989 > 0.05 = \alpha$ then, there is not enough evidence to reject the null hypotheses, $H_0$, at a significant level of 5%. Hence we conclude that the model residuals are independent and identically distributed, and this assumption is verified.

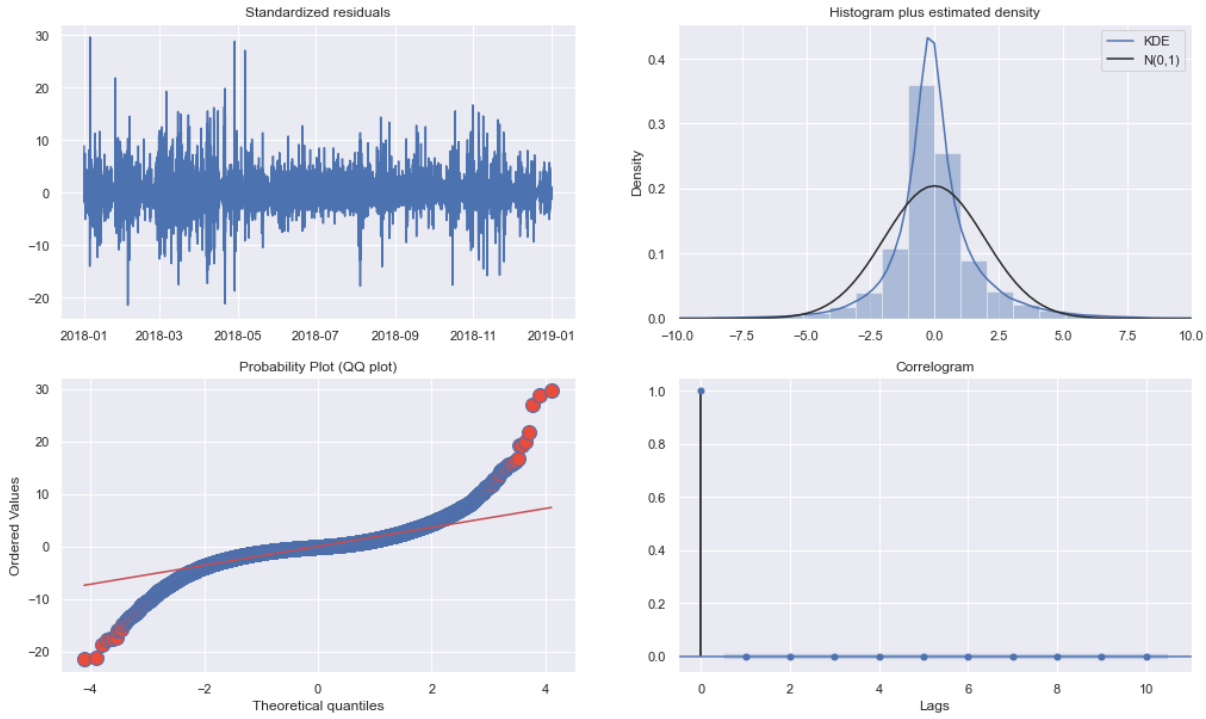


**Figure 6.13:** ARIMAX residues representation

After the residuals analysis, we concluded the residuals do not follow a standard normal distribution, however, they are independent and identically distributed, and as previously mentioned, this is considered the most important assumption to verify. Besides that, we can see that the residuals, represented in the top left corner of Figure 6.13, present a dispersion around the abscissa axis, with zero mean and a variation close to constant, without any trend or seasonality. So, we conclude that the model is well fit to the data and can be used to make predictions.

### 6.4.5 Results

Since the model is well adjusted to the data then we will compare the model results with the persistence and the previous models trained.

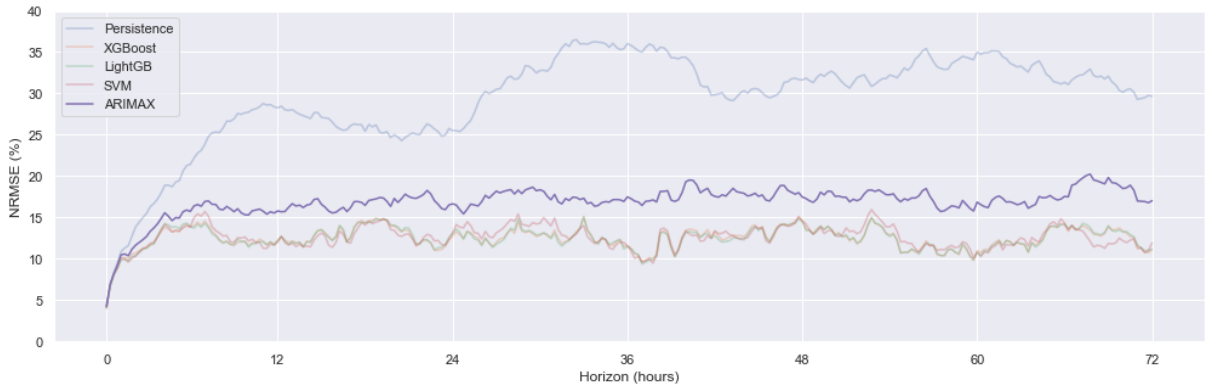


**Figure 6.14:** ARIMAX error over time

As we can see, by Figure 6.14 the ARIMAX results are not the greatest. Even though the ARIMAX outperforms the persistence in almost every time regime, it performs much worse than our previous models, with an average increase, in the NRMSE, of 4.369% over the XGBoost model. For these reasons, the ARIMAX model is not considered in the final architecture of the solution.

One possible explanation of the poor results that the ARIMAX model gave is the fact that this model is using linear regression, this is, it assumes that the relationship between the features and the target variable is linear, which it is not. To try to solve this problem a different model was tested, the GAMAR, explained in the next section.

## 6.5 GAMAR

As was described at the end of the previous section, one possible explanation for the poor ARIMAX results is the fact that this model considers a linear relationship between the features and the target variable, which is not accurate. In the attempt to correct this error a GAMAR model will be considered. The GAMAR model instead of fitting a linear regression, fit's a general additive model (GAM) to the data while keeping an autoregressive part, which was important in the ARIMAX.

General additive models [16] are an extension of general linear (GLM) models that allow a non-linear relationship between the features and the target variable. This is done by replacing the linear predictor used by GLM with a sum of arbitrary functions of each feature. The regression equation becomes:

$$g(E(y_t)) = \beta + f_1(X_1) + f_2(X_2) + ... + f_M(X_M) = \beta + \sum_{i=1}^{M} f_i(X_i) \tag{6.16}$$

where $g(Y)$ denotes the link function that links the expected value to the $M$ predictors variables, and $f_i$ can be any non-linear function, just like polynomials, natural splines, local regressions, etc.

GAM's support a multitude of link functions, that can be used for a regression or classification problem. In this case, since it is a regression problem, it was considered the identity function.

Regarding the non-linear functions, it is common to use a cubic spline as a function to be applied to the variables since it ensures a second derivative to facilitate the optimization problem associated.

Finally, to obtain the GAMAR model, it is only necessary to add an autoregressive component to the GAM model, that is:

$$y_t = \beta + \sum_{i=1}^{M} f_i(X_{ti}) + \sum_{i=1}^{p} \phi_i y_{t-i} \tag{6.17}$$

where $p$ represents the autoregressive order.

### 6.5.1 Residuals

As for the ARIMAX model, before checking the model results for the 2019 data, it is necessary to check if the model assumptions are being verified. The GAMAR assumptions, because of the autoregressive component, are the exact same as the ARIMAX model, this is, the residuals need to be a pure random process. To be a pure random process the residuals need to be independent and identically distributed with standard normal distribution:

$$\varepsilon_i \sim \text{i.i.d with N(0,1).}$$

The normality of the residues is verified by observing graphs such as the QQ plot and the histogram with the estimated density function. It is possible to see, in the top right corner of the Figure 6.15, that the estimated density of the residuals is very far apart from the density function of a standard normal distribution. This is confirmed by the QQ plot, which is represented in the bottom left corner of the same figure, and as it is possible to observe, both ends of the graph are very far apart from the expected quantiles. With this, we conclude that, just like for the ARIMAX model, the normality assumption is not being met. However, as it was referenced when making

the same analysis for the ARIMAX, the normality assumption is the least important assumption when dealing with a large database, just like ours. So, even though this assumption is not being met, we can carry out the study of the residual and check if the residuals are independent and identically distributed.

To verify if the residues are independent and identically distributed it will be taken into consideration the ACF plot and the Ljung-Box test. Looking firstly at the ACF plot, which is represented in the bottom right of the Figure 6.15, we can see that the lag correlation is very small, indicating that the assumption is being verified. To confirm that everything is correct a statistical hypothesis test is going to be applied, the Ljung-Box test. As it was described before, the Ljung-Box test tests whether a series of observations over time is random and independent, with the following hypotheses:

$H_0$ : The time series data at each lag are i.i.d., that is, the correlations between the population are zero

$$VS$$

$H_1$ : The time series data are not i.i.d. and possess serial correlation.

with this, our goal is to not have enough evidence to reject the null hypotheses. The test statistic used is the following:

$$\chi_0^2 = 4.088 \qquad \text{p-value} = 0.999$$

Since p-value $= 0.999 > 0.05 = \alpha$ then, there is not enough evidence to reject the null hypotheses, $H_0$, at a significant level of 5%. Hence we conclude that the model residuals are independent and identically distributed, and this assumption is verified.

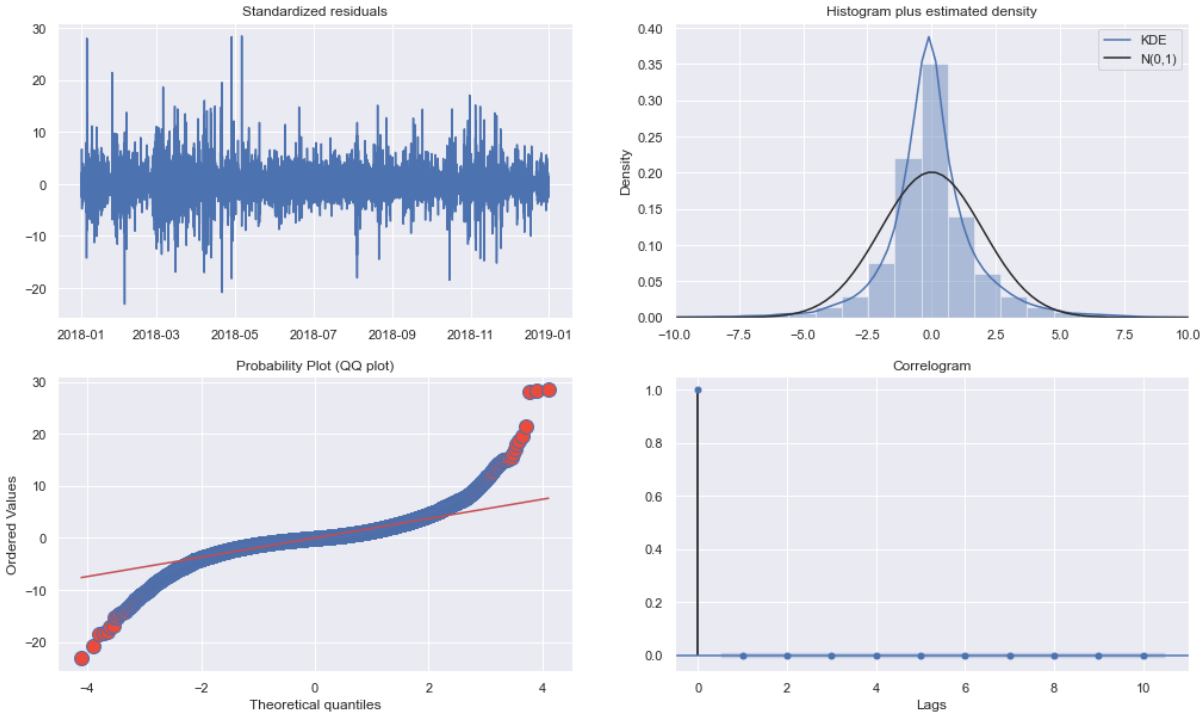


**Figure 6.15:** GAMAR residues representation

After the error analysis, we concluded the residuals do not follow a Standard Normal distribution, however, they are independent and identically distributed. As previously described, the most important assumption is the independence of residuals, which is being verified. Besides that, we can see that the residuals, represented in the top left corner of Figure 6.15, present a dispersion around the abscissa axis, with zero mean and a variation close to constant, without any trend or seasonality. So, we conclude that the model is well fit to the data and can be used to make predictions.

### 6.5.2 Results

As with the other models, GAMAR's results are identical to persistence in short-term forecasts but it is able to obtain much better results in medium-term forecasts, as it is possible to observe in Figure 6.16.

**Figure 6.16:** GAMAR error over time

When comparing the results of GAMAR with ARIMAX, it is possible to see a clear improvement in the results. By this, we confirmed that the problem with the ARIMAX model was the assumption of a linear interaction between the features and the target variable.

If we now compare the GAMAR results with the previous best models, we can conclude that GAMAR can compete with everyone, obtaining great results when the horizon is up to 12 hours, but the results are slightly worse afterward, as it is possible to see in Figure 6.17.

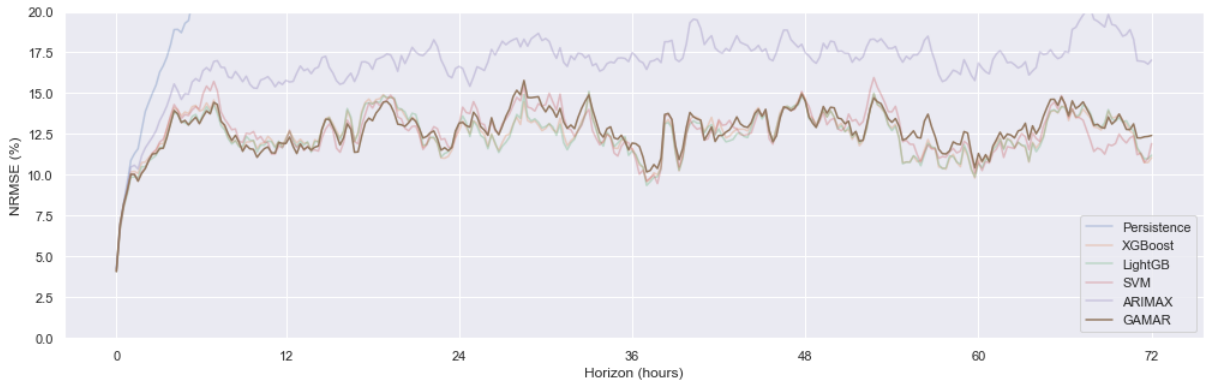


**Figure 6.17:** GAMAR error over time, zoomed in

# Chapter 7

# Ensemble

Ensemble modeling is a process where multiple diverse models are combined to predict an outcome, either by using many different modeling algorithms or using different training data sets. The ensemble model then aggregates the prediction of each base model and results in one final prediction for the unseen data. The motivation for using ensemble models is to reduce the generalization error of the prediction [17].

In this chapter, we will see two different ways of doing model ensemble: the simplest one, which is a weighted mean, and a more sophisticated one, which is stacking. For both techniques, the same four models will be considered: XGBoost, LightGBM, SVM, and GAMAR. As already discussed in chapter 6.4.5, the ARIMAX model will not be used in this process because it obtains significantly worse results than the models previously mentioned.

## 7.1 Weighted ensemble

The first method tested is the simplest because it's just a simple weighted mean. A weight will be associated with each model while ensuring that the sum of all weights is equal to one. Thus, the forecast for this ensemble method is the sum of the product of all weights with the respective model, as shown in Figure 7.1.

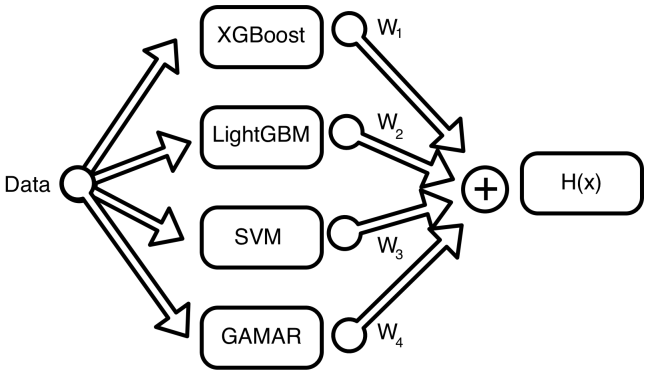


**Figure 7.1:** Representation of weighted mean ensemble

After testing multiple sets of weights, the weights that minimize the error and consequently are the optimal weights are 0.10, 0.35, 0.40, 0.15 for XGBoost, LightGBM, SVM, and GAMAR, respectively.

### 7.1.1 Results

We will then examine the ensemble results, for this, we will compare the error curve obtained by the ensemble method with the curves of various models. It should be noted that, since the four models used in the ensemble already obtain better results than persistence, then it does not make sense to continue to compare the results with persistence. So, when analyzing the error curve, it will only be taken into consideration the four models used in the ensemble. Thus, the results obtained were:

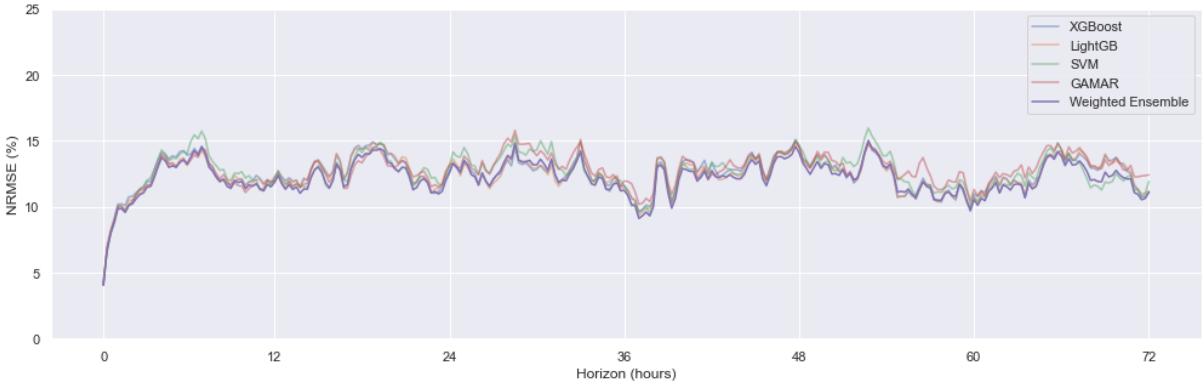


**Figure 7.2:** Weighted ensemble error over time

As it is possible to see in Figure 7.2, the weighted ensemble is able to combine the best results of each model for the different temporal regions, thus minimizing the error. Hence, this method is already better than using only one model, since combining several models also tends to minimize variance. However, this method fails to further reduce the error staying at the same level as the other models.

## 7.2 Stacking

The second ensemble method tested is called stacking (short for stacked generalization). Stacking, instead of using trivial functions (such as the average) to aggregate the predictions of all predictors in an ensemble, uses a different trained model to perform this aggregation [18]. Figure 7.3 is a representation of stacking, where the model that performs this aggregation, called generalizer, can be any. However, it is most common to use a Support Vector Machine or a Regression Tree model.

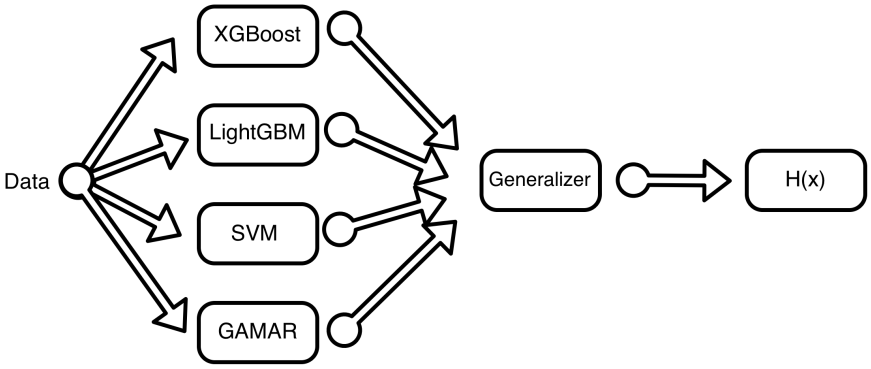


**Figure 7.3:** Representation of stacking ensemble

To the usual stacking ensemble structure, we will add a new stage with a Principal Components Analysis of the model's predictions. This layer is going to be added because the four models considered for the ensemble are based on the same attributes, then it is very likely that the model's forecasts are correlated. In order to decorrelate the forecasts and consequently facilitate the work of the generalizer model, a principal components analysis will be added right before the second layer, the one with the generalizer model.

### 7.2.1 Principal Components Analysis

As it was previously mentioned, in Chapter 6.4.1, when applying a principal components analysis, the data should be normalized, this is, it should have a zero mean and a standard deviation of one. Thus, to the predictions, we subtract the mean and divide by the standard deviation, as shown in Equation 6.14.

To make the choice of the number of components with which we will carry to the generalizer model, two parameters will be considered: the contribution of each component, with the Scree Plot, and the cumulative contribution of the components.

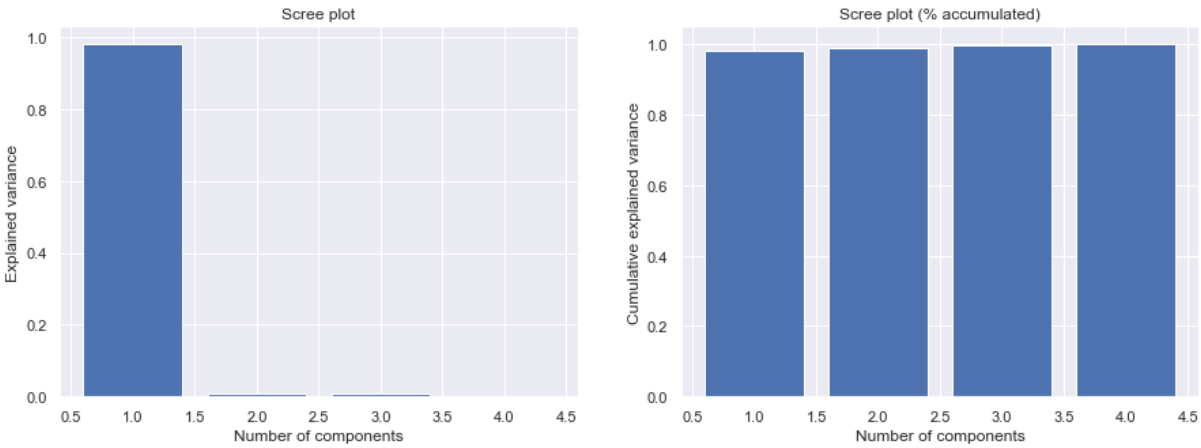


**Figure 7.4:** Scree plot and Cumulative Scree plot for model's predictions

Thus, analyzing the scree plot, the left graph of Figure 7.4, we observe that only one component explains almost all the variance in the model's predictions. The variance explained by this component is 98.1%, which is more than 80%, the minimum required in PCA. With this, we conclude that we must choose only 1 component from the 4.

With this analysis, we also conclude that our assumption, that the model's predictions are correlated, is true.

### 7.2.2 Results

When applying the stacking ensemble methodology, the model considered as generalizer was an SVM. We will now examine the stacking ensemble results, for this, we will compare the error curve obtained by the method with the curves of every individual model and with the previous ensemble technique mentioned.

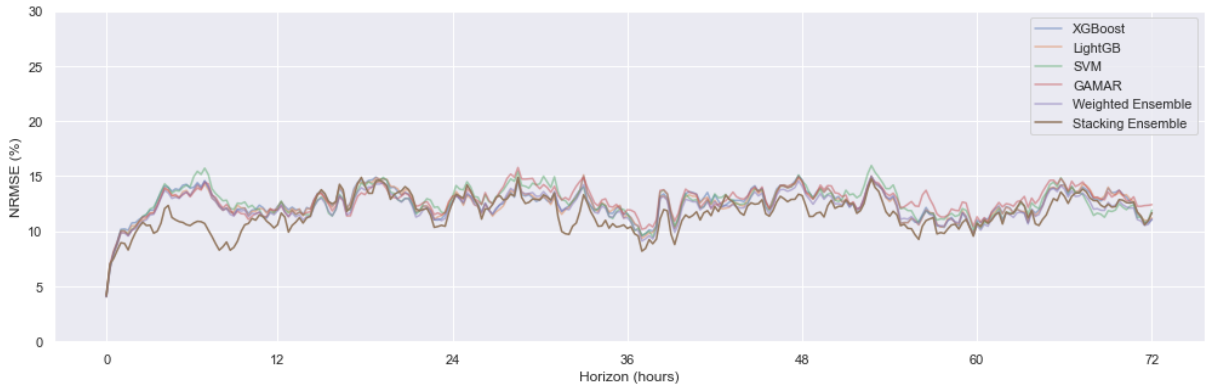


**Figure 7.5:** Stacking ensemble error over time

As it is possible to observe in Figure 7.5, the stacking ensemble strategy is clearly better than any other individual model or the weighted ensemble strategy. This strategy can have a great improvement in a short-time regime results, obtaining improvements up to 3.94%, while keeping up with the previous techniques afterward.

For these reasons, the stacking ensemble is the methodology considered in the final architecture of the project.

# Chapter 8

# Conclusion

With the availability of skillful models and with the increase in the accessibility of computing resources, operational meteorological modeling is increasingly being used as a support tool in areas that need meteorological-derived products to answer their specific needs. It is recognized that better wind power forecasts lead to better decision-making and better electricity market control. Therefore, assessment of the forecast quality is an essential part, bringing confidence to the system being used.

The research presented in this thesis consists of the assessment of various machine learning techniques and models to improve the wind power forecast for the grid utility and better electricity market control.

## 8.1 Discussion of results

Given the objectives proposed for this project in order to analyze the performance of statistical methods, more specifically machine learning models, it is possible to obtain agreement that these were accomplished.

It is possible to conclude that the feature engineering process is fundamental for the success of the models, being the one that requires the most attention. In this process, the addition of the lag feature, to deal with the problem of autocorrelation of variables, is what contributes the most to the reduction of the error.

From feature engineering comes the particularity of dividing the problem into 2 regimes: one for short-term and the other for medium-term. This division was made within a 6-hour horizon in order to minimize the error.

Based on the results obtained, it is possible to conclude that for a model to obtain a good

performance, it needs to be able to capture non-linear relationships between the various attributes and the wind power generated. For this reason, ARIMAX does not manage to obtain as good results as other models tested: XGBoost, LightGBM, SVM, and GAMAR. Thus, only these four models are considered for the ensemble.

Regarding the ensemble, it is possible to conclude that using a model to join the forecasts obtains better results than a simple average. With this strategy, we are able to get an NRMSE of around 10% for the first twelve hours, with the error staying between 10 and 15 percent afterward.

In conclusion, it was possible to show that, with the right preparation and right models, a data-driven approach can provide very good results in a short and medium-term regime for wind power forecast.

## 8.2 Future work

Due to a lack of time, there still is a multitude of ideas that could further improve the architecture of the proposed solution.

For example the usage of a different model, such as a recurrent artificial neural network. The goal of using such a model is to find out if there still is some feature interaction that is not being explained at the moment.

A different possible way of continuing the project is to divide the model based on different wind regimes, such that we would have a different model for slow, medium, and fast winds. With this, we hope to facilitate the model work since the behavior in each regime is very different.

Lastly, we could try to create multiple sets of databases with different features. The goal is to bring diversity into the models and facilitate the ensemble stage to get better results.

# Bibliography


[1] APREN. Balanço da produção de eletricidade de portugal continental (abril 2021), 2021. URL: https://www.apren.pt/pt/energias-renovaveis/producao.

[2] Bart C. Ummels, Madeleine Gibescu, Engbert Pelgrum, Wil L. Kling, and Arno J. Brand. Impacts of wind power on thermal generation unit commitment and dispatch. IEEE Transactions on Energy Conversion, 22(1):44–51, 2007.

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

[4] Rosa Trancoso. Operational modelling as a tool in wind power forecasts and meteorological warnings. PhD thesis, Instituto Superior Técnico, 2012.

[5] Pierre Pinson. Estimation of the uncertainty in wind power forecasting. PhD thesis, École Nationale Supérieure des Mines de Paris, 2006.

[6] Georges Kariniotakis, Pierre Pinson, Nils Siebert, Gregor Giebel, and Rebecca Barthelmie. The state of the art in short term prediction of wind power - from an offshore perspective, 2004.

[7] Michèle Hibon and Theodoros Evgeniou. To combine or not to combine: selecting among forecasts and their combinations. International Journal of Forecasting, 21(1):15–24, 2005.

[8] Andrew Ng. Machine learning and ai via brain simulations, 2011. URL: https://forum.stanford.edu/events/2011/2011slides/plenary/2011plenaryNg.pdf.

[9] Kyung-Seop Sin, Christopher I. Thornton, Amanda L. Cox, and Steven R. Abt. Methodology for calculating shear stress in a meandering channel, 2012.

[10] Jeff Sanny William Moebs, Samuel J. Ling. University Physics Volume 1, chapter 11 - Angular Mometum, pages 539–565. OpenStax, 2019.

[11] N. I. Fisher. Statistical Analysis of Circular Data. Cambridge University Press, 1993.

[12] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. Frontiers in Neurorobotics, 7:21, 2013. URL: https://www.frontiersin.org/article/10.3389/fnbot.2013.00021, doi:10.3389/fnbot.2013.00021.

[13] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery. URL: https://doi.org/10.1145/2939672.2939785.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[15] ALEX J. SMOLA and BERNHARD SCHOLKOPF. A tutorial on support vector regression, 2002.

[16] Trevor Hastie and Robert Tibshirani. Generalized additive models. Statistical Science, 1(3):297 – 310, 1986. URL: https://doi.org/10.1214/ss/1177013604.

[17] Giorgio Valentini and Francesco Masulli. Ensembles of learning machines. In Maria Marinaro and Roberto Tagliaferri, editors, Neural Nets, 13th Italian Workshop on Neural Nets, WIRN VIETRI 2002, Vietri sul Mare, Italy, May 30-June 1, 2002, Revised Papers, volume 2486 of Lecture Notes in Computer Science, pages 3–22. Springer, 2002. URL: https://www.researchgate.net/publication/220956778_Ensembles_of_Learning_Machines.

[18] Rising Odegua. An empirical study of ensemble techniques (bagging, boosting and stacking), 2019.

[19] Maritza Prieto Emhart. Exploring machine learning models for wind speed prediction, 2018.

[20] Enercast. Accurate power generation forecasts for variable renewable energy sources using enercast $e^3$ technology, 2020.

[21] Muhammad Shahzad Nazir, Fahad Alturise, Sami Alshmrany, Hafiz. M. J Nazir, Muhammad Bilal, Ahmad N. Abdalla, P. Sanjeevikumar, and Ziad M. Ali. Wind generation forecasting methods and proliferation of artificial neural network: A review of five years research trend. Sustainability, 12(9), 2020. URL: https://www.mdpi.com/2071-1050/12/9/3778, doi:10.3390/su12093778.

[22] Laboratório Nacional de Energia e Geologia (LNEG). Provision of meteorological data predicted by a mesoscale model for 12 locations in mainland portugal, 2020.

[23] Vaclav Uruba. On reynolds number physical interpretation, 2018. URL: https://www.researchgate.net/publication/326898812_On_Reynolds_number_physical_interpretation.

[24] C.J.C. Burges and Chris J.C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2:121–167, January 1998. URL: https://www.microsoft.com/en-us/research/publication/a-tutorial-on-support-vector-machines-for-pattern-recognition/.

[25] Xiaodan Yu, Zhiquan Qi, and Yuanmeng Zhao. Support vector regression for newspaper/-magazine sales forecasting. Procedia Computer Science, 17:1055–1062, 2013. URL: https://www.sciencedirect.com/science/article/pii/S1877050913002676, doi:https://doi.org/10.1016/j.procs.2013.05.134.

[26] Kunhui Lin, Qiang Lin, Changle Zhou, and Junfeng Yao. Time series prediction based on linear regression and svr. In Third International Conference on Natural Computation (ICNC 2007), volume 1, pages 688–691, 2007.

[27] Hui Qu and Yu Zhang. A new kernel of support vector regression for forecasting high-frequency stock returns. Mathematical Problems in Engineering, page 9, 2016. URL: https://doi.org/10.1155/2016/4907654.

[28] Rob J Hyndman and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2021.

[29] Kaveh A. Adli. Forecasting steel prices using arimax model: A case study of turkey. The International Journal of Business Management and Technology, page 7, 2021.

[30] Jae Duk Seo. Trend, seasonality, moving average, auto regressive model : My journey to time series data with interactive code, 2018. https://towardsdatascience.com/trend-seasonality-moving-average-auto-regressive-model-my-journey-to-time-series-data-with-edc4c0c8284b. Accessed on 2021-05-26.

[31] Michael Clark. Generalized additive models, 2019. https://m-clark.github.io/generalized-additive-models/. Accessed on 2021-05-26.

[32] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research, 11:169–198, Aug 1999. URL: http://dx.doi.org/10.1613/jair.614, doi:10.1613/jair.614.

[33] Thomas G. Dietterich. Ensemble methods in machine learning. In Multiple Classifier Systems, 2000.

[34] Bernard Ženko Saso Džeroski. Is combining classifiers with stacking better than selecting the best one? Machine Learning 54, page 255–273, 2004. URL: https://doi.org/10.1023/B:MACH.0000015881.36452.6e.

[35] Ryan Turner, David Eriksson, Michael McCourt, Juha Kiili, Eero Laaksonen, Zhen Xu, and Isabelle Guyon. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020, 2021. arXiv:2104.10201.

[36] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019. arXiv:1907.10902.

# Appendix A

# Hyperparameter tunning

Several of the tested models have hyperparameters that influence the predictive ability of the model. Thus the main goal of hyperparameter tunning is to discover the optimal set of hyperparameters that minimizes a specific loss function and, consequently, ensures the best predictions. Failure to do so provides sub-optimal forecasts. Although this procedure is important for the performance of each model, it is computationally expensive and time-consuming. Additionally, it is extremely difficult to find the optimum set of parameters.

The evaluation of each set of hyperparameters should not be done according to the test data, to avoid data bias, and it should not be done based on the training data, to avoid overfitting. For this reason, the training data is going to be divided into K data folds, where each fold is divided into a training set and a validation set, shown in Figure A.1.

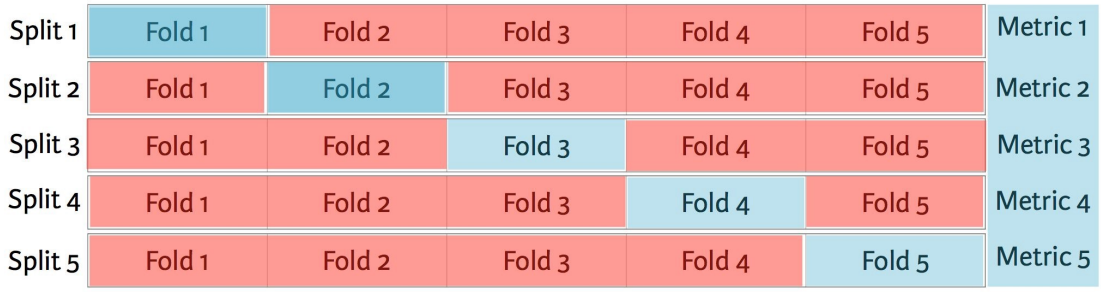


**Figure A.1:** Cross validation with K=5, where the red shapes represent the training set and the blue shapes represent the validation set.

The models are trained using the training sets, and their selection will be done according to the error retrieved by each structure in the validation sets. Since this process is very time-

consuming we chose K=4.

To select which set of hyperparameters to evaluate, a Bayesian Optimisation tuner was considered. A Bayesian Optimisation tuner monitors the results of the previous evaluations and uses them to build a probabilistic model, which is then exploited to make decisions about the next set of hyperparameters to evaluate [35]. This tuner was considered, over a random approach, because it is more likely to lead to better results in fewer trials since its choice of hyperparameters set is based on a determined function and not a random distribution.

The ideal hyperparameters values found for the models in a short-term regime were:

| Model | Hyperparameter | Value |
|---|---|---|
| XGBoost | n_estimators | 316 |
| | max_depth | 6 |
| | learning_rate | 0.0344695627108135 |
| | min_split_loss | 0 |
| | min_child_weight | 6 |
| | reg_lambda | 5.725745074994592 |
| | reg_alpha | 8.49671254352238 |
| | colsample_bytree | 0.9 |
| | subsample | 1 |
| LightGBM | n_estimators | 279 |
| | max_depth | 6 |
| | learning_rate | 0.036596735841001225 |
| | min_split_gain | 0 |
| | min_child_weight | 3 |
| | reg_lambda | 9.987890855343101 |
| | reg_alpha | 9.962119160931348 |
| | colsample_bytree | 1 |
| | subsample | 1 |
| SVM | Kernel | Linear |
| | C | 0.02408999463029706 |
| ARIMAX | p | 15 |
| | d | 1 |
| | q | 1 |
| GAMAR | lambda | 2.33653853605221 |
| | p | 15 |

**Table A.1:** Hyperparameters for each model in a short-term regime

While for a medium-term regime, the ideal hyperparameters values found were:

| Model | Hyperparameter | Value |
|---|---|---|
| XGBoost | n_estimators | 440 |
| | max_depth | 5 |
| | learning_rate | 0.03517086616661376 |
| | min_split_loss | 0 |
| | min_child_weight | 7 |
| | reg_lambda | 8.583257540867343 |
| | reg_alpha | 3.5110206194236695 |
| | colsample_bytree | 0.9 |
| | subsample | 1 |
| LightGBM | n_estimators | 257 |
| | max_depth | 5 |
| | learning_rate | 0.04681119738908505 |
| | min_split_gain | 0 |
| | min_child_weight | 10 |
| | reg_lambda | 8.70606067758942 |
| | reg_alpha | 1.2711989016747252 |
| | colsample_bytree | 1 |
| | subsample | 1 |
| SVM | Kernel | RBF |
| | C | 4.650380553088765 |
| ARIMAX | p | 15 |
| | d | 1 |
| | q | 1 |
| GAMAR | lambda | 9500 |
| | p | 15 |

**Table A.2:** Hyperparameters for each model in a medium-term regime