MASTER'S DEGREE THESIS

Interuniversity Master in Statistics and Operations Research

Title: Wind Power Forecasting by Nonparametric and Parametric Time Series Models

Author: Patricia Diana Tencaliec

Advisor: prof. M. Pilar Muñoz Gracia

Department: Statistics and Operations Research

University: Universitat Politecnica de Catalunya

Academic year: 2012-2013

Facultat de Matemàtiques i Estadística
UNIVERSITAT POLITÈCNICA DE CATALUNYA

UNIVERSITAT DE BARCELONA

Master thesis

Wind Power Forecasting by Nonparametric and Parametric Time Series Models

Patricia Diana Tencaliec

Advisor: prof. M. Pilar Muñoz Gracia

Department of Statistics and Operations Research

Preface

This report represents the documentation of the project entitled "Wind Power Forecasting by Nonparametric and Parametric Time Series Models". The project was prepared between March 2012 and the 10th of October 2012, at Universitat Politècnica de Catalunya.

The main focus of the project was to study methods for forecasting wind speed and wind power for a short term horizon. In order to find the estimates, parametric and nonparametric models were used.

All the work was implemented in the statistical program R Software, with the exception of some large simulations that were passed to Matlab for being faster.

The literature references are shown in square brackets by numbers. The list of the references is presented in the chapter Bibliography. Appendices are assigned with letters, and arranged in alphabetical order at the end of the report. Figures and tables are numbered in a continuous form without referring to the chapter.

I would like to thank my supervisor, prof. Maria Pilar Muñoz Gracia, for the constructive feedback during the entire period of the project and for her stress-free attitude that made the meetings easier to bear.

10th of October 2012

Abstract

One of the major challenges to supporting, facilitating and developing wind generated power is matching supply and demand. Wind power is subject to fluctuations due to the stochastic nature of the wind. Predicting the power from the wind turbines is currently an important research topic. During the past decades several studies, including autoregressive models, Kalman filters, Bayesian models, were conducted in order to forecast short-term prediction of wind power. In this investigation the parametric ARMA models and the nonparametric Nadaraya-Watson estimator will be used in order to predict wind speed. Besides these two models, two semiparametric approaches are presented, by combining the above models and using the residuals. The accuracy of the predictions was tested by using the RMSE indicator. Results show that nonlinear models are a better fit for the wind data.

Keywords: ARMA, nonparametric, imputation, wind, forecast

Index

Chapter 1. Introduction

The introduction serves mainly as an overview of the current world energy condition relative to the need for renewable energies in the energy production scheme. Special attention is paid to wind, as a modern and environmentally friendly source of energy. The motivation, the goals as well as the limitations that were set at the beginning of the project are also covered in this chapter.

1.1. Historical overview of wind power development

The wind has been used for many centuries as power for sailing ships and until the discovery of the engines this was the only way for the ships to sail. Wind turbines date from centuries B.C., sources report them being used by the Babylonians for irrigations [1].

Denmark was reported as the first country that used the wind for power generation. About twenty years later, different types of turbine appeared on the American market [2], becoming more and more popular over the entire world. The costs for the wind power energy started to decline slowly.

The energy used for the past comes mainly from oil, coal and natural gas, also known as conventional energy sources [3]. Coal and oil are reported as the oldest sources of energy and they became an essential key point in the entire world leading to different economic and political conflicts [4]. The energy security, the reliability of energy supply and the differences in consumption between poor and rich countries, are some real problems that at the present moment still generate friction between nations. Adding to the before-mentioned social-economic factors the increased environmental concern, new sources of energy started to be studied and considered.

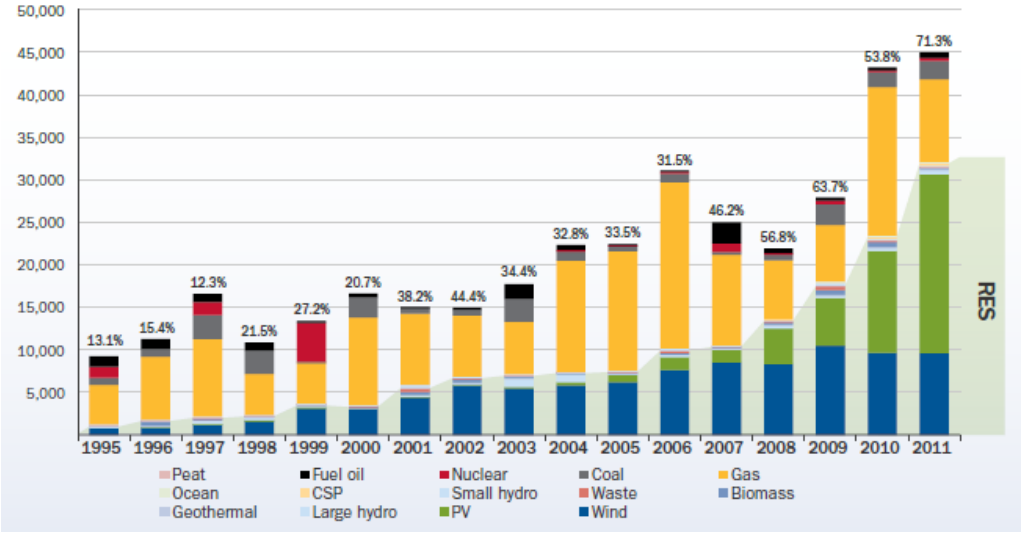
Renewable energy is obtained from sources that are inexhaustible, unlike fossil fuels, involving natural phenomena such as sunlight, wind, tides, plant growth and geothermal heat, as the International Energy Agency explains: *"Renewable energy is derived from natural processes that are replenished constantly. In its various forms, it derives directly from the sun, or from heat generated deep within the earth. Included in the definition is electricity and heat generated from solar, wind, ocean, hydropower, biomass, geothermal resources, and biofuels and hydrogen derived from renewable resources."* [3]

Many countries are introducing policies meant to speed up the transition toward a low carbon economy and to increase the use of renewable energy. European Union is pursuing the implementation of its aspiring 20/20/20 targets, which aim, by 2020, to reduce the carbon emissions by 20% (as compared to 1990). Moreover, the amount of renewable energy should

be increased to 20% of the energy supply and the overall energy consumption should be reduced by 20% through energy efficiency [5].

In 2006, Europe was importing 54% of its energy and unless a change was going to be made in Europe's supply policy, that share was likely to increase. Most of Europe's oil was coming from the Middle East and the greater share of its gas from just three countries: Russia, Algeria and Norway [6].

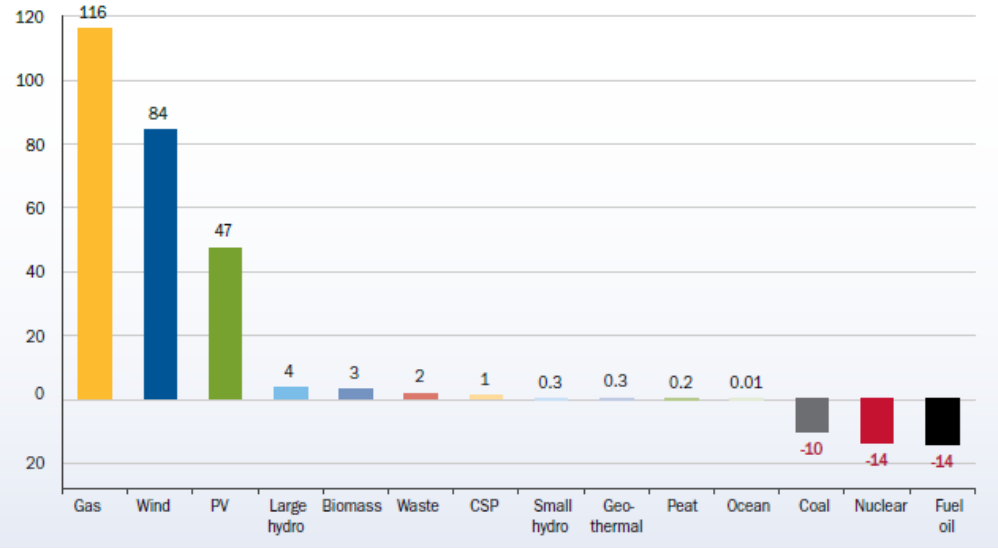
As Europe is running out of local fossil fuels, the renewable energy sources became very popular. Renewable capacity installations have been growing over the past eleven years. In 2000, new renewable power installations equaled 3.5 GW (20.7% of the new power installations) and it has been growing over the last eleven years arriving in 2011 to reach 32 GW (71.3% of the new power installation). What it must not be ignored, is the share of renewables in the total generated capacity. As it can be seen in Figure 1, in 2000 the renewable energy share represented less than 5% of the total generated capacity, while in 2011 it has been increasing to more than 30% [7].



Source: EWEA

Figure 1. EU installed power generating capacity per year in MW and Renewable Energy Source (RES) share (%)

The evidence that European Union begins to move away from conventional energy sources like coal, oil or nuclear is very clear and it is presented in the 2011 Wind annual report of EWEA. Analyzing the amount of electricity installations over the last decade from Figure 2, it can be noticed that coal, nuclear and oil are reduced. This means that some conventional installations have been closed and replaced by renewable energy sources. The net growth in the last eleven years of gas power is of 116 GW, wind power of 84.2 GW and solar photovoltaic of 47.4 MW, while oil-based installation reduced with 14.2 GW, nuclear with 13.5 GW and coal with 10.3 GW. The other renewable sources (hydro, biomass, ocean energies, etc.) are also reporting increases in total installed capacity over the past decade, but at slower pace when compared with wind and photovoltaic [7].

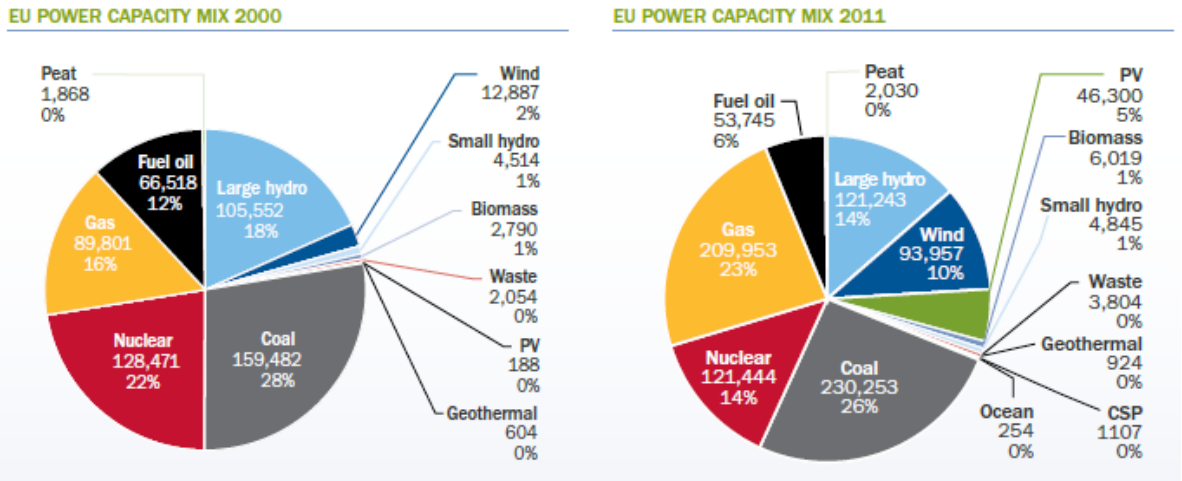


Source: EWEA

Figure 2. Net electricity generating installations in EU 2000-2011 in GW

As shown previously, due to significant changes in the electricity production-map, in the last years, renewable sources such as wind, hydro, solar photovoltaic plants or biomass, started playing an important role in our modern society.

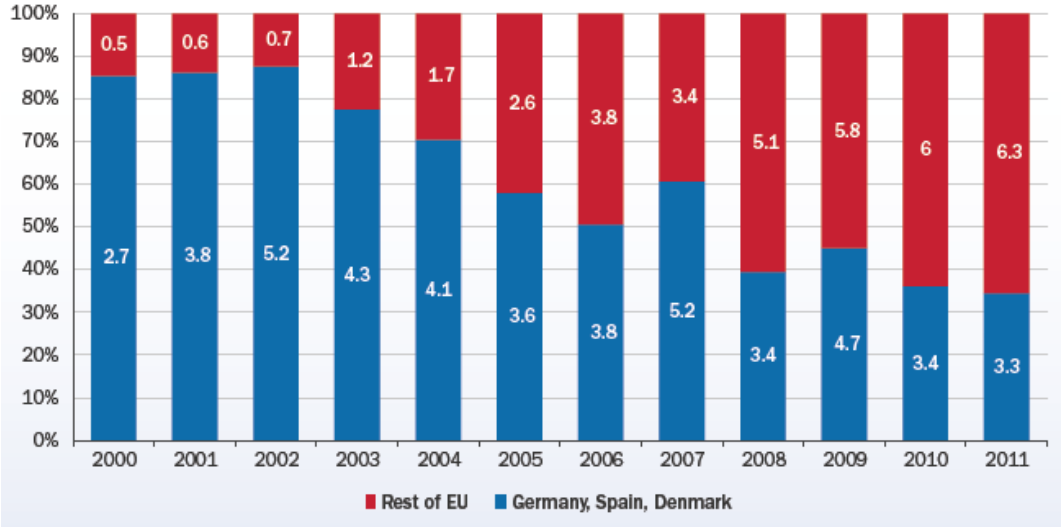
Focusing on wind power, a study regarding the European power capacity over the last decade revealed that total wind installed power capacity has increased more than four times from 2000 to 2011. As it can be seen in Figure 3, in 2000 the power capacity mix was govern by coal and nuclear energy sources, gathering 50% of the total power while the renewable energy sources occupied less than 23%. At the same time, wind power capacity summed only a 2% of the total capacity, placing it on the sixth place. In 2011, a decrease of 10% in coal and nuclear power capacity was registered, gathering this time only 40% of the total power capacity mix. Wind power's share of the total capacity has increased five times, arriving at 10% of the total European capacity. The renewable capacity has increased, occupying in 2011 31% of the mix, so it has been growing more than a third [7] comparing to 2000.



Source: EWEA

Figure 3. EU Power Mix in 2000 and 2011

The production of wind power evolved differently among European countries over the past decade. In 2000, Denmark, Germany and Spain were the leaders for wind energy development, as it can be seen in Figure 4, representing more than 85% of the total 3.2 GW capacity of Europe. In 2011 from the total wind power of 9.6 GW, only a third (3.3 GW) came from Germany, Spain and Denmark. According to EWEA statistics, other countries like France and Italy overpowered Denmark, Germany and Spain remaining on the first and second place, respectively.



Source: EWEA

Figure 4. Denmark, Germany and Spain's share of EU wind power market (GW)

The continuous decreasing of conventional energy resources and the negative effect on the environment led to the need of finding new alternatives. The wind represents the energy source with the fastest development in the past decades from all the other renewables. Knowledge about the wind is very important for the planning and designing of the wind turbines and wind parks, therefore studies in this direction pose great interest for the energy-industry.

1.2. Project motivation and goals

Taking into account the data and the statistics exposed previously, it is clear that wind energy is an important player in nowadays energy technology. However, the uncertainty in the wind power production raises concerns, both for the generators and the operators, and it has become a major topic in the last decades. The motivation for this project came mainly from the special attention in the field of wind power forecasts, the accuracy in this area being a valuable factor in the production planning and the optimization of the financial income.

From a statistical point of view, the wind power has some distinct characteristics. On one hand, the relationship between the wind speed and wind power is nonlinear. Then, this relationship is time varying because the wind speed depends on several factors as: wind direction, air density, temperature, terrain type or height, etc. One cannot include all these

variables in a forecasting model because some of them are difficult to predict or even to measure. So a plausible method will be to take into consideration as much information as possible about this relationship.

There are several models regarding the wind speed forecasts reported in the literature over the last decades as Kalman filters [8], ARIMA models [9], Bayesian models [10], local quantile regression [11] or empirical orthogonal function analysis [12]. Recently, models based on artificial intelligence techniques have been used for wind speed forecasts such as neural networks [13] or fuzzy logic [14].

The goal of this project is to reduce the error of wind power prediction by bringing a new approach by means of a nonparametric method. Since it has been witnessed that it is not realistic to formulate assumptions regarding the shape of the wind power distributions, the Nadaraya-Watson estimator will be used as a method of predicting the wind speed. This approach involves a nonparametric assumption of the variables' densities and also it takes into account the nonlinear relationship between the wind speed and the wind power.

1.3. Project limitations

A number of limitations need to be noted regarding the present study. The project timeline was delayed due to some issues found in the preprocessing phase. At the time of the descriptive statistics of the data, an important number of missing values were discovered, so a step of imputation had to be included in the project.

The forecast process and analysis was also delayed as a consequence of the duration of the simulations. Hence, for time optimization, Matlab was used for some of the procedures from the project, R Software being very slow. The amount of the data made the analysis possible for only a representative sample of the wind turbines.

1.4. Project outline

Chapter 2 starts by introducing the datasets used in the project and the methodology used to collect the data. Then, it offers a statistical description of the wind characteristics, speed and direction, along with the analysis of the wind power, ending by a short study of the outliers.

Chapter 3 contains a detailed analysis of the missing values. Five methods were introduced for the missing estimation: imputation by median, *k*-nearest neighbor imputation, linear models (ARMA+GARCH), Estimation-Maximization (EM) algorithm and a nonparametric approach. The yielded results can be found in Section 3.2.5 of this chapter.

Then, a time series concepts introduction is presented in the first part of Chapter 4. The forecasting methodologies for the wind speed and wind power are discussed in Section

4.2., being followed by the characterization of the prediction accuracy and the computation of the prediction intervals.

The conclusions that were taken based on the results from all the approaches can be found in Chapter 5. In this part there is also a discussion about further research topics related to this investigation.

Chapter 2. Descriptive analysis

This chapter starts with a description of the wind data collection process, along with some short details about the wind farm location and layout. It continues with an analysis of the wind speed and wind power, reaching points as distributions, tendencies and dispersions study. Afterwards, a study regarding the relationship wind power-speed-direction is instigated. In the last part of this chapter, a preprocessing step is handled, namely the outlier's analysis.

2.1. Wind data collection

The data used for this study was collected from two onshore wind farms situated in the south of Spain, in Cadiz province, during the years of 2009 and 2010. One wind farm contains 6 turbines with a total power of 10 MW while the other wind farm has 28 turbines with a total power of 47 MW. There are two types of turbines installed in the farms, both from the same French manufacturer Ecotecnia, with the same nominal power of 1670 kW. In Table 1 one can find these details and in Appendix A one can find details about the technical data of the turbines.

Table 1. Wind farms general information [15]

Wind farm name	# of turbines	Type of turbines	Operator	Total power
Wind Farm 1	6	Ecotecnia 74	Aerogeneradores del Sur	10,020 kW
Wind Farm 2	28	Ecotecnia 80 1.6	Aerogeneradores del Sur	46,760 kW

The position of a wind farm is very important and multiple factors must be taken into account when designing a new farm. One of these factors is the climatic environment. A turbine output is influenced by the wind direction and wind speed. Focusing on the province of Cadiz, it can be said that it has a Mediterranean climate with influences from the ocean. The summers are warm and mostly dry. The winters are mild and rainy. Because of its proximity to the Atlantic Ocean, precipitation is quite high especially in December and January and only small variation in temperature are noticed [16].

The detailed locations of the 34 turbines are illustrated in Figure 5. They are situated from -5.73 and -5.7 longitude, 36.16 and 36.17 latitude. The turbines are labeled from 101 to 106, 201 to 210, 301 to 311 and 401 to 407.

Two datasets were used in this study. One dataset is containing measurements at a 3 hour interval of wind speed and wind direction for the Cadiz province, obtained from AEMET.

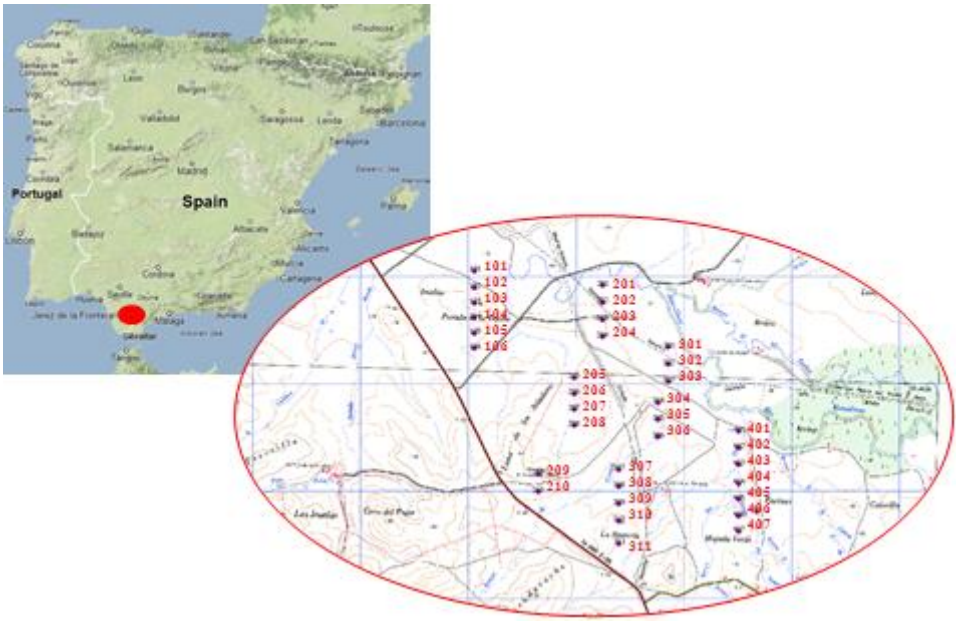


Figure 5. Location and layout of wind power farms

The other dataset consists of 10-minute sample rate measurements, each one containing 17 variables: time (year, month, day, hour, minute), location (Universal Transverse Mercator coordinate system – XUTM and YUTM, latitude, longitude, turbine, generator) and measurements (wind power – minim, maxim, average, wind speed – minim, maxim, average).

It must be mentioned that missing data are present in this second dataset. This topic will be treated later in Chapter 3.

2.2. Statistical description

Wind power usually presents frequent changes in its shape, mainly due to the nonlinearity of the wind behavior and the power transformation process in each wind farms. In order to bring an accurate model of forecasting the wind power, firstly, the relationship between wind speed and wind power must be understood. As it is explained in [17], the behavior of the power output depends not only on the wind speed, but also on different meteorological variables as wind direction, temperature, air density or precipitation.

An exact analysis of statistical wind data is an important step in the prediction of wind power generated by a turbine. After the wind speed probability distribution is identified, the wind energy distribution can be obtained. Therefore, wind speed analysis is important in the evaluation of wind energy potential.

In order to analyze the behavior of the wind speed and power in the area described in the previous section, several wind turbines were selected randomly from the wind farm: turbine 105 – west of the farm zone, turbine 401 – east, turbine 201 – north and turbine 311 – south.

2.2.1. Wind speed

It is essential in the wind industry to be capable of describing the variation of wind speed. Turbine manufacturers need this information to adjust and improve the design of the turbines and turbine operators need this information to assess their income from the power generation.

In order to describe the wind speed data, first the distribution of the data will be studied then the central tendency and dispersion. In the last part of the study, the monthly and hourly behavior will be discussed.

a) Distribution, central tendency and dispersion

In order to study the distribution of the data, a histogram representation will be used. As mentioned in [18], the histogram is "the oldest and most widely used density estimator" and it is define by:

$$\hat{f}(x) = \frac{1}{nh}(\text{no. of } X_i \text{ in the same bin as } x) \quad (1)$$

As it can be seen in Eq. (1) the histogram estimator depends on the bin width *h*. Over the past years, there were made several attempts in order to find the optimal number of bins, but most of them make assumptions about the shape of the distribution [19]. Sturges [20] formulated an approach considering that the ith bin is the binomial coefficient, so as the number of bins increases the histogram has the shape of a normal density. If the data are not normally distributed, one could consider Doane's rule [21] which is an improvement to Sturges' formula for data that is not normal. Other two alternatives for finding the number of bins are Scott's [22] and Freedman–Diaconis' [23] rules. These last two rules have a more sophisticated statistical theory than the first two, but they are not difficult to use. One can find in Table 2 the formula for the presented approaches.

Table 2. Optimal bin's width approaches

Sturges	$k = 1 + \log_2 n$ where *n* is the total sample size	(2)
Doane	$k = 1 + \log_e n + \log_e(1 + \hat{k}\sqrt{\frac{n}{6}})$ where $\hat{k}$ is the estimated kurtosis of the distribution	(3)
Scott	$h = \frac{3.5\hat{\sigma}}{\sqrt[3]{n}}$ where $\hat{\sigma}$ is the standard deviation	(4)
Freedman–Diaconis'	$h = 2\frac{IQR(x)}{\sqrt[3]{n}}$ where *IQR* is based on the interquartile range	(5)

Applying the approaches presented above, the resulted histograms are presented in Figure 6. Due to the space amount, only turbine 105 will be represented here, but the other turbines a similar in shape.

Analyzing the histograms it can be noticed that, in this case, Scott's and Freedman–Diaconis' approaches (FD) gives similar results, so from now on the histograms resulted from Freedman–Diaconis' rule will be used in the study.

Looking at the histogram it can be said that the data is not normal distributed. The fact that the first classes have lot of observations makes the shape of the distribution an asymmetrical one, more exactly an asymmetry to the right. This phenomenon is usually found in processes that have many values close to zero or a natural limit, indicating that maybe a transformation may help make data normal.

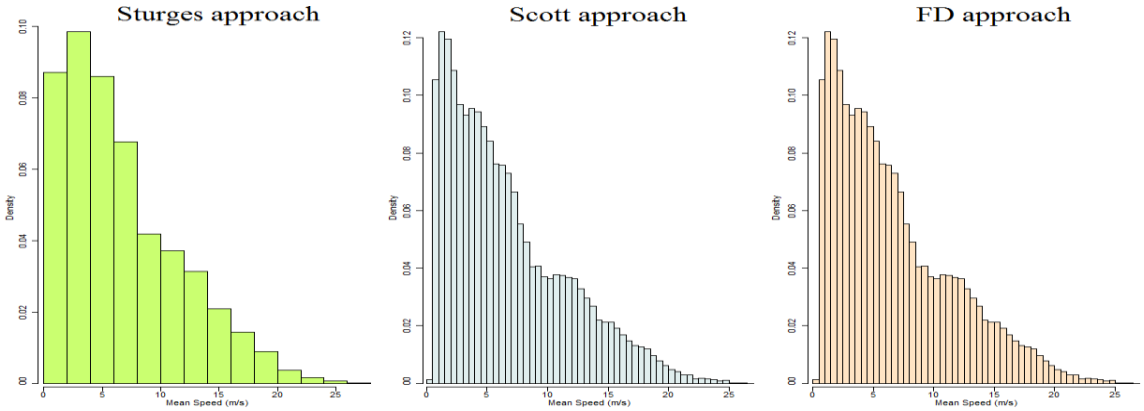


Figure 6. Turbine 105 histograms for 2009

To prove these properties, the moments of the distribution will be analyzed. Three commonly reported measures of central tendency are the mean (first moment), median and mode. The mean is the central location of the distribution, the median is the value separating the higher half of the ordered data from the lower half and the mode is the number that appears most often in the dataset. It must be mentioned that the mean is influenced by the extreme data, so the mean tends to "run" in the direction of the outliers and it can distort the data's central tendency.

In order to present the results, turbine 105 will be used as a detailed example; stating that the other turbines have similar behavior. Table 3 presents the values for all the turbines and all the indicators. In the discussed case, the mean and the median are 6.74 m/s and 5.50 m/s, respectively. Note that the median is lower than the mean, implying that it is not influenced by the outliers like the mean, but it depends on the number of data values.

Another point that must be discussed is the value obtained for the mode in 2010 for turbine 105 and 311, both being out of the ordinary in comparison with the rest of the results. A mode of 0.6 m/s in 2010 for turbine 105 means that the values registered for the wind speed are more often seen around this proportion, comparing with the previous year, where the mode was twice as large. In other words, it can be concluded that in 2010 it was registered a decreasing in the wind speed for turbine 105. The contrary effect is noticed at turbine 311, where an increase of wind speed lead to an almost three times higher mode.

Table 3. Descriptive statistics for wind speed

Year	Turbine	#obs.	#missings	Mean	Median	Mode	Std. dev.	Skewness	Kurtosis
2009	**105**	52560	2374	6.74	5.50	1.20	4.93	0.95	3.30
2010		52560	5857	7.03	5.80	0.60	5.09	0.89	3.06
2009	**201**	52560	3286	6.45	5.10	1.30	4.81	1.01	3.41
2010		52560	5785	6.77	5.50	1.30	4.95	0.96	3.24
2009	**311**	52560	2876	6.58	5.10	1.30	5.10	0.87	3.87
2010		52560	5920	6.17	5.00	3.70	4.44	1.11	4.03
2009	**401**	52560	3080	6.38	5.00	1.50	4.83	0.99	3.34
2010		52560	6282	6.77	5.50	1.30	4.92	0.82	2.91

Another important property that must be studied for a dataset is the dispersion of the data by analyzing the variance (second central moment), the standard deviation or the interquartile range.

The standard deviation describes the location of the data with respect to the mean. Using the standard deviation of 4.93 m/s for turbine 105 in 2009, bounds were created around the mean in order to quantify the data at ±1, ±2, 3 or 4 standard deviations away from the mean as can be seen in Figure 7. Looking at Figure 7, it can be seen that most of the data are in the first range standard deviation from the mean, but in the same time a tail can be observed on the right part of the plot, making the wind speed distribution an asymmetric one.

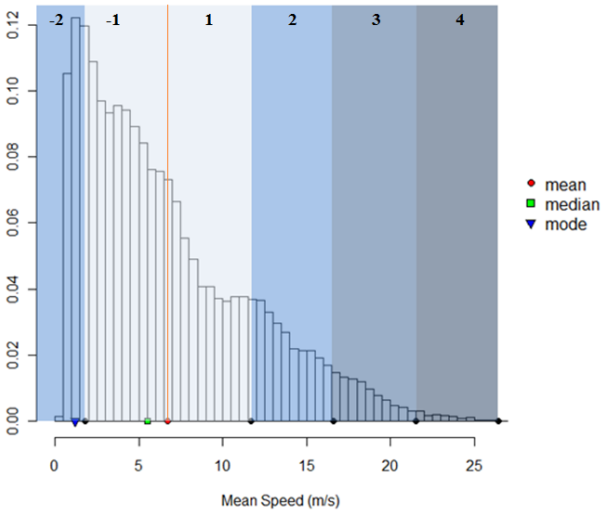


Figure 7. Central tendencies and standard deviation of wind speed for turbine 105, year 2009

The interquartile range (IQR) is another useful method of observing data dispersion and it is equal to the difference between the upper and the lower quartiles [24]. The Box-whisker plots or boxplots show a simple illustration of the data based on the quartiles, see Figure 8. One can see that wind speed data have the median positioned toward the lower part of the data, indicating an asymmetrical distribution to the right. The minimum value is at zero and the maximum is around 20 m/s, except for turbine 311, which in 2010 noticed an increase in wind speed.

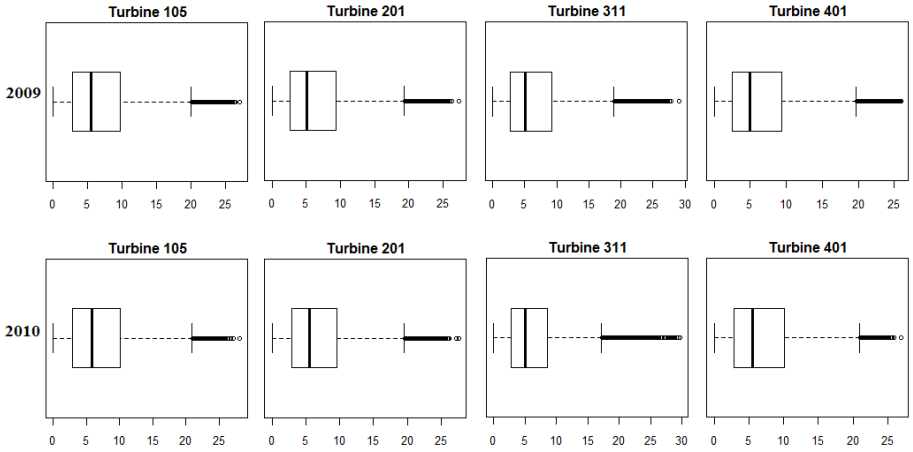


Figure 8. Boxplots for wind speed data

Skewness (third central moment) is a measure of asymmetry and kurtosis (forth central moments) is an indicator of the tail behavior. It is said that a distribution has positive excess kurtosis or heavy tails if the afferent value is greater than 3, and negative excess kurtosis or short tails if it is lower than 3; 3 being the value for a normal distribution [25]. On the other hand, a distribution can have negative skew, meaning that its left tail is longer, or positive skew when its right tail is longer.

In the wind speed data case, see Figure 9, the distribution of the data is positively skewed to the right and it has positive excess kurtosis, for all the turbines.

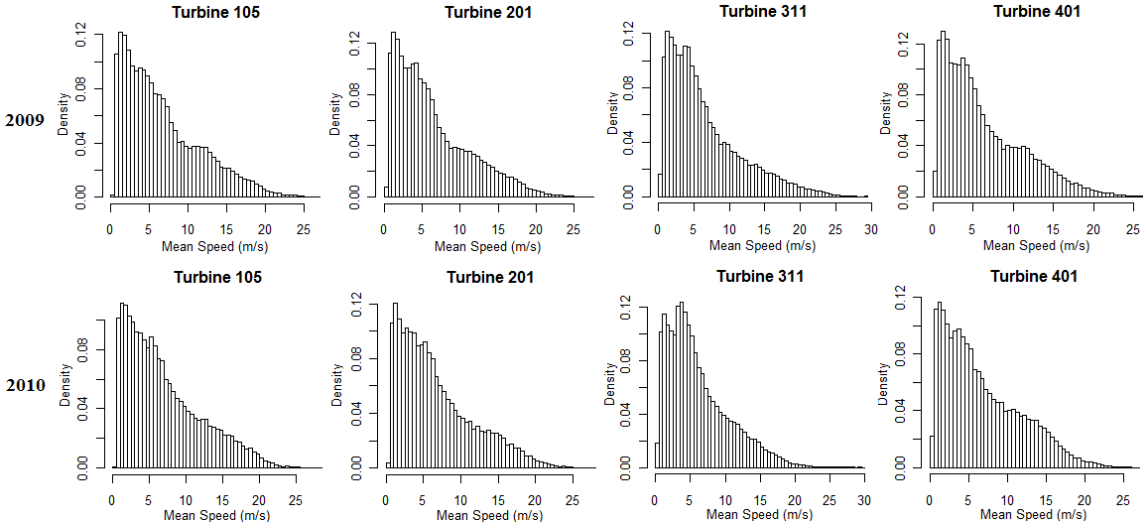


Figure 9. Histograms for the wind speed data

These two central moments can help in testing for the normality of the data. In order to test this, the Jarque-Bera test [26], which uses the sample skewness $\hat{S}$ and sample kurtosis $\hat{K}$, will be carried on. The results show that, for all the turbines, the data are not normally distributed because the Jarque-Bera statistic indicator is less than the significance level of 0.05. These results reinforce the conclusion that was dropped from the histogram plots from above.

Even though the usage of histograms as a method of data analysis is very simple and easy to use, it can have drawbacks. For example, the discontinuity of the histograms can

cause problems if a derivative of the estimates is required. Then, the histograms depend on the origin and the bin's width, which can be difficult to choose.

There are people that consider this method "mathematically insufficient" and do not recommend it, but nevertheless, the histograms remain the easiest tools in providing a quick overview of some data [18]. In the next chapter of this thesis more mathematically accurate methods of estimating the density will be presented.

b) *Monthly and hourly variation of wind speed*

To better understand the behavior of wind speed data, a monthly and hourly mean wind speed analysis will be carried on, for both the 2009 and 2010 measurements; see Figure 10 and Figure 11, respectively. It must be mentioned that in June 2010 there are not any registered measurements.

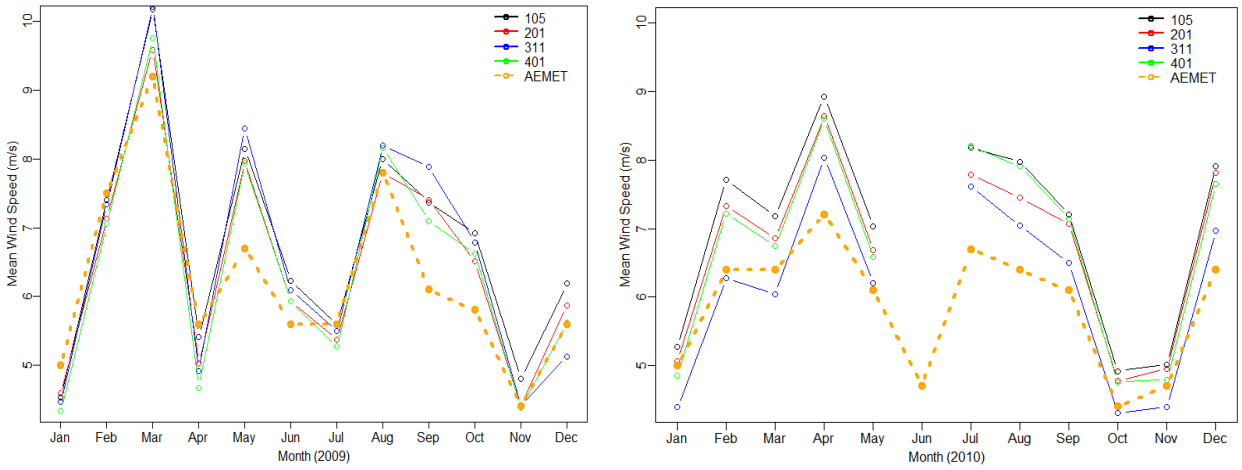


Figure 10. Monthly variation of the mean wind speed in 2009 and 2010

In 2009 the behavior of the wind speed is almost similar for all the locations selected. In 2010 the picture is different; the lines representing the wind speed are not superimposed, see for example turbine 311. Comparing the two years, it can be observed that an overall increase is present though; the mean wind speed in 2009 is 6.4 m/s while in 2010 it is 6.74 m/s, but nevertheless, in 2009 the maximum wind speed was 10.17 m/s and in 2010 it was 8.93 m/s. This behavior may conclude that in 2009 there were months with stronger wind speed, while in 2010 the wind speed blew more constantly bringing a higher average.

The maximum wind speed value for 2009 is registered in March while in 2010 in April. Analyzing the plots one can notice the differences in wind speed from one year to another. While in 2009 from January to March it was an increasing pattern in wind speed, in 2010 the wind speed is decreasing from February to March, followed by an increase until April. The last part of the year is more alike, excepting the last 2 month where in 2010 the wind speed was stronger.

In order to have an exact analysis, the wind speed measurements were compared with the measurements registered by AEMET (Agencia Estatal de Meteorologia)[1], the behavior resulted being almost the same.

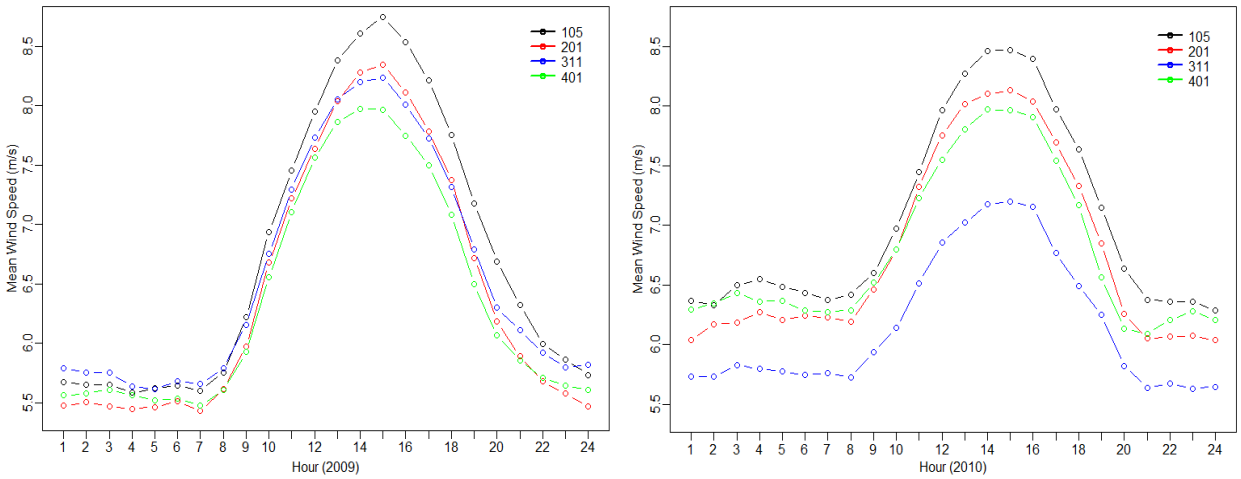


Figure 11. Hourly variation of the mean wind speed in 2009 and 2010

Figure 11 describes the hourly variation of the mean wind speed in the four areas. A clear feature is that the hourly mean wind speed is much higher during the hours of the day than during the night. A distinct increase of wind speed is observed at 7:00 – 8:00; the highest mean wind speed occurs around 14:00-15:00 with values, for all four turbines, of 8.75 m/s, 8.35 m/s, 8.24 m/s and 7.98 m/s in 2009 and 8.47 m/s, 8.13 m/s, 7.20 m/s and 7.97 m/s in 2010. The afternoon is described by a decreasing pattern of the wind speed until the minimum mean wind speed is reached at 24:00. Generally speaking, the hourly mean wind speed fluctuates from day to night and it does not vary from site to site.

2.2.2. Wind power

It is known from physics that the mean wind power depends on the wind speed v (m/s), the blade sweep area A (m^2) and on the air density ρ (kg/m^3) [27], and it is given by:

$$P = \frac{1}{2}\rho A v^3 \tag{6}$$

Because wind speed is not a constant during a certain period, the wind power will become:

$$P = \frac{1}{2}\rho A \int_0^\infty v^3 f(v) dv \tag{7}$$

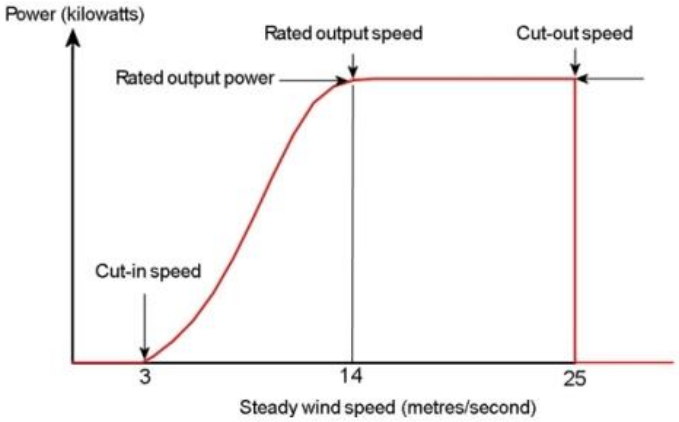
where *f(v)* is the wind speed distribution function [28].

[1] The data can be found online at: ftp://ftpdatos.aemet.es/series_climatologicas/valores_mensuales/anual/

a) Wind turbine power output variation with wind speed

In practice, the real relationship between wind power and wind speed, can be more complicated than the one presented in Eq. (6). Figure 12 illustrates this relationship, also called the power curve, for the two turbines that are discussed in this study. The information included in the power curve is supplied by the manufacturer, in the turbine's datasheet.

One can see from the power curve that below a minimum wind speed (called cut-in speed), in this case 3 m/s, the turbine does not produce any power. After this point, the power increases as the wind speed does, following a pattern similar to the one presented in Eq. (6). When the wind speed approaches the nominal speed, in this case 14 m/s, the wind power reaches the rated output power of the turbine. It maintains this level of power until the wind speed exceeds the so-called cut-out speed, in this case 25 m/s, when the turbine is disconnected in order to prevent certain damages.



Source: WindPower Program

Figure 12. Wind power curve for Ecotecnia 74 and Ecotecnia 80 1.6 turbines

b) Distribution, central tendency and dispersion

In the wind power distribution there is a higher frequency in the extremes than the mean wind power. This means that the wind power histograms will have a no definable shape, some peaks will appear on the higher and lower end of the distribution, see Figure 13. The fact that the wind power distribution has a spike on the right part is due to the cut-out speed point. At that point the wind power cannot be measured anymore because the turbine is disconnected.

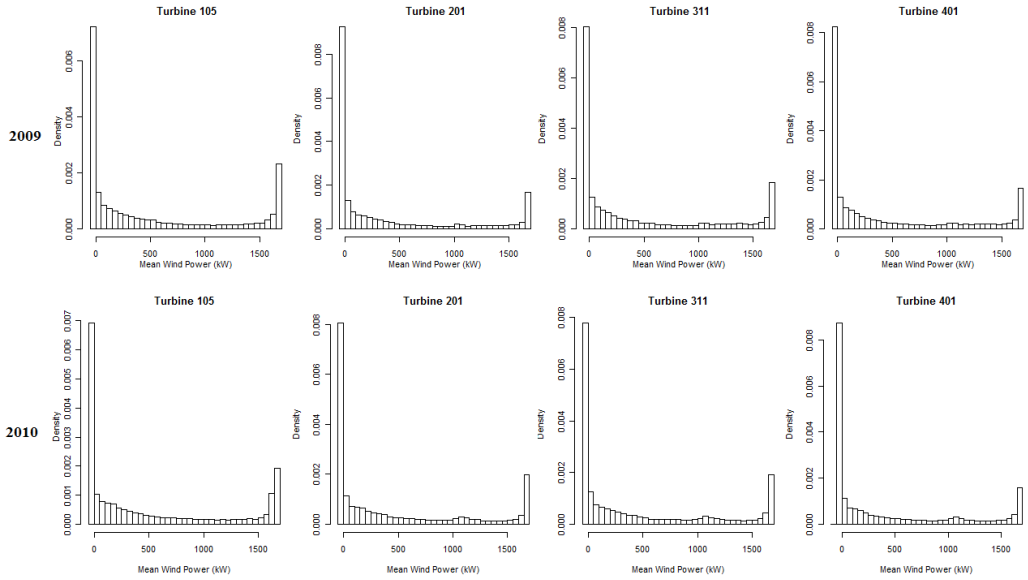


Figure 13. Histograms for the wind power data

As, it was said before, the standard deviation describes the location of the data with respect to the mean. Looking at Figure 14, it can be seen that most of the data are in the first range standard deviation from the mean and that the data are, as in the case of wind speed distribution, asymmetric.

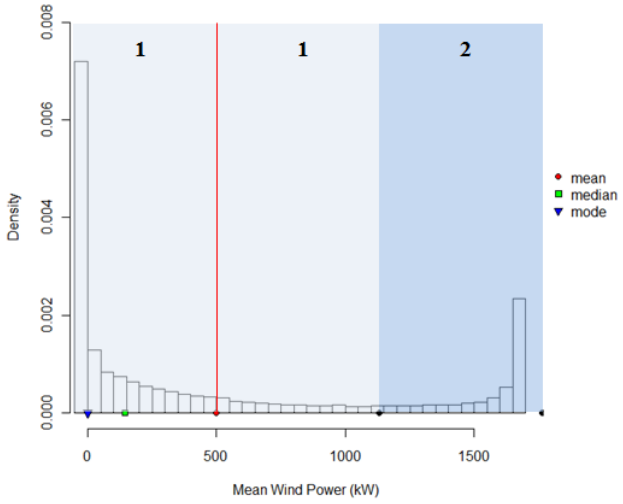


Figure 14. Central tendencies and standard deviation of wind power for turbine 105, year 2009

c) Monthly and hourly variation of wind power

Figure 15 shows monthly variations of the wind power density for the years of 2009 and 2010. Important monthly changes in the wind power density were found with a maximum value of 853.65 kW in March for turbine 105 and a minimum value of 133.08 kW in November for turbine 201. This difference may be due to the fact that wind power is proportional to the cube of the wind speed, which is three times greater in March than in November, as shown in Figure 10.

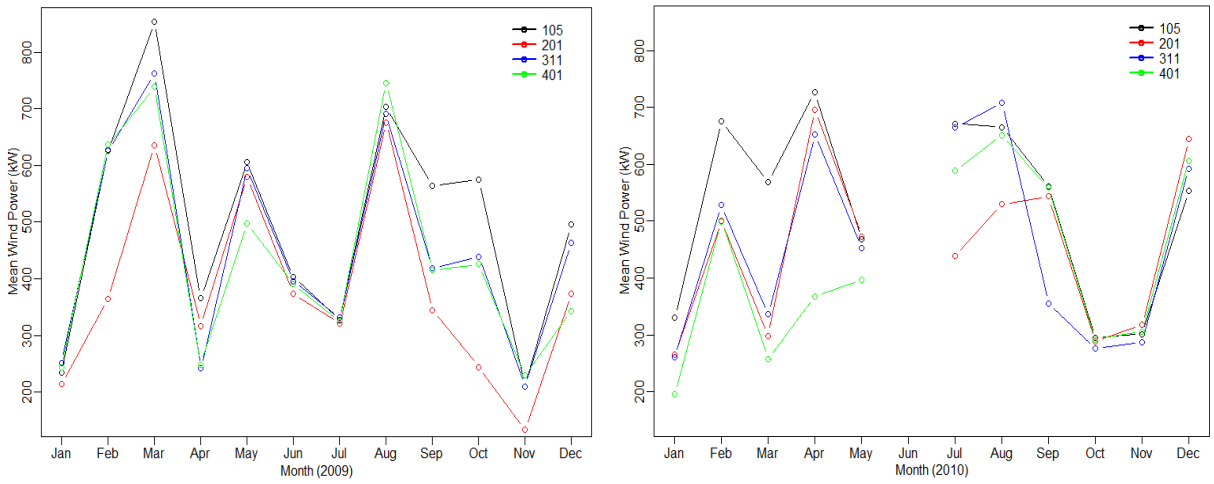


Figure 15. Monthly variation of the mean wind power in 2009 and 2010

Some discrepancies were identified in the wind power monthly variation from 2010. For example, the mean wind speed from March and May is found to be similar, see Figure 10, but wind power is different in these two months. This anomaly can be accounted for by differences in the standard deviations of the wind speed distributions in these months. As shown in Table 4, the standard deviations in May are greater than those in March. Therefore, if two months with the same mean wind speed, but one registering a higher standard deviation, this latter one will be more probable in experiencing higher wind power.

Table 4. Wind speed monthly standard deviations

Turbine	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
105	3.79	4.63	4.84	6.33	5.12	NA	5.44	5.71	4.84	3.55	3.59	5.34
201	3.59	4.47	4.55	6.14	5.01	NA	5.41	5.35	4.76	3.42	3.77	5.43
311	2.89	3.55	4.07	5.91	4.66	NA	5.17	4.74	4.2	2.91	2.98	4.54
401	3.59	4.31	4.54	6.14	5.02	NA	5.33	5.45	4.56	3.42	3.68	5.15

It can be seen also in Figure 15 that the wind power at different sites does not vary greatly in 2009, but there are some significant differences in 2010 at turbine 105.

The hourly wind power variations illustrated in Figure 16, are very similar to the variations of the mean wind speed (Figure 11). Comparing the four turbines it is noticed that in 2009 turbine 201 produces the less power and turbine 105 the greater amount, while in 2010 the less productive is turbine 401 and the more productive is still turbine 105.

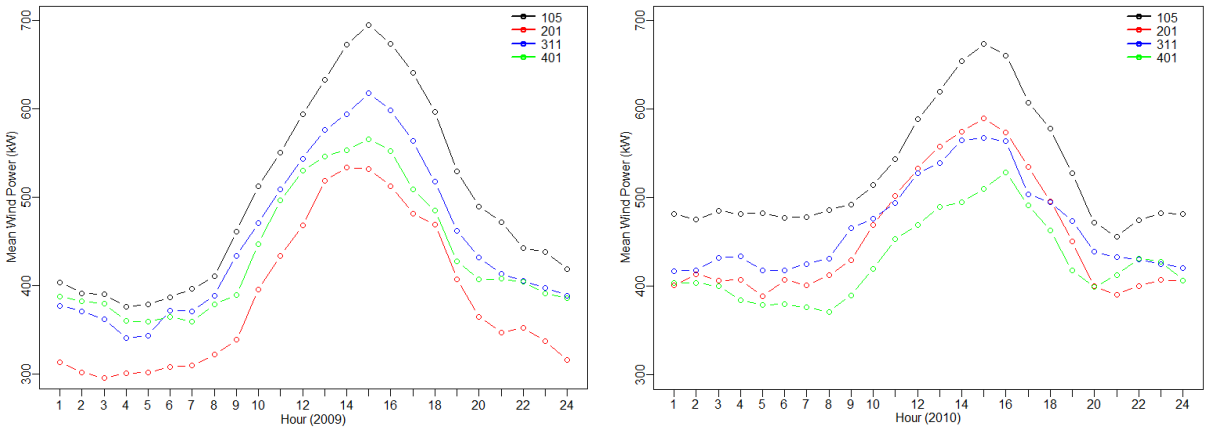


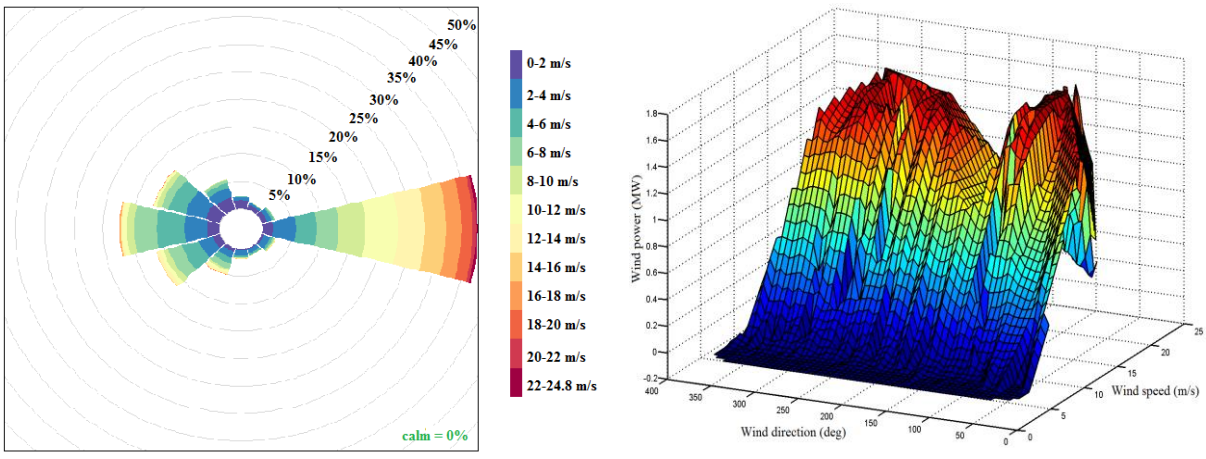
Figure 16. Hourly variation of the mean wind power in 2009 and 2010

2.2.3. The relationship wind power – wind speed – wind direction

The wind speed behavior of an area depends on the landscape and its surface or the local climate conditions. Even though the turbines are capable of turning according to the wind direction [29], there are studies in the literature that confirm that wind direction may be an important variable in the wind power prediction [30] [31].

In this part of the project, the variability of the relationship wind power – speed with respect to the wind direction will be explored and it will be determined if the last one must be included in the forecasting model.

Analyzing the wind rose for the wind farm area from Figure 17(a), it can be noticed that the most frequent wind direction is from the West and East, which is from the ocean or the sea. The length of the spikes indicates the frequency of the wind from each direction and the concentric circles represent a different frequency. It is observed that the high rated occurrences of wind speed (wind speed > 14 m/s) are coming only from the East, incorporating around 10% of the total wind speed manifestations of 2009.



(a) (b)

Figure 17. Wind rose for Cadiz province wind farm (a); a smooth surface of the wind power against wind speed and wind direction (b) for year 2009

The impact of the wind direction on the relationship wind power-wind speed is shown by plotting the output power against both wind speed and wind direction, illustrated in Figure 17(b). The surface indicates that for speed above 15 m/s the wind power tends to be higher for the wind blowing from the East or West.

Summarizing, it was found that the wind speed depends on the wind direction and even the relationship wind speed-wind power is influenced by the wind blowing direction.

2.3. Data preprocessing – outliers' study

In spite of the deterministic relationship shown in Figure 12, the empirical power curve obtained from the real data is different. In Figure 18, one can see an example of power curve for four turbines from Cadiz region wind farm, each point representing average wind speed against wind power sampled at 10 minute intervals.

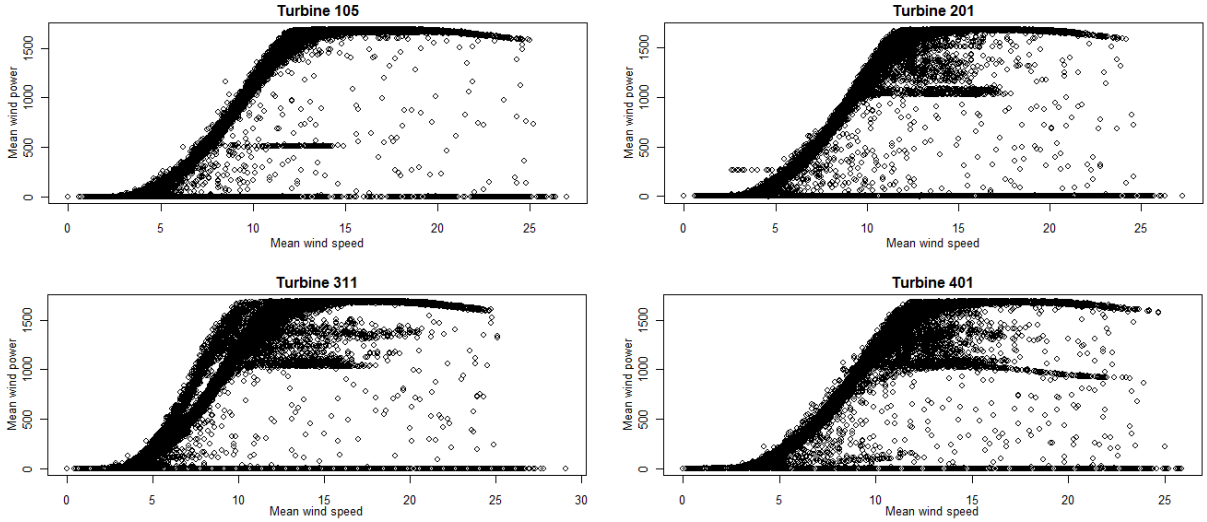


Figure 18. Empirical values of wind speed and power for turbines 105, 201, 311 and 401 from Cadiz region wind farm

Due to the large amount of data collected from each turbine, the measurements usually contain errors caused by sensors and malfunctions of the data acquisition system. These errors appear as outliers and missing values [32]. For example, the recorded wind speed should be in the range [0, 25 m/s] and the power should be in the range [0, 1670 kW], according to the turbine's manufacturer datasheet, following a logistic relationship. But, as shown in Figure 18, there are clearly some points outside the power curve, which are power losses and can be considered abnormalities.

According to Hawkins [33] "an outlier is an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". Scientists consider that removing outliers from the data is "cheating", but letting outliers may be considered the same because it can lead to invalid results [34].

On a power curve, the points with zero wind speed but non-zero wind power point out to a defective anemometer because it is clearly that only a non-zero wind speed measurement

is producing power. Furthermore, the points with zero power but large wind speeds indicate a defective turbine. Generally, except outliers, the majority of the points are ranged around the power curve [35].

Considering all above and also the strong arguments mentioned in [36] with respect to outliers' removal, the points that do not follow the power curve and are not in the range mentioned above will be considered outliers, studied and, then, if they are proved to be random, they will be removed from the study.

Therefore, in order to identify the outliers from the data, the next steps were carried on:

1. Extract negative wind speed and wind power data.
2. Extract data with zero wind speed and non-zero wind power.
3. Extract data with zero wind power and wind speed greater than 4 m/s.
4. Extract data from the left side of the nominal power curve, which is considered as over-rated, and the data from the right side of the nominal power curve, considered as under-rated, possibly because of a defective anemometer or turbine or many other reasons.

The empirical power curve with the cleaned data is illustrated in Figure 19 and it will be analyzed later in this chapter.

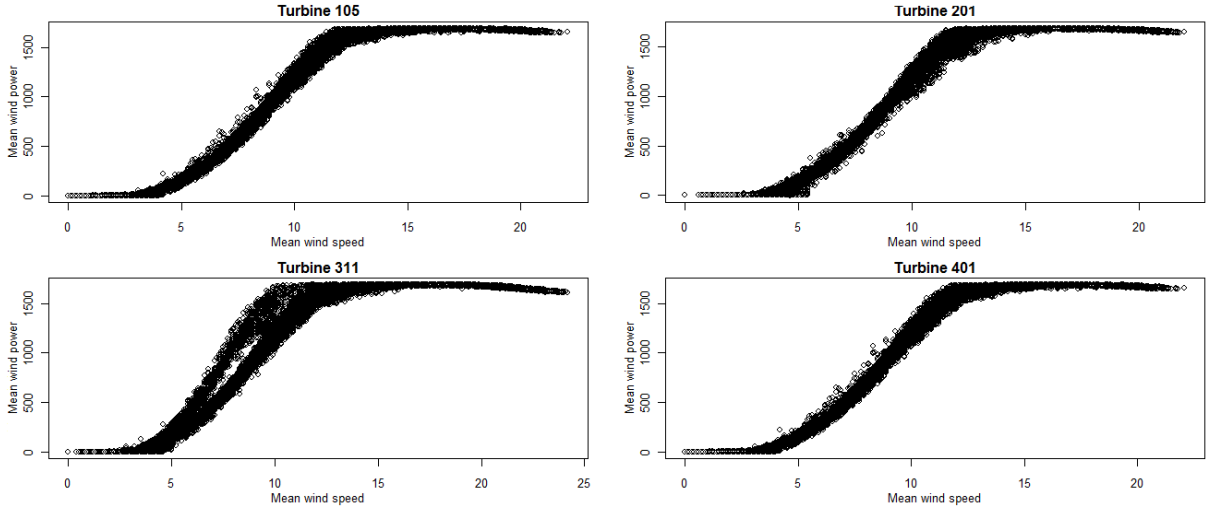


Figure 19. Empirical power curve with cleaned data for turbines 105, 201, 311 and 401

Before removing the outliers from the dataset, a study regarding their pattern was handled. In order to find some possible patterns, a monthly distribution of the outliers was realized with the idea of identifying some "rules" with respect to the outliers' positions in the dataset. It was encountered that the outliers do not depend on the month and a certain pattern cannot be found. Similar results were detected for the other turbines too.

An additional assumption that may be displayed, regarding the out of the pattern values, is that those observations may originate from the delay that the turbine have when it turns toward the wind. It was considered that there are many other parameters that can influence the behavior of the turbine and they cannot be all contained in the model. Furthermore this pattern of the observations can be due to some faulty devices or some errors in the data collection process. Considering all these, it was deliberated that, for the further analysis in this project, the outliers to be random and to be removed from the dataset.

Another topic that must be explored is the shape of turbine 311. Looking at Figure 19, one can see that the power curve is unusual. It presents two curves not one as in the other turbines. This effect may be due to different meteorological behavior or because of a faulty turbine. As the defective turbine was ruled out, in order to find the answer, the monthly behavior of the wind power at different levels of the wind speed was investigated. Analyzing the monthly behavior, it can be said that at the points that the curve is split in two, in December it is noticed a greater generated power which gets out of the pattern, as can be seen in Figure 20.

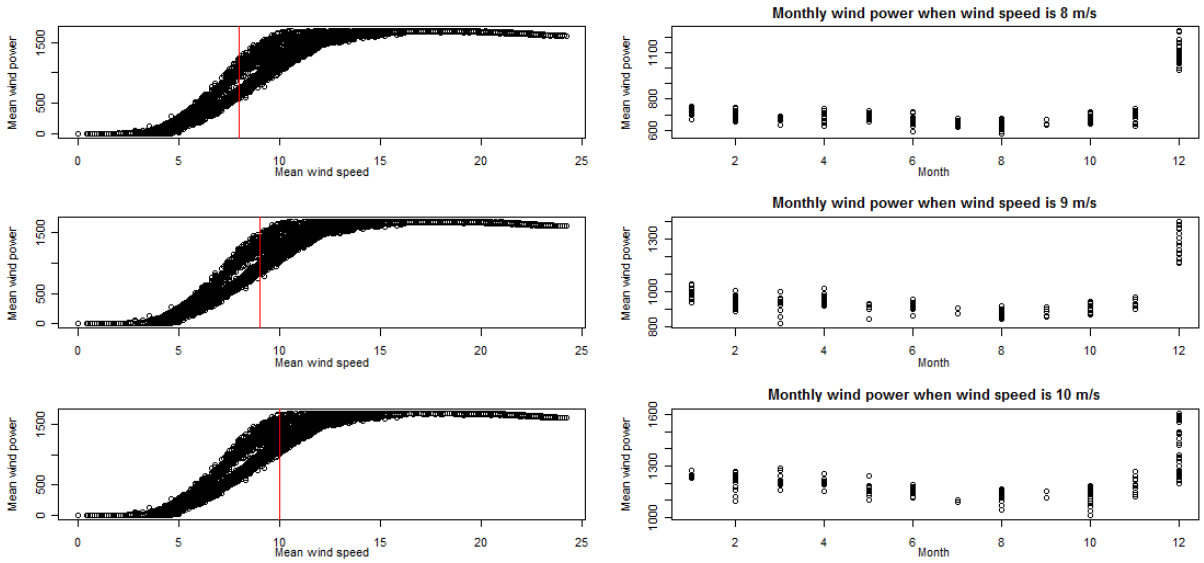


Figure 20. Monthly behavior of wind power at different level of wind speed for turbine 311

So, turbine 311 has two power curves, one that is related to the period January-November and one that is associated with the month of December. An illustration of the two power curves can be observed in Figure 21.

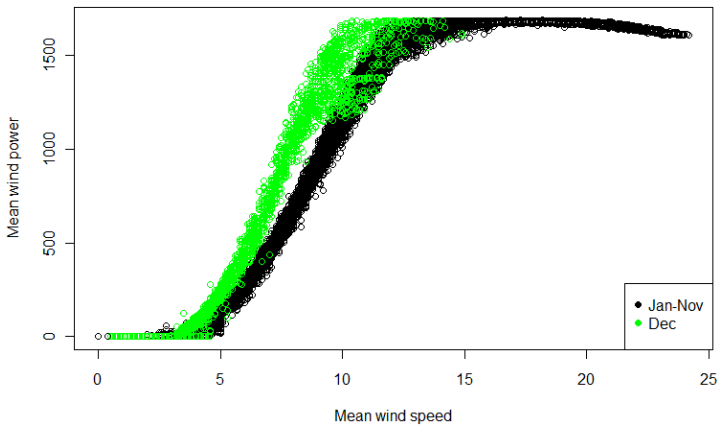


Figure 21. Power curve for turbine 311 depending on the month

When extending the research for all the turbines, the same behavior was found only at turbine 101. Turbine 101 and 311 are the two extremes from the north and the south and for reasons vague yet, they have the same behavior contrary to the others. One possible explanation could be that, while the blade sweep area of the turbine remains the same and the wind speed is identical for different intensities of the power, the only parameter from Eq. (6) that could change is the air density. It is likely that this behavior of the measurements may be

affected by the small differences in altitude, air pressure, temperature or humidity. Another possible argument is that these turbines have failures in functionality.

Chapter 3. Techniques for missing data imputation

This chapter surprises the procedures carried on for the missing values imputation. It begins with a briefly presentation of the missing value concept and some references found in the literatures concerning different methods of imputation. In the remaining part of the chapter, the wind data is analyzed and several procedures of imputation are tried and compared.

3.1. Missing data introduction and background

The missing values concept is essential in order to successfully work with datasets. If the missing values are not handled properly, inaccurate results may appear in the analysis.

Scientists and researchers encountered the issue of incomplete data many decades ago. First articles about missing data appeared around 1970 and the first book, Statistical Analysis with Missing Data (Little & Rubin), appeared in 1987 [37].

Most of the typical statistical methods yield results only for complete data, so the missing value treatment is important for a study. The quality of a complete value dataset depends on the method of treatment used. Therefore, before taking any action about the missings, one should find the source of the absent values and understand the process behind them. There were reported in the literature [38] [39] multiple mechanisms, such as:

- Missing completely at random (MCAR) – the missing values do no depend on the rest of the variables, the observed or missing ones
- Missing at random (MAR) – the missing values depend only on the observed values
- Missing not at random (MNAR) – the missing values depend on the rest of the variables from the dataset, the observed or missing ones

In practice, it is very difficult to fit the absent data in one of the above categories. Generally, one considers the data as being MAR and as many variable as possible are included in the model in order to decrease the bias.

Several approaches to handle missing data have appeared over the last years, each having its advantages and disadvantages. Like it is mentioned in [37] and [40], there are traditional methods of working with missings like likewise, pairwise deletion or mean imputation. Those methods can lead to biases in the parameters or the standard errors, due to

the exclusion of data from the analysis or the distortion that may be created among the values, even more if the number of missing values is high. More recent studies bring new improved approaches like multiple imputations (MI), estimation-maximization algorithm (EM) or maximum likelihood method (ML). Nevertheless, these methods can bring problems at the time of the implementation or interpretation. The algorithms behind them are not easy and there are not many software-tools that can handle them.

3.2. Missing data in the wind dataset

In this project, five approaches for estimating the missing values will be analyzed. It must be mentioned that the missing values number and pattern is identical for both wind speed and wind power.

The simpler procedure is the replacing of missing values with the mean or median (M/MD) of the variable, the last one being suitable for asymmetric distribution, like the case of wind speed or wind power data. The other approach applied here is the nearest neighbor averaging (NN). It identifies the observations with missing data and it fills them with the average value of the k nearest observations that have non-missing values.

The third method discussed is the replacement of missing data with the estimates of an autoregressive model, more exactly an ARMA(p,q) + GARCH(r,s). As the ARMA process cannot consider the heteroscedastic effects of the time series (spikes), the Generalized Autoregressive Conditional Heteroscedastic (GARCH) model will be introduced.

Another technique used here is a modified Expectation-Maximization algorithm for multivariate normal time series and finally, the last procedure used in this project for the missing value imputation is a nonparametric approach, namely a nonparametric regression by means of the Nadaraya-Watson estimator.

Considering that the output of a turbine is highly related to the wind speed, the estimation of the missing values for the wind power will be done by using the imputed value for the wind speed and the relationship between these two variables, which is the power curve.

In the next part of the chapter, a descriptive analysis of the missing data will be presented, along with a brief description of the concepts and methods used in the study. As the median and nearest neighbor imputations are simple methods, only the third, fourth and fifth approaches will be presented in detail. In the last part of the chapter, results and a comparison between the five methods will be displayed.

3.2.1. Description and patterns identification for the missing data

Before applying the four methods for estimating the missing data mentioned above, an analysis of the missings will be employed. The procedure of identification and treatment

of the missing values is the same for each turbine, even though they may have different behavior. So, due to time difficulties, in this chapter only turbine 105 will be analyzed.

A summary of the missing data is presented in Table 5. The amount of missing value is within 8%-18% of the total number of observations, which may be considered high and not to disregard.

With the idea of finding a pattern in the missing data, two studies were instigated. One is the analysis of the missing data by months and seasons. The seasons were considered as follows: Spring – March, April, May; Summer – June, July, August; Autumn – September, October, November; Winter – December, January, February. It was found that in autumn there are most missings in the data and during the summer the less. An illustration of missings can be found in Figure 22. Even though this pattern was found, a clear conclusion about the missings behavior could not be dropped from the data. There were not incidents with very high speed, for examples, in the autumn so that the turbines may break and stop functioning.

Table 5. Missing values summary for turbine 105

Season	# obs.	Missings	
		#	%
Total	52560	4392	8%
Spring	*13248*	*1050*	*8%*
Summer	*13248*	*238*	*2%*
Autumn	*13104*	*2052*	*16%*
Winter	*12960*	*1052*	*8%*

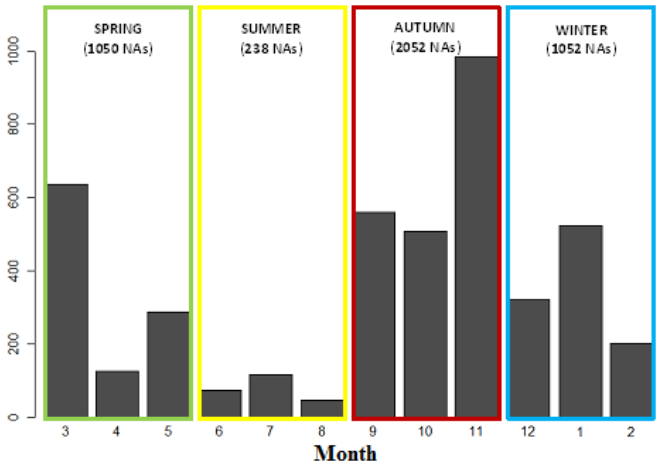


Figure 22. Missings pattern for 2009 data

The other study is related to the sequence structure of the missing data. Considering that the interval of measuring the wind is every 10 minutes, for every season it was computed the number of missing values for the following sequences: isolated observation, from 2 to 6 observations (missing from 10 minutes up to 1 hour), 7 to 12 (1 hour – 2 hours), 13 to 18 (2 hours – 3 hours), 19 to 144 (3 hours – 24 hours) and greater than 145 observations (more than 1 day). The results are presented in Table 6.

Analyzing the outputs, it was revealed that the majority of the missing values come from the last two interval patterns, meaning that 70% of the absent data are not quite random. This can mean that, either it was a software or hardware malfunction of the turbine or turbine components (ex. sensors) caused by some meteorological events, or just some failure of the turbine independent of the weather.

It can be observed also that the behavior of the missing values is almost the same in spring and winter, but on other hand in summer there are only short term interruptions. On the contrary on autumn, more than 80% of the missing data are from the last two intervals, more exactly it was found that there were 4 incidents that last more than 1 day, scoring 983 missing observations, and 15 breaks that persisted from 3 hours to 24 hours, totalizing 739 observations.

Table 6. Missing values pattern by intervals of time

Interval	10 min	10min – 1h	1h – 2h	2h – 3h	3h – 24h	> 24h	TOTAL
# obs.	1	2 – 6	7 – 12	13 – 18	19 – 144	≥145	
Spring	60	154	104	33	329	370	**1050**
Summer	42	115	16	43	22	0	**238**
Autumn	50	124	84	72	739	983	**2052**
Winter	75	226	93	0	277	381	**1052**
TOTAL	**227**	**619**	**297**	**148**	**1367**	**1734**	**4392**
%	**5%**	**14%**	**7%**	**3%**	**31%**	**39%**	

Continuing more deeply with the analysis on the missings, the other turbines from the wind park were investigated. It was considered only the missing pattern greater than 3 hours found at turbine 105, and it was discovered the same behavior at all the turbines from the park. This means that during those periods the entire park was shut down, most probably for maintenance.

3.2.2. ARMA combined with GARCH model for estimating the missing wind speed data

A short introduction into the autoregressive and moving average models will be presented, along with some details about the conditional heteroscedastic models. Autoregressive models are based on the idea that the value of an observation from a time series can be described as a function of p past values. The notation AR(p) denote an autoregressive model of order p and it is define in Eq. (8):

$$X_t = \phi_0 + \sum_{i=1}^{p} \phi_i X_{t-i} + \varepsilon_t \tag{8}$$

where $\phi_1, \dots, \phi_p$ are the parameters of the model, ϕ_0 is the intercept and $\varepsilon_t$is assumed to be a Gaussian white noise series ($\varepsilon_t \sim N(0, \sigma^2)$).

As an alternative to the autoregressive representation the moving average assumes that the white noise ε_t is combined linearly to form the observed series. The notation for a moving average model of order q is MA(q) and it is defined in Eq. (9):

$$X_t = \theta_0 + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{9}$$

where $\theta_1, \dots, \theta_q$ are the parameters of the model, $\theta_0$is the intercept (mean of the series) and $\varepsilon_t, \varepsilon_{t-1}$, ..., $\varepsilon_{t-q}$are white noise error terms, as mentioned above.

There are situations when a higher order model is needed to describe the data and an AR or MA is not sufficient. In these circumstances, one should apply an autoregressive moving average, which is a combination of an AR with a MA model so that the resulted model uses as fewer parameters as possible [41]. The notation for this model is ARMA(p,q) and it is defined in Eq. (10):

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \quad (10)$$

As ARMA models assume a constant variance, there are situations when the time series contains a lot of variability (volatility) and one should consider the variance in the model in order to have a good fit. These type of approaches are called autoregressive conditionally heteroscedastic models and were introduced by Engle in 1982. This field of volatility studies is very developed and it contains several model-families to study the volatility: GARCH, IGARCH, EGARCH, etc. [25], but for the interest of this project, it will be introduced only the generalized autoregressive conditional heteroscedastic model (GARCH) [42]. The notation for the model is GARCH(*r*,*s*) and it is defined in Eq. (11):

$$\begin{gathered} \varepsilon_t = \sigma_t \omega_t \\ \sigma_t^2 = \alpha_0 + \sum_{i=1}^{r} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2 \end{gathered} \quad (11)$$

where $\varepsilon_t$are the residuals (error term), $\omega_t$is a sequence of independent and identically distributed (iid) random variables with mean zero and variance 1 and α_i, β_j are the parameters of the model.

Now that the basic concepts were introduced, the procedure used for the estimation of the wind speed and wind power will be presented. In order to find the proper model, the initial series had to be differentiated once in order to become stationary. Then, the ACF and PACF of the differentiated series were plotted and it was found that the order of ARMA for the wind speed data is (0, 7), so the model is a MA (7). For the GARCH part of the model, the ACF and PACF of the squared ARMA's residuals were studied and some correlation was found. So, it was decided that the model used for the wind speed data will be an ARMA(0,7) + GARCH (1,1). This model was trained initially on the data until the first missing value, and then when the estimate arrived, it was considered as a new observation and the model was applied again on the new data and a new estimate was found. This procedure was carried on until all the missing values were replaced with an estimate. One can find more details about this estimation procedure in the next chapter, where the autoregressive models will be used for forecasting reasons.

3.2.3. EM algorithm for estimating the missing wind data

The EM algorithm was introduced by [43] and it computes the maximum likelihood estimates from incomplete data. The algorithm consists of two steps: one is the expectation and the other is the maximization step. This approach involves a series with incomplete data (either missing values or unobserved data points), which is split in two data series. One series, Y_t, contains the observed data and the other series, X_t, holds the unobserved or missing data. More exactly, the EM algorithm alternates between two steps. During the E-step (expectation), the expected value of the log likelihood function is computed, with respect to the conditional distribution of X_t given Y_t under the current estimate of the parameters.

During the M-step (maximization), the new parameter that maximizes the above function is found.

The EM algorithm will be applied by means of the *mnimput* function from the *mtsdi* package of R. As the function has three choices of imputing the missings, using ARIMA model, non-parametric smoothing splines or generalized additive model, the one used here is the spline method. The basic idea is to find a smooth function that captures the behavior of the data by minimizing the noise [44].

In order to have a good fit for the wind power estimated values, as it was said at the beginning of this chapter, one should take into account the relationship between wind speed and power, so in this case, the generalized additive models will be used. This concept was introduced first by Hastie and Tibshirani in 1986 [45]. A generalized additive model (GAM) is a generalized linear model (GLM) in which the linear predictor is given by a sum of smooth functions of the predictor variables. This means that the linear form of the predictors $\sum_i \beta_i X_i$ is replaced with the additive form $\sum_i f_i(X_i)$. Unfortunately, not any smooth functions are allowed in a model because they could produce overfitting. For this reason, the models are usually fit by using a smoothing parameter in order to control the fitting. The representation of the smoothing functions f_i can be done by using different basic approaches as regressions splines, cubic splines, smoothing splines or penalized regression splines and the smoothing parameter choice can be solved by using cross validation criterion (CV), generalized cross validation criterion (GCV), un-biased risk estimator criterion (UBRE) or others. As this topic is not covered by this project, more details can be found in [46] [47].

For the wind data, the *gam* function from the *"mgcv"* package of R was used. The distribution of the response variable (wind power) and the link function for the effects of the predictor variable (wind speed) on the response variable were selected, proving that the best fit of the model is with the Gaussian distribution and the log link, respectively.

3.2.4. Nonparametric approach for estimating the missing wind speed data

The methods explained so far are coming from the parametric statistics area. In this approach a nonparametric procedure will be used in order to estimate the missing values. Contrary to the parametric models, the nonparametric model structure is not fixed, it is determined from the data. These techniques are preferred because they make fewer assumptions and in most of the cases they are simpler and faster.

In order to estimate the missing values for the wind speed data, multivariate Kernel regression was used, namely the regression of the wind speed at time *t* on its past values *(t-1…t-d).* Firstly, some brief theoretical aspects will be introduced and then the imputation procedure will be described.

Kernel density estimation is a nonparametric technique for density estimation. It can be seen as a generalization of the histograms but with enhanced statistical properties. Kernel density estimators were first introduced in the literature around the middle of the 20th century [48] [49], but not very far after, studies about multivariate kernel density estimation appeared [19].

A multivariate kernel density estimator for the *d*-dimensional case is defined in Eq. (12):

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{12}$$

where *K* is a multivariate kernel function, $K: \mathbb{R}^d \to \mathbb{R}$ and $h=[h_1, h_2, \dots, h_d]$ is a vector containing the smoothing parameters, also called bandwidths. A multivariate kernel function can be defined by using the multiplicative kernels or the spherical kernels [50]. In the multiplicative kernels case, *K* is defined as:

$$K(u) = K(u_1) \cdot K(u_2) \dots K(u_d) = \prod_{j=1}^{d} K(u_j) \tag{13}$$

where *K* is a univariate kernel function. A kernel function is a real, integrable, non-negative symmetrical function such that:

$$\int_{\mathbb{R}} K(x)\,dx = 1; \qquad \int_{\mathbb{R}} xK(x)dx = 0; \qquad \int_{\mathbb{R}} x^2K(x)dx = k_2 > 0 \tag{14}$$

The most popular kernel functions are the Gaussian, Epanechnikov, Uniform, etc.

The alternative spherical kernels solution is to define *K* as:

$$K(u) = K(\|u\|) \tag{15}$$

where $\|u\|$ is the Euclidean norm of the vector *u*.

A regression can be seen as the relationship between a response variable Y_t and an multivariate explanatory variable X_t:

$$Y_t = r(X_t) + \varepsilon_i \tag{16}$$

where *r* is the regression function and ε is the error.

Considering that the conditional expectation is a regression function:

$$r(X_t) = E\left(Y_t \middle| X_1, \dots, X_d\right) = E(Y_t|X_t) \tag{17}$$

and replacing the multivariate densities by the kernel estimates, the regression *r* can be estimated by the estimator $\hat{r}$, called Nadaraya-Watson estimator [51]:

$$\hat{r}(x) = \frac{\sum_{i=1}^{n} Y_i \cdot K\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)} \tag{18}$$

An important point in the kernel regression can be the computation of the bandwidths. These aspects will be discussed in detail in the next chapter.

Now that the theoretical part was presented, in the next lines of this section, the procedure for the missing values estimation of the wind speed will be discussed. As in section 3.2.2., the nonparametric approach for missing imputation is a recursive algorithm. The estimation is one-step ahead and each time a new prediction arrives, it is included in the training data. So, first training set will be the observations until the first missing value, and then by applying the presented nonparametric approach, the first missing value is estimated. As a next step, this value replaces the missing spot in the data. This algorithm continues until all the missing values are estimated. It must be mentioned that the bandwidth used was a unique one, *h=0.04* for missing intervals greater than 30 observations and $h = 1/(\#\ of\ missing\ observations)$ otherwise, and there were consider 3 past values of influence (*d=3*). The multivariate kernel function was defined using the spherical kernels.

3.2.5. Results

As it was mentioned at the beginning of the chapter, in this section a comparison between the five methods of imputation will be handled and the best method will be pointed so that is can be used in the rest of the study.

First of all, the wind power imputation method will be detailed. Because the output power is related to the wind speed, its imputation was not handled using the above presented methods. Taking into consideration the time variable, the empirical power curve, see Figure 19, and the wind speed data imputed, the missing values for the wind power were estimated as it follows:

- Store the time position of the missing value from the wind power time series $\rightarrow t$
- Store the wind speed from the time position $t \rightarrow s$
- Store all wind power observed values that have a wind speed in the interval $[s - 0.25\ m/s, s + 0.25\ m/s] \rightarrow p$
- Replace the missing value from the wind power with the average of p

The results for the five imputations can be noticed in Figure 24.

In order to have a clearer image of the outputs, a sample of the first 4000 observations will be taken. As one can see, the estimations yielded by the median, k-nn and ARMA approaches are not a good fit. This output was expected somehow, because the interval with missing values is relatively high at some time positions. The EM results are better when looking at the entire time series, but when focusing on the 5000 sample, it can be seen that the imputation is too smooth and it does not characterize the real state. The last plot representing the nonparametric estimates shows some good results. It takes into account the variability of the time series, not being so smooth as the EM algorithm results.

As the first four method of imputation were ruled out for the wind speed, only the nonparametric results will be presented for the wind power, see Figure 23.

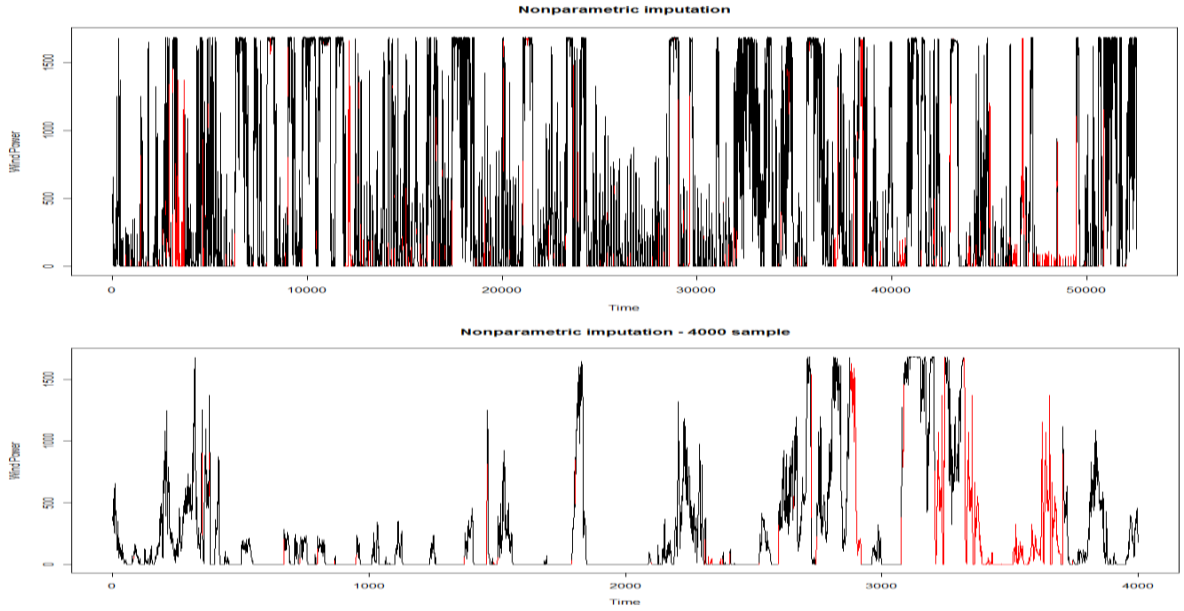


Figure 23. Results of wind power missing imputation for turbine 105

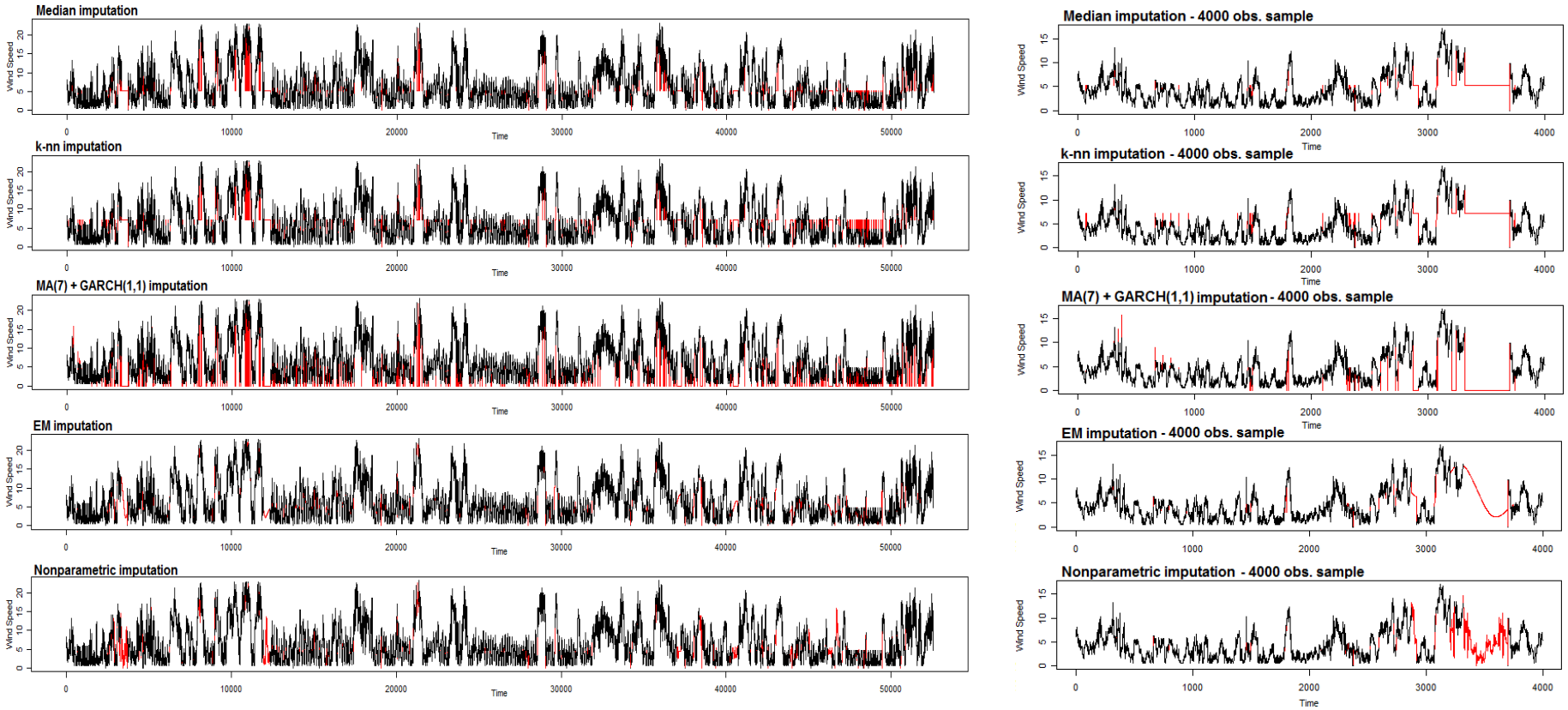


Figure 24. Results of wind speed missing imputation for turbine 105

Chapter 4. Short-term time series forecasting

In the first part of this chapter some basic concepts about time series and their characteristics are introduced, followed by some short literature overview about wind speed and power prediction. Then four methods of wind speed forecast are presented, consisting in a parametric, nonparametric and two semiparametric approaches. The results are presented by comparing the root mean squared error. In the last part of this chapter the economic impact of a bad imputation will be displayed.

4.1. Time series and their characteristics

Over the last decades the sequential collected data has increased in popularity and nowadays it became very common and used in everyday decisions. According to [52], "when a variable is measured sequentially in time over or at a fixed interval, known as the sampling interval, the resulting data form a time series". The list of fields in which time series are studied is boundless. One could find time series analysis in business, meteorology, medicine, agriculture, engineering and many other fields. Time series analysis is proven to be very useful; on one hand to model the behavior of the collected data, meaning to study the past performance, then to predict future values of the series based on past observations [53].

Time series data can be either univariate or multivariate. The univariate time series consists of single observations documented chronologically over equal time increments. The multivariate case contains two or more univariate time series, with the idea that their relationship may play an important role in a model [25].

As it was mentioned above, there are two main and distinct interests in the time series field: one is the analysis and the other the modeling of the data. While time series analysis seeks for patterns between the observations, the time series modeling uses a model based on theoretical and mathematical assumptions in order to explain a behavior and to estimate future values. In the raw data series one can find complex and random processes, but one would expect some dependence between close observations in time and less dependence between those very far from each other.

A stochastic process is told to be strictly stationary if it is time invariant, this means that given the time $t_1, t_2, \ldots, t_n$ the joint statistical distribution of the sample $X_{t_1}, X_{t_2}, \ldots, X_{t_n}$ is

the same as the joint statistical distribution of a time shift sample $X_{t_{1+k}}, X_{t_{2+k}}, \dots, X_{t_{n+k}}$. These assumptions are very strong while taking into account real data, so a weaker definition is introduced, namely the weak stationarity. Weak stationarity means that the mean and variance of a stochastic process will be constant and the autocovariance between X_t and X_{t+k} will only depend on the lag k [25].

One other important subject in the context of time series analysis and modeling is the correlation between the observations, more precisely the autocorrelation function (ACF) and partial autocorrelation function (PACF). Taking into account that a series is weakly stationary, the ACF at lag k is the coefficient that measures the linear dependence between two observations, X_t and X_{t+k}, returning a value around zero when they are uncorrelated. It is defined as:

$$\rho_k = \frac{cov(X_t, X_{t+k})}{\sqrt{var(X_t)var(X_{t+k})}} = \frac{cov(X_t, X_{t+k})}{var(X_t)} = \frac{\gamma_k}{\gamma_0} \tag{19}$$

Taking into account the above characterization, it is consider that $\rho_0 = 1$.

The partial autocorrelation function at lag k is just the autocorrelation between X_t and X_{t+k}, where the observations between them, $X_{t+1,\dots,}X_{t+k-1}$, are considered constant. The PACF is defined only for lags greater or equal to 2, because the first element will be exactly the ACF at lag 1 (ρ_1).

The graph that represents both ACF and PACF is called correlogram. The intensity of the coefficients is usually represented by some lines or blocks having their height in relation with the coefficient value at that lag. Besides these values, the graph also contains the upper and lower bound for autocorrelation, with a significance level α. When interpreting the graph, one can tell that if a spike is outside the bounds the null hypothesis that there is no autocorrelation at a given lag is rejected at a significance level of α. One can find an example at the end of this section.

Besides being a useful tool in studying the randomness of a time series, these coefficients are important in the model identification process. Some basic econometric models were introduced in the previous chapter in Section 3.2.2., namely the autoregressive model (AR), moving average model (MA) and the autoregressive moving average model (ARMA), all of them having at the base a stationary process. Relating to the ACF and PACF coefficients, one can identify the orders of the above mentioned models by analyzing the coefficient values. The order p of the AR model can be found by using the sample PACF while the order q of the MA model can be established by means of the sample ACF. For instance, looking at a PACF plot, if the p-th lag is above the significance level and all the next lags are close to zero then an AR model of order p can be fitted to the data. The same process can be followed for finding the order q of an MA(q) model, but this time looking at the ACF plot. An issue in time series modeling is that the above presented theory about stationarity does not hold in the real life data. The data may contain trend, variability or even seasonality and most of the autoregressive models cannot work with these features. One straight forward method for verifying if a time series is or is not stationary, is the graphical representation. In order to have a clearer image, the wind data studied in this project will be analyzed. In Figure 25 one can find the wind speed observations from 2009, for turbine 105.

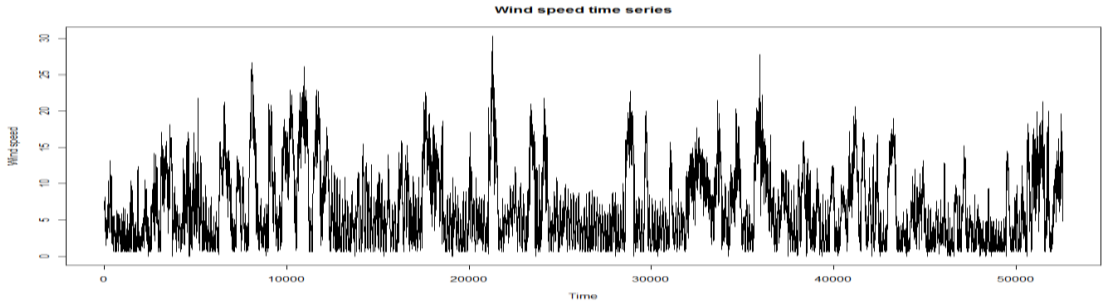


Figure 25. Time series plot of 2009 wind speed for turbine 105

According to the definition of a stationary time series, the data should fluctuate around a constant mean, independent of time, and the variance of the fluctuation should stay constant over the time. Looking at the wind speed plot, it is clear that the series is not stationary. Besides the series plot, one can verify the stationarity by looking at the correlogram. If a series is stationary, the autocorrelations drop to zero quick, while in the non-stationary series they decrease to zero very slow [54]. Taking a look at the correlogram from Figure 26 for the wind speed data, one more time it appears that the time series is not stationary.

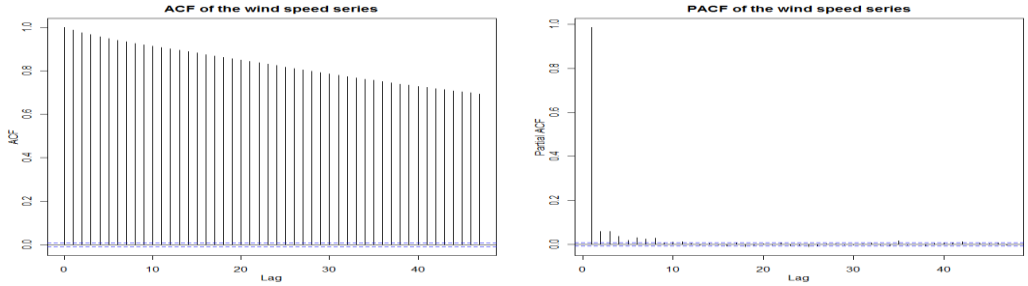


Figure 26. Correlogram of wind speed 2009 data for turbine 105

Since some models require data stationarity it is important to transform the series into a stationary one. There are several approaches that handle this, but the most used are the Box-Cox transformation [55], which deals with the not-constant variance by taking logarithm of the series, and the Box-Jenkins transformation [41], which recommends the differencing for achieving stationarity.

Once that the model is set, one can use that model to estimate future values. The prediction can be done one-step ahead or multiple-steps ahead. The one-step ahead case, is based on the fact that the observations $X_1, X_2, \dots, X_t$ are known and we want to estimate one value, X_{t+h}, where h=1,2,…. In the multiple-steps ahead case, having the same past observations, we want to estimate, not only one, but multiple values in the future. Each of these methods has advantages and drawbacks. For instance, the forecasting error is smaller in the one-step ahead case, because the model is based only on the past values without including the estimates, as in the multiple-step ahead case. On the other hand, the multiple-step ahead forecasting is more useful in the prediction area, it can offer an overview of the time series on a definite horizon.

Having a short introduction to time series modeling and forecasting procedures, in the next part of this chapter a study of the wind speed and power forecasting will be initiated.

4.2. Forecasting wind speed and wind power

4.2.1. Introduction and literature overview

During the last years, an important attention was directed to the wind power forecasting output. As it was presented in the first chapter, wind energy technology had developed rapidly in Europe and it is in continuous increasing. Therefore, a good prediction of the wind behavior is crucial. One of the main difficulties encountered by power system operators is the unpredictability and variability of wind farms output. This has, besides the operational issues, some important financial implications.

The wind power production depends on the wind speed. Taking into consideration that the wind speed is influenced by many atmospheric conditions like temperature, air density, pressure, it fluctuates a lot and it is very difficult, or even impossible, to include all the factors in a forecasting model.

Nowadays many procedures that try to deal with the challenging prediction of wind speed and wind power are known. In the literature, it is usually reported that wind power prediction is done by converting, through the power curve, the wind speed predicted time series, rather than direct time series modeling of wind power [56].

Numerous methods that try to forecast the wind speed can be found. The simplest of them, the persistence model [57], calculates the estimates of tomorrow relying on the today's value, more precisely, the tomorrow value will be equal to today's value. This model becomes more inaccurate as the forecasting horizon increases, but on the other hand it is simple to implement. Classical, but very popular methods are the time series models like autoregressive moving average [58], or more complicated approaches like Kalman filters [59] or Bayesian methodologies [60]. As the technology developed a lot and more data is collected and the traditional methods of forecasting proved to be very difficult to use on large datasets, advanced methods like artificial neural networks [61] [62] or support vector machine [63] appeared to yield very good results.

4.2.2. Wind speed forecasting methodology

Three approaches will be used in this study for the wind speed forecasting: a parametric, a semi-parametric and a nonparametric model. These three methods will be discussed in the following parts. The procedures will be applied for the 2009 wind speed data of turbine 105, the other turbines' prediction being done similarly.

a) Parametric approach

For the parametric approach ARMA and GARCH models were used. Although the two models were introduced in Chapter 3, their definitions will be presented one more time for clarity. Therefore, an ARMA model is defined in Eq. (20),

$$X_t = c + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} + \varepsilon_t \tag{20}$$

where c is the intercept, $\phi_1, \dots, \phi_p$ and $\theta_1, \dots, \theta_q$ are the parameters of the AR and MA part, respectively, and ε_t is assumed to be a Gaussian white noise series ($\varepsilon_t \sim N(0, \sigma^2)$).

In addition a GARCH model is defined in Eq. (21):

$$\varepsilon_t = \sigma_t \omega_t$$
$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{r} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{s} \beta_j \sigma_{t-j}^2 \tag{21}$$

where $\varepsilon_t$are the residuals (error term), $\omega_t$is a sequence of independent and identically distributed (iid) random variables with mean zero and variance 1 and $\alpha_1, \dots, \alpha_r$ and $\beta_1, \dots, \beta_s$ are the parameters of the model.

The first step in finding the model, according to Box-Jenkins methodology, is the identification phase. In this phase one should prepare the data and select the model orders. The input time series for an ARMA model must be stationary, so in order to check the stationarity assumption, the time series itself and the ACF/PACF plots will be studied. As already was determined in the previous section, see Figure 25 and Figure 26, the wind speed series is not stationary and, as first step, a first order differentiation was applied. The new differenced series along with its correlogram can be observed in Figure 27 and it can be stated now that the series is stationary.

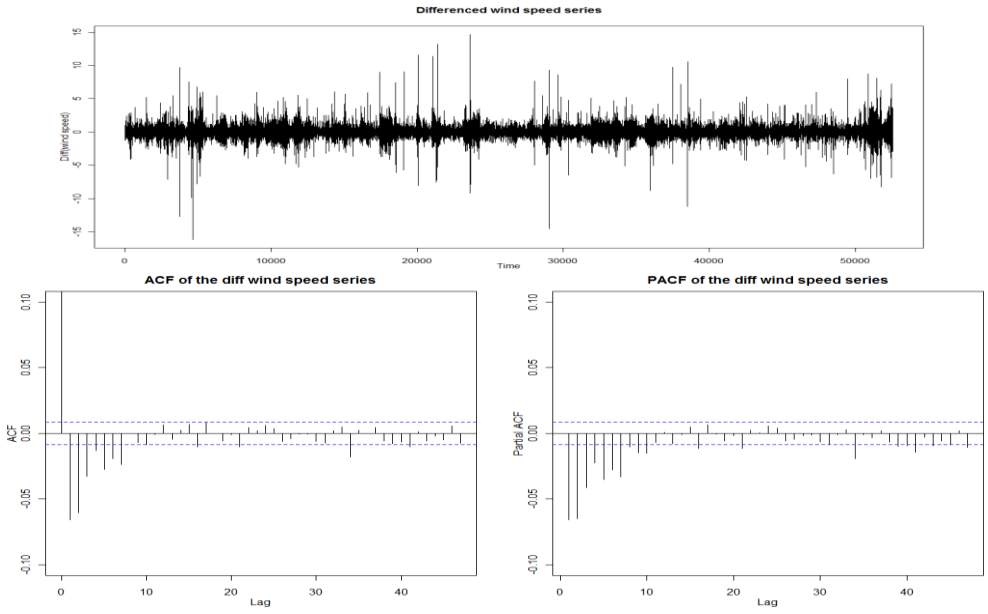


Figure 27. First order differenced wind speed time series and the ACF/PACF plot for turbine 105, year 2009

Once the data are prepared for modeling, one can establish the models' orders. Looking at the ACF and PACF plots from Figure 27, some significant lags can be observed

until lag 7 in the ACF and lag 10 in the PACF. As first option, a MA model of order 7 will be used.

The next phase in the time series modeling is the parameters estimation. Using the *arima* function from R package *tseries*, the model parameters were computed and they all proved to be significant, as it can be seen in Table 7:

Table 7. MA(7) model coefficients for wind speed data

	ma1	**ma2**	**ma3**	**ma4**	**ma5**	**ma6**	**ma7**
coef	-0.0771	-0.0683	-0.0387	-0.0191	-0.0318	-0.0224	-0.0257
s.e.	0.0044	0.0044	0.0044	0.0044	0.0044	0.0043	0.0044

The model diagnosis was done by analyzing the residuals. If the model is adequate, the ACF/PACF plots of the model's residuals and squared residuals should appear like a white noise, meaning that it must be no correlation between the observations. The residuals plots are illustrated in Figure 28. As it can be observed, the squared residuals are correlated, meaning that a GARCH model is needed to model the volatility.

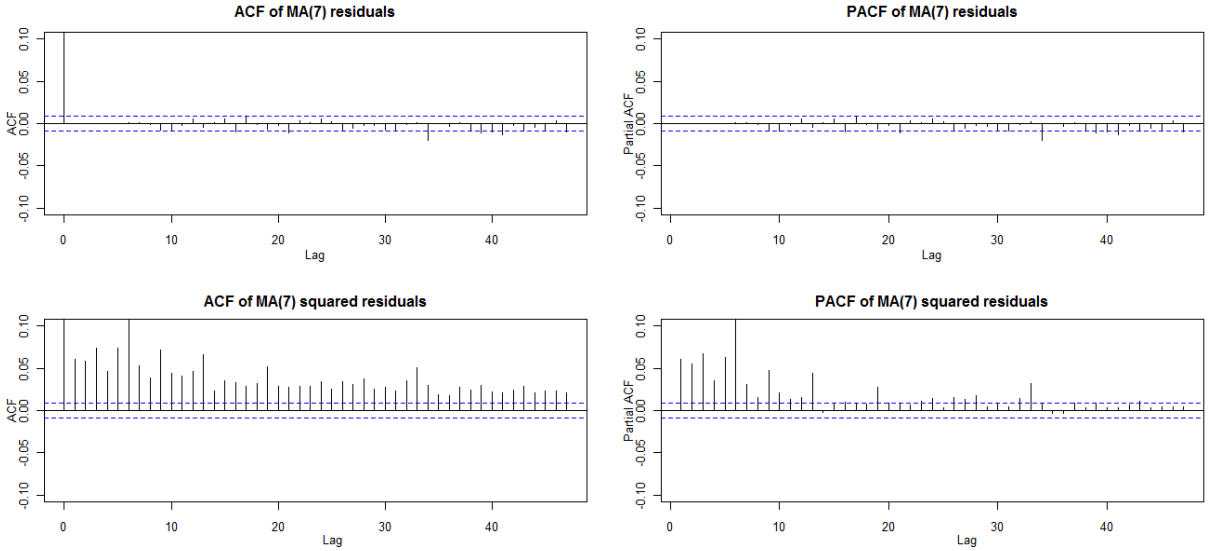


Figure 28. ACF/PACF of the MA(7) residuals (top) and squared residuals (bottom)

After modeling the residuals with a GARCH(1,1) using the function *garch* from the R package *tseries*, all coefficients resulted to be significant. The new ACF/PACF plots for the new model residuals are plotted in Figure 29.

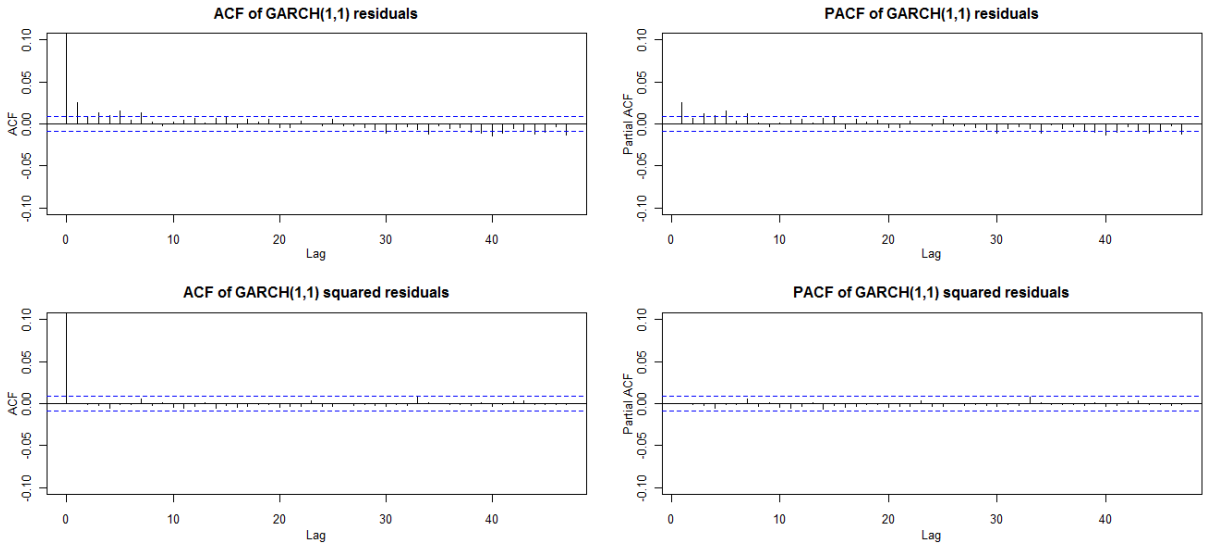
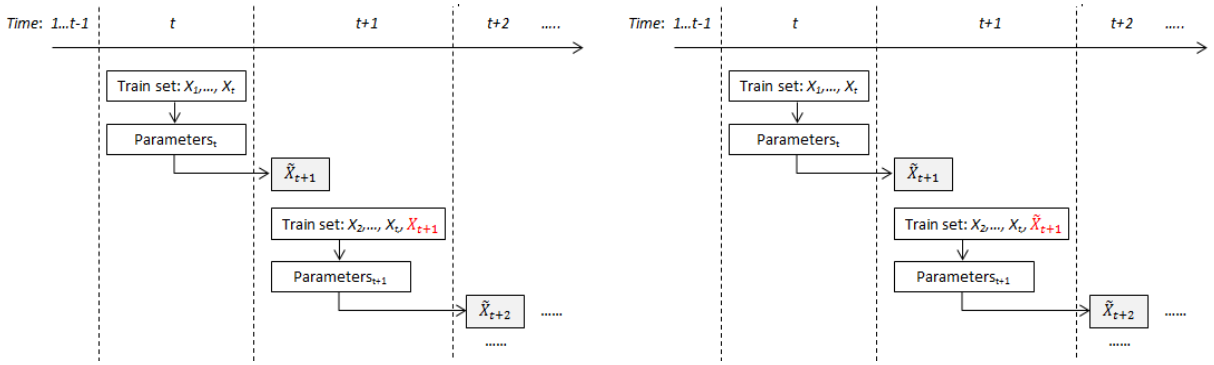


Figure 29. ACF/PACF of the GARCH(1,1) residuals (top) and squared residuals (bottom)

The ACF plot shows that there are still some correlated observations because the first lags are significant. Therefore, some new models were tried, like AR(10)+GARCH(1,1), ARMA(10,3)+GARCH(1,1) or ARMA(5,7)+GARCH(1,1), but almost the same results were returned. Taking into account that the autocorrelation is not so high and there are few significant lags, it was decided that the model for the wind speed data is a MA(7)+GARCH(1,1).

Once the model was identified, one can pass to the next phase, the prediction. In this investigation, two types of prediction were made, a one-step ahead and a multiple-step ahead (long-run). The procedures' steps are illustrated in Figure 30. The difference between them is that the one step ahead prediction is estimating the *t+1* value and then in order to estimate the *t+2* value, one should wait for the *t+1* value to be observed. While, in the multi-step prediction algorithm, the *t+1* estimated value is added to the initial training set and, using this new set, the *t+2* value is predicted.



a) One-step ahead approach *b) Multi-step ahead (long-run) approach*

Figure 30. One-step and multi-step ahead prediction algorithms in general

There are several advantages and drawbacks for each of the above mentioned algorithms, but they will be presented in the results section.

b) Nonparametric approach

Forecasting with nonparametric models was used in many fields such as finance, agriculture, energy and others [64] [65]. In this investigation, the Nadaraya-Watson estimator [51] will be used in order to estimate the wind speed. This method does not require any constraints on the wind speed probability distribution, it only assumes that this one exists and it is continuous and differentiable.

As stated in [66], for nonparametric models it is not necessary for a series to be stationary. So, in this approach, the series that will be model is the initial one.

This approach was used for the missing imputation procedure, so some of the theoretical concepts were already introduced in Section 3.2.4, but the important aspects will be revised one more time here for clarity.

The main idea in the nonparametric modeling is that, having the relationship $Y_t = r(X_t) + \varepsilon_i$ between two variables, X_t and Y_t, no assumptions are made about the function *r*. Therefore, the model will not have precise structure, it will be constructed using the data. It was shown in Section 3.2.4. that an estimator for the function *r* can be the Nadaraya-Watson estimator, presented in Eq. (22):

$$\hat{r}(x) = \frac{\sum_{i=1}^{n} Y_i \cdot \prod_{j=1}^{d} K_{h_j}(x_j - x_{ji})}{\sum_{i=1}^{n} \prod_{j=1}^{d} K_{h_j}(x_j - x_{ji})} \tag{22}$$

where $K_{h_j}(x_j - x_{ji}) = \frac{1}{h_j} K\left(\frac{x_j - x_{ji}}{h_j}\right)$ and *d* shows the number of observations to take into account from the past, called sometimes dimension or lag. Furthermore, *K* is a univariate kernel function and h_j is the smoothing parameter. Since the kernel function and the smoothing parameter are two important points in the computation of the regression estimator, special attention will be directed towards them.

A kernel function is a weighting function used in the nonparametric estimation techniques. There are several types of kernel functions, but the most commonly used are the uniform, triangle, Epanechnikov, quartic, tricube, triweight, Gaussian, and cosine [67]. In this study the kernel function used is the Gaussian one, presented in Eq. (23):

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2h^2}} \tag{23}$$

In other words, a kernel function is weighting the similarity between two vectors. In the wind speed estimation case, the kernel function is simply quantifying which sequence of *d* observations from the past is more related to the last *d* observation, and for each measure it gives a weight. These weights are then assigned to the response vector and the new estimation computed.

Mathematically speaking, the Nadaraya-Watson estimator presented in Eq. (22) can be rewritten as:

$$\hat{r}(x) = \sum_{i=1}^{n} Y_i \, w_i \tag{24}$$

where $w_i = \frac{\prod_{j=1}^{d} K_{h_j}(x_j - x_{ji})}{\sum_{i=1}^{n} \prod_{j=1}^{d} K_{h_j}(x_j - x_{ji})}$.

If the kernel function is replaced with the Gaussian kernel function, the weight's formula becomes:

$$w_i = \frac{e^{-\sum_{j=1}^{d} \frac{(x_j - x_{ji})^2}{2h_j^2}}}{\sum_{i=1}^{n} e^{-\sum_{j=1}^{d} \frac{(x_j - x_{ji})^2}{2h_j^2}}} \tag{25}$$

The choice of the smoothing parameter is very important in the context of a kernel function because it may influence the performance of the forecast method. In order to have a clearer overview of this significance, in Figure 31, one can find a sample of 1000 observation from the wind speed time series (black line) with a bandwidth of 0.6. In the same plot one can find three estimates for the sample using a bandwidth of 3 (red lines), 50 (green line) and 300 (blue line). When the smoothing parameter is small, the case of the red line, it almost represents the data, while when h is too big it appears the over smoothing effect yielding almost constant estimates.

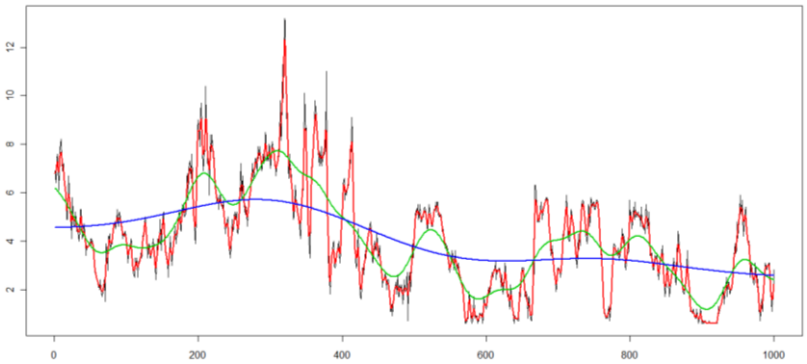


Figure 31. Comparison of different kernel density estimates constructed with different bandwidths

One of the most common method for computing the bandwidth is the normal reference rule. The idea behind this procedure is that the bandwidth becomes optimal when the mean integrated squared error (MISE) is minimized. The MISE measures the accuracy of an estimator $\hat{f}$ over a range x, in relation with its variation and bias [67], as shown in Eq. (26):

$$MISE = E\left(\int \left[\hat{f}(x) - f(x)\right]^2 dx\right) \approx \frac{h^4 \sigma_K^4}{4} R(f'') + \frac{1}{nh} R(K) \tag{26}$$

Where $R(f'') = \int [f''(x)]^2 dx$ and $R(K) = \int K^2(w) dw$

The MISE becomes minim when the bandwidth is:

$$h^*(x) = \left(\frac{R(K)}{n\sigma_K^4 R(f'')}\right)^{\frac{1}{5}} \tag{27}$$

The moment when f'' has to be calculated, a problem arises because the second order derivative cannot be computed due to the fact that the density function is unknown.

A very common alternative of computing h^* is to assume that f is a normal density function, Eq. (27) becoming then:

$$h_{NRR} = \left[\frac{4}{(d+2)n}\right]^{\frac{1}{(d+4)}} \hat{\sigma} \tag{28}$$

where $\hat{\sigma}$ denotes the sample standard deviation and *d* is the number of dimensions, or in other words how many past observations to take into account when looking for similarities in the past. Even though this assumption does not seem very suitable in a nonparametric modeling, it was proven to give competitive results in the unimodel distributions and it is very easy to implement [67]. Due to the fact that the standard deviation $\hat{\sigma}$ is sensitive to the outliers, one may want to use instead the interquartile range (IQR).

A more robust technique of bandwidth computation reported in the literature is the cross-validation method. In this approach, the optimal bandwidth is found when the integrated squared error (ISE) is minimized. According to [67], considering that the ISE is defined as:

$$ISE = \int \left[\hat{f}(x) - f(x)\right]^2 dx = \int \hat{f}^2(x)dx - 2\int \hat{f}(x)f(x)dx + \int f^2(x)dx \tag{29}$$

the optimal bandwidth can be computed with the estimator defined in Eq. (30):

$$h_{CV} = \int \hat{f}^2(x)dx - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i}(x_i) \tag{30}$$

where $\hat{f}_{-i}$ is the density estimator after removing the i^{th} observation [67].

The best choice of the smoothing parameter is still an ongoing research and the opinions are divided. In this investigation the smoothing parameter was fixed using two methods. One method is using the normal reference rule procedure from Eq. (28) and the second one, uses some "visually"/by trial measure like 0.01, 0.1 and 1. In the one-step ahead procedure it was used only the normal reference rule method.

The prediction algorithm is exactly the same as the one presented in the parametric approach, Figure 30, with exception of the parameters' setting step, which in this approach it does not appear.

c) <u>Mixt approach</u>

This last method is a combination of the above presented approaches, resulting two new semiparametric models. The idea is to study the behavior of the nonparametric approach on the parametric residuals, on one hand, and also to apply the parametric models on the nonparametric residuals. In other words, after fitting the raw data with the parametric and nonparametric model, respectively, the residuals in each case are computed. These residuals are then modeled with the nonparametric and parametric model, respectively. In the end, the mixt estimation is the sum of the parametric/nonparametric estimations and the residuals' estimations.

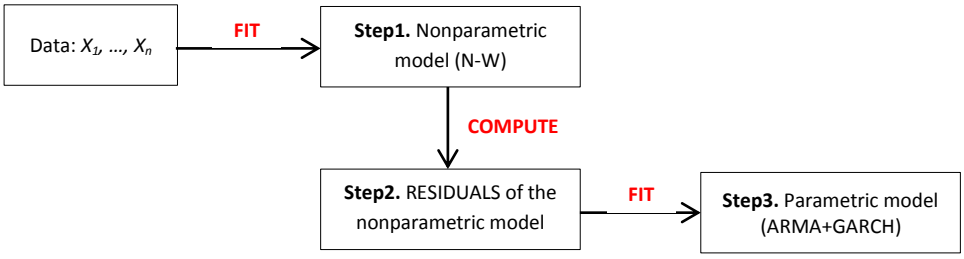
More precisely, the prediction will be decomposed as the sum of an optimal linear prediction and an optimal (nonlinear) prediction, for each of the two semiparametric models, as follows:

- Mixt 1 : $E(X_t|X_u, u<t) = E(X_t|X_u, u<t) + EL(\varepsilon_t|\varepsilon_u, u<t)$ (31)
- Mixt 2 : $E(X_t|X_u, u<t) = EL(X_t|X_u, u<t) + E(\varepsilon_t|\varepsilon_u, u<t)$ (32)

The optimal linear prediction in this case is an ARMA+GARCH model estimates and the optimal (nonlinear) prediction is the Nadaraya-Watson regression.

For a better comprehension, an illustration of the algorithm is shown in Figure 32.

a) Mixt1 semiparametric model



b) Mixt2 semiparametric model

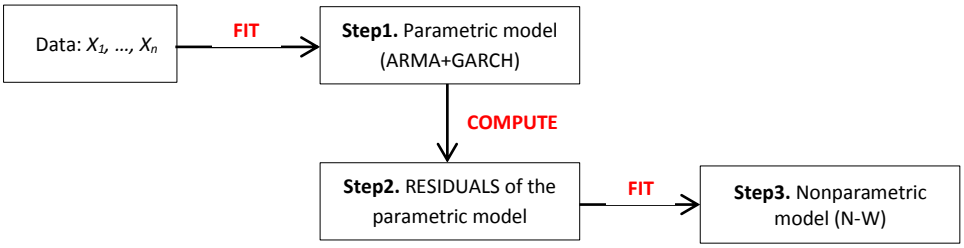
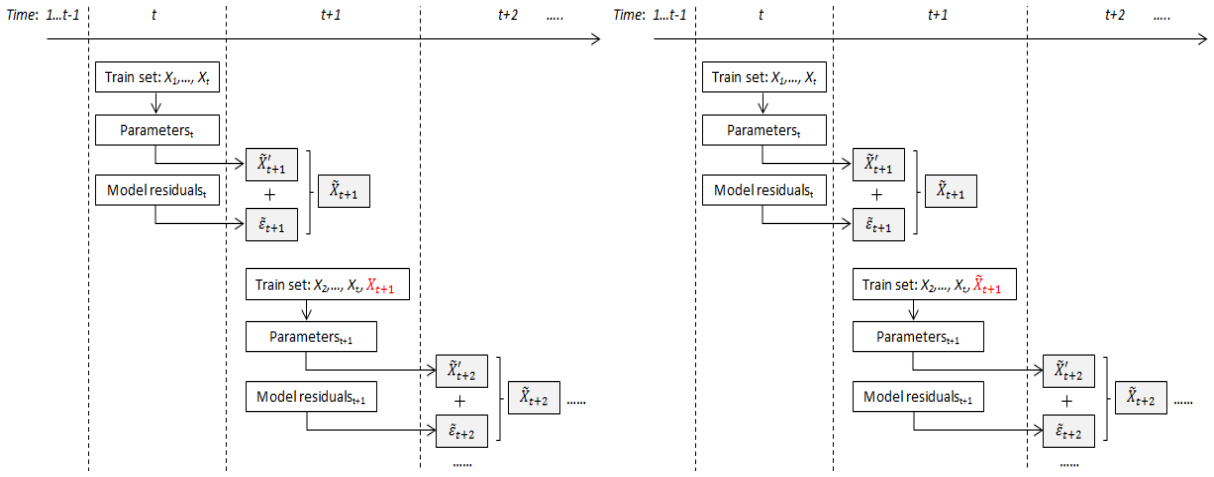


Figure 32. Semiparametric model's algorithm

The methodology for the model fitting is the same as the one presented in a) and b), for the parametric and nonparametric models, respectively. There is one difference, though, in Step 3 of model Mixt 1, at the parametric modeling of the residuals. In this case, after conducting the same actions describe in a) for the model identification and validation, the model resulted to be a MA(1)+GARCH(1,1).

The algorithm for the one-step and multi-step predictions for the mixt models are similar to the one presented in a), with the difference that in this case some extra computations are done, namely the modeling of the residuals. The algorithm is presented in Figure 33.



a) One-step ahead approach *b) Multi-step ahead (long-run) approach*

Figure 33. One-step and multi-step ahead prediction algorithms for the mixt approach

The idea behind this method is to improve the linear parametric estimation by using a nonparametric estimation and vice versa. Considering that one cannot take into account a large number of lagged values from the past process to describe the dynamics of the time series due to the curse of dimensionality, this mixt model can bring better results by combining the linear and nonlinear methods.

4.2.3. Wind power forecasting methodology

In this investigation the wind power will be estimated taking into account its strong relationship with the wind speed. This relationship, as discussed in Section 2.2.2, is represented by the empirical power curve. The power curve of turbine 105, that is the object of the study in this chapter, is shown in Figure 34.

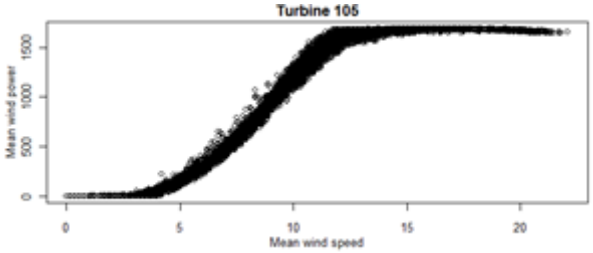


Figure 34. Wind power curve for turbine 105

Hence, once the wind speed is estimated, the wind power predictions will be established through the power curve. So, the prediction for the wind power series Y at time $t+1$ (Y_{t+1}), when there are data until time t ($Y_1,\dots,Y_t$) and the wind speed series X is estimated until lag h ($X_1,\dots, X_t, X_{t+1},\dots X_{t+h}$), is done by average all the wind power values from $Y_1,\dots,Y_t$ which have a wind speed equal to X_{t+1}.

4.3. Forecasting accuracy and confidence intervals

4.3.1. Measure of accuracy

The forecasting methods presented in Section 4.2. are used to make one-step ahead and multi-step ahead predictions. The models will be trained on a sample of 360 days, from 1st of January 2009 until 26th of December 2009 and then estimation are made for a period of up to five days, from 27th of December 2009 until 31st of December 2009.

The accuracy of the results will be tested by computing the root mean squared error, defined in Eq. (33):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(obs^2 - prev^2)}{n}} \tag{33}$$

A lower RMSE implies that the forecast is more accurate, whereas a high RMSE value implies less accuracy.

The results were then compared with those of the persistence model, where the value of the estimate is exactly the nearby value.

The dimension in the kernel regression was set up to 3, meaning that, in this study, up to three past observations will be taken into consideration for the nonparametric modeling. There are references in the literature about a mathematical computation of kernel's proper dimension [68] [69], but this subject will be consider as further research.

The results presented in Table 8 and Table 9 are for the 1-day ahead and 5-days ahead prediction, for both one-step and multi-step procedures. The smaller RMSE in each case is underlined.

Table 8. RMSE for wind speed one-step ahead prediction

	1-day ahead			5-days ahead		
	d=1	**d=2**	**d=3**	**d=1**	**d=2**	**d=3**
MA(7) + GARCH(1,1)	1.1202			6.9741		
Nonparametric (N-W)	***0.4882***	0.5386	0.5861	1.2118	1.2401	1.3086
Mixt 1 (nonparam. + residuals param.)	0.5422	0.5316	0.5277	1.1754	***1.1714***	1.1720
Mixt 2 (param. + residuals nonparam.)	4.9218	7.0882	6.9959	3.5856	4.7461	5.1945
Persistence model	0.0076			0.0115		

Table 9. RMSE for wind speed multi-step ahead prediction

		1-day ahead			5-days ahead		
		d=1	**d=2**	**d=3**	**d=1**	**d=2**	**d=3**
MA(7) + GARCH(1,1)		1.1119			7.0551		
Nonparametric (N-W)	*h=0.01*	1.1981	***0.8875***	1.4133	6.8078	7.6352	***3.7898***
	h=0.1	1.1452	1.0563	1.4172	6.6548	6.9834	6.5701
	h=1	1.0587	0.9145	0.9334	6.9777	7.2829	7.3767
	h_{NNR}	0.9188	0.9286	0.9753	7.2349	7.3226	7.2761
Mixt 1 (nonparam. + residuals param.)	*h=0.01*	9.4288	10.3290	2.0045	14.0410	14.7355	4.1625
	h=0.1	8.8041	7.0705	6.4379	13.0946	11.1751	10.3998
	h=1	1.7184	1.6727	1.6118	4.7038	4.9217	4.3937
	h_{NNR}	1.4114	7.3036	7.2060	4.1076	11.5012	11.5247
Mixt 2 (param. + residuals nonparam.)	*h=0.01*	4.5551	6.7577	4.8440	5.7330	8.7926	11.3209
	h=0.1	8.7432	6.0598	10.2802	15.6092	12.7748	17.5976
	h=1	3.0184	7.6158	8.0460	7.2822	14.4881	14.9266
	h_{NNR}	2.4652	6.6706	7.8677	7.2706	13.4739	14.7497
Persistence		1.0144			7.0431		

The overall results show that for the long-run predictions, the nonparametric model yield better results with respect to the RMSE than the persistence and parametric models, while in the one-step ahead case, the persistence model is in advantage for both horizon predictions. Analyzing more deeply the behavior of the results for the one-step ahead case, a closer look was directed to the RMSEs of the persistence model and the next best RMSE

from each horizon, results that are underlined in Table 8. Therefore, a step by step comparison of the RMSEs score was realized. The idea is to pass from the overall horizon prediction RMSE to the step-by-step prediction RMSE and to notice which model was the best in each step. For this reason, there were computed the ratios of the persistence RMSE to the next best model for each horizon, namely the nonparametric with dimension 1 for 1-day horizon and Mixt1 model with dimension 2 for 5-days horizon prediction. Ratios lower than 1 would indicate that the persistence model is a better fit, bringing a smaller error. More than 51% of the estimates for both 1-day and 5-days ahead horizon indicate that the nonparametric and the Mixt1 model, respectively, are yielding better predictions than the persistence model, even though the overall RMSE scores indicate a highly big difference. This means that when the persistence model "fails", it has some big error. In financial terms this may cause more damage than a constant RMSE like in the other case.

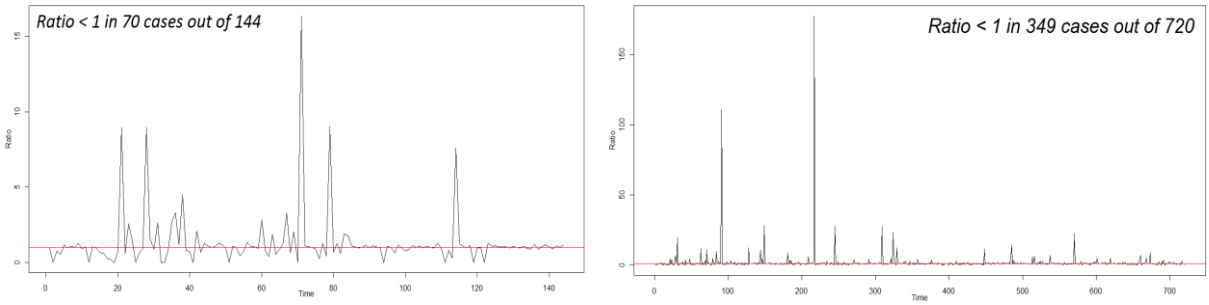


Figure 35. Ratios plot of the persistence model RMSE and the next best RMSE

The linear (parametric) models returned similar results for the one-step and multi-step ahead prediction, values around 1 in the 1-day ahead case and considerably increased in the 5-days ahead predictions, arriving around 7. Seeing the results, it is reasonable to state that the data need a nonlinear modeling.

The first semiparametric approach, that model the data nonparametrically and the residuals parametrically, brought good results in the 1-day and 5-days ahead case for the multiple-step ahead prediction. Even more for the one-step ahead predictions, it resulted to be the best fit for the 5-days ahead prediction, underlining that for long term predictions, taking into consideration both linear and nonlinear models is an advantage. The second semiparametric approach is not suitable for modeling these data for neither one of the procedures, despite the fact that it gave good estimations in the study of financial time series presented in [70].

The value of the smoothing parameter is proven to be of great importance. While in the nonparametric approach, the results are pretty similar for all the bandwidths, in the Mixt1 case, the choice of bandwidth results of being critical.

As in the case of the smoothing parameter, the right choice of dimension (d) is imperative. Even though there are studies that make possible the computations of the estimates for several choices of dimension and then choose the better solution, in the real life data this process may become a time and resources consumer due to the large amount of data.

Taking into consideration the best RMSE score for the wind speed of each prediction procedure, the wind power will be estimated according to the technique presented in Section. 4.2.3.. The results are shown in

Table 10.

Table 10. RMSE for wind power estimates

Horizon / Speed prediction method	1 day	5 days
one-step prediction	4.9265	444.3519
multiple-step prediction	1.7123	1226.7049

Analyzing the results one can see that for the one-step ahead prediction, due to the fact that the wind speed had, for that period, a stationary process with mean around 2-3m/s, the wind power estimates are almost all zeros. On the other side, looking at the 5-days ahead estimates, it is clear that one-step ahead predictions are better than multi-step ahead ones.

4.3.2. Prediction intervals

In order to increase in the credibility of the forecasted values, information about the uncertainty of the estimate must be provided. One method to handle this is by using the prediction intervals which inform about the range within future values will fall. There are several methods reported in the literature for computing the intervals depending on the statistic of interest and the sample characteristics. As in modeling, the confidence intervals can be computed parametrically or nonparametrically [71]. The most common parametric method for fixing the intervals is the one based on the assumption of approximate normality for the underlying sample. In this investigation, the bootstrapping approach will be used in order to compute prediction intervals, by using the percentile method.

Bootstrapping is a technique of figuring out the properties of statistical estimators by using simulation. The steps initiated in order to find the prediction intervals are:

- Obtain the residuals for the estimates $\hat{\varepsilon}_1, \dots, \hat{\varepsilon}_n$
- Resample the residuals *n=1000* times, in order to have 1000 samples ε_i^*
- Compute lower bound for each point time of the estimation: L_i = sample quantile corresponding to the probability 0.025, where $i = \overline{1,n}$
- Compute upper bound for each point time of the estimation: U_i = sample quantile corresponding to the probability 0.975, where $i = \overline{1,n}$

In Figure 36 and Figure 37 there is an illustration of the prediction intervals for wind speed and wind power.

In the 1-day ahead prediction for the wind power, due to the fact that few values are different from zero, the bootstrapped samples are almost the same and confidence intervals are not so clear.

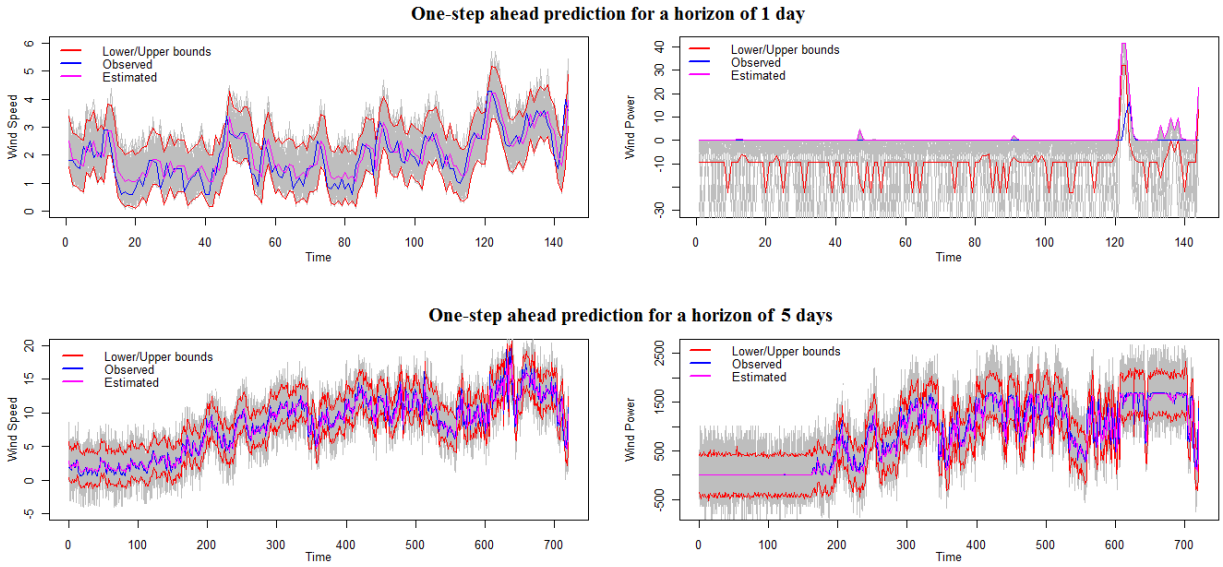


Figure 36. One-step ahead predictions for wind speed and wind power for 1 and 5 days

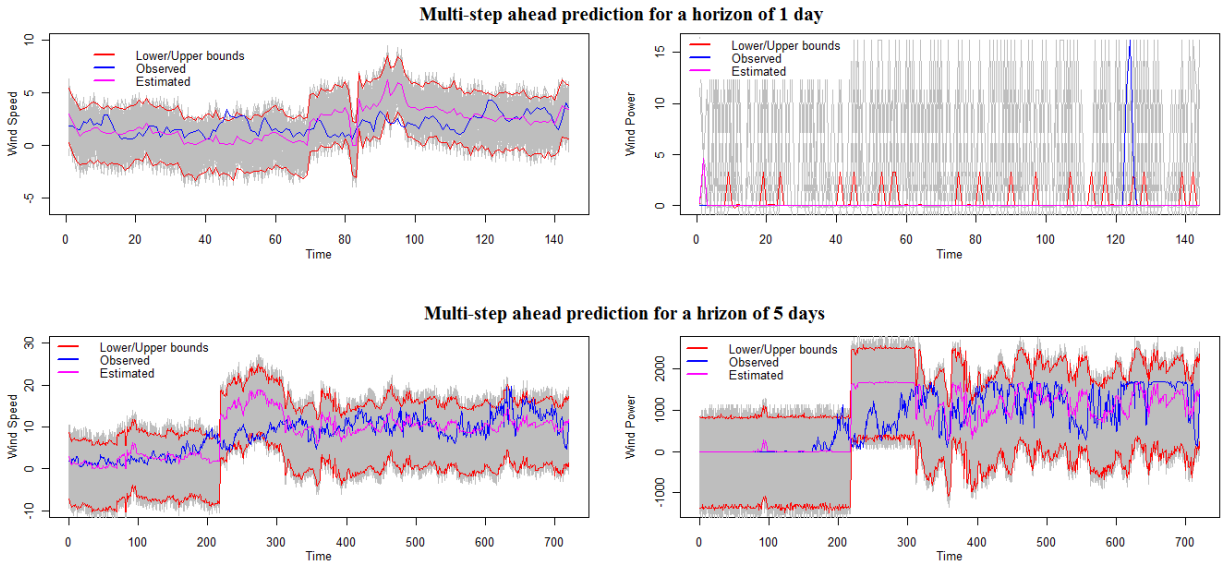


Figure 37. Multi-step ahead predictions for wind speed and wind power for 1 and 5 days

Consequently, in order to test the accuracy and measure the uncertainty of the models discussed in this investigation, the RMSE index and confidence intervals were used. The confidence intervals bring a clearer illustration of the model errors.

Chapter 5. Conclusion and future work

Wind power forecasts are essential for an efficient process and integration of wind power into the national grid. Due to the fact that wind fluctuates, wind energy cannot be stored efficiently and power shortages may appear during periods of low wind speed. Moreover, when wind speeds are too high, wind turbines need to be shut down, leading one more time to a power supply issue. In order to plan efficiently the power reserve, power system operators need to quantify accurately the uncertainties of wind power generation. On the other hand, wind farm operators want accurate predictions of wind power to reduce penalties and maximize the incomes from the electricity market.

This project has reviewed some forecasting techniques for the prediction of wind speed and wind power. A general conclusion that may be drawn from the obtained results is that the inclusion of nonparametric approaches in the wind speed modeling process, brought benefits to the prediction results. The dynamics and nonlinearity of the wind time series, was not possible to be taken into account in the parametric models used in this project, (ARMA and GARCH), but by modeling the data with a nonparametric approach like Nadaraya-Watson estimator, the results improved considerably. Furthermore, it was proved by the competitive results, that the mixt approach which model the data nonparametrically and then the residuals parametrically, it worth to be considered in nonlinear time series modeling.

Even though the performance of the forecasts for the one-step ahead period was surpassed by the persistence model, in the multiple-step ahead procedure the introduced methods of forecasting yielded improved results compared to the persistence model. Although, one-step-ahead prediction gave better estimates for the wind speed and wind power, it may not provide enough information, especially when it is required to understand the behavior of multiple steps in the future.

A number of possible future studies starting from this project can be discussed. One interesting feature would be to take into account some other options for the data transformation process in the identification phase of a model. In this investigation the stationarity process was considered only for the parametric models. As a future work, an idea will be to consider the input data for the nonparametric models also stationary. One other option would be to work with standardized data in order to avoid a big variability among values.

Two more improvements in the nonparametric models can be carried on by, on one hand, optimizing the choice for the smoother parameter, like for example a cross validation approach. On the other hand, the dimension, which refers to how many observations to take into account from the past, can be automatically computed by using different methods reported in the literature. These two new factors can improve considerably the results of the nonparametric and mixt models from this study.

It would be interesting to compare the wind power estimates presented in this project with those coming from the direct prediction of wind power. It is possible that the prediction error to be less than the bias introduced in the forecast by the transformation of the wind speed through the empirical wind power curve.

The accuracy of the wind power forecasts can bring a lot of economic benefits. Wind can be described as a stochastic process, meaning that the power output from a wind power plant can vary substantially through time, and it is not controllable in the same way as that of conventional power plants. In many areas, the winds strength is too low to support a wind turbine and during these breaks, electricity demand must be supplied by other resources. So, the simplest benefit of an accurate wind forecast is that wind-generated energy can be planned and used by the utility, so that the utility avoids the need to consume fuel to produce electricity. Even more, researchers are addressing these problems by means of energy storage, which implies adding a battery or other types of storage devices to the overall system. The forecast improvement plays an important role in the area of storing of energy. By using energy storages the exceeded power could be used later when the consumption is greater than the need, this being a huge progress in the wind power field.

Concluding, the aim of this study is to investigate some improved methods of predicting the wind power. As wind plants continue to become more economically competitive with conventional energy sources and generate green energy with the environmental advantages that entails, it is important to correctly assess the capacity of wind.

Appendix

Appendix A

Technical details for the turbines [15]

Ecotecnia 74		
	General data	Wind turbine name: 74 Nominal power: 1670 kW Rotor diameter: 74 m Offshore model: no Swept area: 4301 m² Power density: 0.03 m²/kW Number of blades: 3 Power control: Pitch Commissioning: 2003/01
	Manufacturer	Name: Ecotecnia Country: France Website: http://www.alstom.com/power/
	Weight	Hub: 63 tons Tube: 126 tons Rotor: 18 tons Total: 207 tons
	Rotor	Minimum rotor speed: 10 rd/min Maximum rotor speed: 19 rd/min Start-up wind speed: 3 m/s Nominal wind speed: 14 m/s Maximum wind speed: 25 m/s
	Gear box	Gear box: yes Speed number: 3 Manufacturer: Flender
	Tower	Minimum hub height: 60 m Maximum hub height: 80 m

Ecotecnia 80 1.6		
	General data	Wind turbine name: 80 1.6 Nominal power: 1670 kW Rotor diameter: 80 m Offshore model: no Swept area: 5027 m² Power density: 0.04 m²/kW

		Number of blades: 3 Power control: Pitch
	Manufacturer	Name: Ecotecnia Country: France Website: http://www.alstom.com/power/
	Weight	Hub: 63 tons Tube: 126 tons Rotor: 18 tons Total: 207 tons
	Rotor	Minimum rotor speed: 9,7 rd/min Maximum rotor speed: 18,4 rd/min Start-up wind speed: 3 m/s Nominal wind speed: 12 m/s Maximum wind speed: 25 m/s
	Gear box	Gear box: yes Speed number: 3 Manufacturer: Flender
	Tower	Minimum hub height: 70 m Maximum hub height: 80 m

Appendix B

R Code

Import data

```
install.packages("Hmisc"); library(Hmisc)
data2009=csv.get('Aero con missings 2009 tabla.csv',head=TRUE,sep=';', dec=",")
names(data2009) = c("Year","Month","Day","Hour","Minute","WindTurbine","XUTM",
"YUTM","Latitude","Longitude", "PowerMean","PowerMax", "PowerMin", "SpeedMean", "SpeedMax",
"SpeedMin", "Generator")
data2010=csv.get('Aero con missings 2010 tabla.csv',head=TRUE)
names(data2010) = c("Year","Month","Day","Hour","Minute","WindTurbine","XUTM",
"YUTM","Latitude","Longitude", "PowerMean","PowerMax", "PowerMin", "SpeedMean", "SpeedMax",
"SpeedMin", "Generator")

#### AEMET - ftp://ftpdatos.aemet.es/series_climatologicas/valores_mensuales/anual/
other_data2009=csv.get('2009.csv',head=TRUE,sep=';', dec=",")
other_data2009=subset(other_data2009,Nombre=="TARIFA")
other_data2009=other_data2009[,c(5,4,6,7,31)]
other_data2010=csv.get('2010.csv',head=TRUE,sep=';', dec=",")
other_data2010=subset(other_data2010,Nombre=="TARIFA")
other_data2010=other_data2010[,c(5,4,6,7,31)]

summary(data2009); summary(data2010)
data2009_105 = na.omit(subset(data2009, WindTurbine==105))
data2009_201 = na.omit(subset(data2009, WindTurbine==201))
data2009_311 = na.omit(subset(data2009, WindTurbine==311))
data2009_401 = na.omit(subset(data2009, WindTurbine==401))
```

```
data2010_105 = na.omit(subset(data2010, WindTurbine==105))
data2010_201 = na.omit(subset(data2010, WindTurbine==201))
data2010_311 = na.omit(subset(data2010, WindTurbine==311))
data2010_401 = na.omit(subset(data2010, WindTurbine==401))
```

Chapter 2. Descriptive statistics

```
library(lattice); library(Hmisc)
```

Figure 6 / pag. 10

```
win.graph()
par(mfrow=c(1,3))
hist(data2009_105[,14], xlab = "Mean Speed (m/s)", breaks="Sturges", main="Sturges approach", freq = F,
col="darkolivegreen1")
hist(data2009_105[,14], xlab = "Mean Speed (m/s)", breaks="Scott", main="Scott approach",freq = F,
col="azure2")
hist(data2009_105[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="FD approach", freq = F,
col="bisque")
```

Table 3. / pag 11

```
library(fBasics)
Mode = function(x) {
  ux = unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
## 105
round(basicStats(data2009_105[,14]),4)
Mode(data2009_105[,14])
normalTest(data2009_105[,14],method='jb') #test for normality Jarque Bera
round(basicStats(data2010_105[,14]),4)
Mode(data2010_105[,14])
normalTest(data2010_105[,14],method='jb') #test for normality Jarque Bera
```

Figure 7. / pag. 11

```
win.graph()
hist(data2009_105[,14], xlab = "Mean Speed (m/s)", breaks="FD", freq = F, main="Turbine 105 mean speed
histogram ")
points(mean(data2009_105[,14]),0, pch=21, bg="red");
points(median(data2009_105[,14]),0, pch=22, bg="green");
points(Mode(data2009_105[,14]),0, pch=25, bg="blue");
points(mean(data2009_105[,14])+sqrt(var(data2009_105[,14])),0, pch=21, bg="black")
points(mean(data2009_105[,14])-sqrt(var(data2009_105[,14])),0, pch=21, bg="black")
points(mean(data2009_105[,14])+2*sqrt(var(data2009_105[,14])),0, pch=21, bg="black")
points(mean(data2009_105[,14])+3*sqrt(var(data2009_105[,14])),0, pch=21, bg="black")
points(mean(data2009_105[,14])+4*sqrt(var(data2009_105[,14])),0, pch=21, bg="black")
legend(20,0.1, bty="n", x.intersp  = 1, y.intersp  = 1,legend=c("mean", "median", "mode"),
pt.bg=c("red","green","blue"), pch=c(21,22,25))
dev.off()
```

Figure 8. /pag.11

```
win.graph()
par(mfrow=c(2,2))
boxplot(data2009_105[,14],horizontal=T, main="Turbine 105")
boxplot(data2009_201[,14],horizontal=T, main="Turbine 201")
boxplot(data2009_311[,14],horizontal=T, main="Turbine 311")
boxplot(data2009_401[,14],horizontal=T, main="Turbine 401")
win.graph()
par(mfrow=c(2,2))
boxplot(data2010_105[,14],horizontal=T, main="Turbine 105")
```

```
boxplot(data2010_201[,14],horizontal=T, main="Turbine 201")
boxplot(data2010_311[,14],horizontal=T, main="Turbine 311")
boxplot(data2010_401[,14],horizontal=T, main="Turbine 401")
dev.off()
```

Figure 9. / pag. 12

```
win.graph()
par(mfrow=c(2,2))
hist(data2009_105[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 105", freq=F)
hist(data2009_201[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 201", freq=F)
hist(data2009_311[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 311", freq=F)
hist(data2009_401[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 401", freq=F)
win.graph()
par(mfrow=c(2,2))
hist(data2010_105[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 105", freq=F)
hist(data2010_201[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 201", freq=F)
hist(data2010_311[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 311", freq=F)
hist(data2010_401[,14], xlab = "Mean Speed (m/s)", breaks="FD", main="Turbine 401", freq=F)
```

Figure 10. / pag. 13 – Monthly variation of the mean wind speed in 2009 and 2010

```
monthly = function(data1, data2)
{
 aux = matrix(numeric(24),2,12)
 for (i in (1:12))
 {
  aux[1,i] = aux[1,i] + mean(data1[data1[,2]==i,14])
  aux[2,i] = aux[2,i] + mean(data2[data2[,2]==i,14])
 }
 return(aux)
}
plot(monthly(data2009_105,data2010_105)[1,], xlab="Month (2009)", ylab="Mean Wind Speed (m/s)",
type="b", xaxt="n", ylim=c(4.4,10))
lines(monthly(data2009_201,data2010_201)[1,], xlab="Month (2009)", ylab="Mean Wind Speed (m/s)",
type="b", col="red")
lines(monthly(data2009_311,data2010_311)[1,], xlab="Month (2009)", ylab="Mean Wind Speed (m/s)",
type="b", col="blue")
lines(monthly(data2009_401,data2010_401)[1,], xlab="Month (2009)", ylab="Mean Wind Speed (m/s)",
type="b", col="green")
lines(other_data2009[,5], xlab="Month (2009)", ylab="Mean Wind Speed (m/s)", type = "b", col= "orange",
lty=3,lwd=4)
axis(1, at=1:12, lab=c("Jan", "Feb", "Mar", "Apr",  "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))
legend(10.2,10.7, bty="n", x.intersp = 0.2, y.intersp = 0.2, legend = c("105", "201", "311", "401", "AEMET"),
col = c("black","red","blue","green","orange"), lty=c(1,1,1,1,2), pch = c(21, 21, 21, 21, 21), seg.len=rep(0.7,5),
lwd=rep(2,5))

plot(monthly(data2009_105,data2010_105)[2,], xlab="Month (2010)", ylab="Mean Wind Speed (m/s)",
type="b", xaxt="n",  ylim=c(4.4,10))
lines(monthly(data2009_201,data2010_201)[2,], xlab="Month (2010)", ylab="Mean Wind Speed (m/s)",
type="b", col="red")
lines(monthly(data2009_311,data2010_311)[2,], xlab="Month (2010)", ylab="Mean Wind Speed (m/s)",
type="b", col="blue")
lines(monthly(data2009_401,data2010_401)[2,], xlab="Month (2010)", ylab="Mean Wind Speed (m/s)",
type="b", col="green")
lines(other_data2010[,5], xlab="Month (2010)", ylab="Mean Wind Speed (m/s)", type="b", col="orange",
lty=3,lwd=4)
axis(1, at=1:12, lab=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov", "Dec"))
legend(10.2,10.7, bty="n", x.intersp = 0.2, y.intersp = 0.2, legend =c("105","201","311","401", "AEMET"),
col=c("black","red","blue","green","orange"), lty=c(1,1,1,1,2), pch = c(21, 21, 21, 21 ,21), seg.len=rep(0.7,5),
lwd=rep(2,5))
```

Figure 11. / pag. 14 - Hourly variation of the mean wind speed in 2009 and 2010

```
hourly = function(data1, data2)
{
  aux = matrix(numeric(48),2,24)
  for (i in (1:24))
  {
    aux[1,i] = aux[1,i] + mean(data1[data1[,4]==i-1,14])
    aux[2,i] = aux[2,i] + mean(data2[data2[,4]==i-1,14])
  }
  return(aux)
}

plot(hourly(data2009_105,data2010_105)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (m/s)", type="b",
xaxt="n",  ylim=c(5.4,8.7))
points(hourly(data2009_201,data2010_201)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (m/s)",
type="b", col="red")
points(hourly(data2009_311,data2010_311)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (m/s)",
type="b", col="blue")
points(hourly(data2009_401,data2010_401)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (m/s)",
type="b", col="green")
axis(1, at=1:24, lab=c(1:24))
legend(21,9, bty="n", x.intersp = 0.2, y.intersp = 0.2,legend=c("105","201","311","401"), col =
c("black","red","blue","green"), lty=rep(1,4), pch=rep(21,4), seg.len=rep(0.7,4), lwd=rep(2,4))

plot(hourly(data2009_105,data2010_105)[2,], xlab="Hour (2010)", ylab="Mean Wind Speed (m/s)", type="b",
xaxt="n", ylim=c(5.4,8.7))
points(hourly(data2009_201,data2010_201)[2,], xlab="Hour (2010)", ylab="Mean Wind Speed (m/s)",
type="b", col="red")
points(hourly(data2009_311,data2010_311)[2,], xlab="Hour (2010)", ylab="Mean Wind Speed (m/s)",
type="b", col="blue")
points(hourly(data2009_401,data2010_401)[2,], xlab="Hour (2010)", ylab="Mean Wind Speed (m/s)",
type="b", col="green")
axis(1, at=1:24, lab=c(1:24))
legend(21,8.8, bty="n", x.intersp = 0.2, y.intersp = 0.2,legend=c("105","201","311","401"), col =
c("black","red","blue","green"), lty=rep(1,4), pch=rep(21,4), seg.len=rep(0.7,4), lwd=rep(2,4))

max(hourly(data2009_105,data2010_105)[1,]); max(hourly(data2009_201,data2010_201)[1,])
max(hourly(data2009_311,data2010_311)[1,]); max(hourly(data2009_401,data2010_401)[1,])
max(hourly(data2009_105,data2010_105)[2,]); max(hourly(data2009_201,data2010_201)[2,])
max(hourly(data2009_311,data2010_311)[2,]); max(hourly(data2009_401,data2010_401)[2,])
```

Figure 13. / pag. 16

```
win.graph()
par(mfrow=c(1,4))
hist(data2009_105[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 105", freq = F)
hist(data2009_201[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 201", freq = F)
hist(data2009_311[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 311", freq = F)
hist(data2009_401[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 401", freq = F)
win.graph()
par(mfrow=c(1,4))
hist(data2010_105[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 105", freq = F)
hist(data2010_201[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 201", freq = F)
hist(data2010_311[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 311", freq = F)
hist(data2010_401[,11], xlab = "Mean Wind Power (kW)", breaks="FD", main="Turbine 401", freq = F)
```

Figure 14. / pag.16 - mean, median, mode

```
win.graph()
hist(data2009_105[,11], xlab = "Mean Wind Power (kW)", ylim=c(0,0.008), xlim = c(0,1700), breaks = "FD",
freq = F, main="")
points(mean(data2009_105[,11]),0, pch=21, bg="red");
```

```
points(median(data2009_105[,11]),0, pch=22, bg="green");
points(Mode(data2009_105[,11]),0, pch=25, bg="blue");
points(mean(data2009_105[,11])+sqrt(var(data2009_105[,11])),0, pch=21, bg="black")
points(mean(data2009_105[,11])-sqrt(var(data2009_105[,11])),0, pch=21, bg="black")
points(mean(data2009_105[,11])+2*sqrt(var(data2009_105[,11])),0, pch=21, bg="black")
legend(1300,0.006, bty="n", x.intersp  = 1, y.intersp  = 1,legend=c("mean", "median", "mode"),
pt.bg=c("red","green","blue"), pch=c(21,22,25))
dev.off()
```

Figure 15. / pag.17 - Monthly variation of the mean wind power in 2009 and 2010

```
monthly = function(data1, data2)
{
  aux = matrix(numeric(24),2,12)
  for (i in (1:12))
  {
    aux[1,i] = aux[1,i] + mean(data1[data1[,2]==i,11])
    aux[2,i] = aux[2,i] + mean(data2[data2[,2]==i,11])
  }
  return(aux)
}
plot(monthly(data2009_105,data2010_105)[1,], xlab="Month (2009)", ylab="Mean Wind Power (kW)",
type="b", xaxt="n",  ylim=c(150,850))
lines(monthly(data2009_201,data2010_201)[1,], xlab="Month (2009)", ylab="Mean Wind Power (kW)",
type="b", col="red")
lines(monthly(data2009_311,data2010_311)[1,], xlab="Month (2009)", ylab="Mean Wind Power (kW)",
type="b", col="blue")
lines(monthly(data2009_401,data2010_401)[1,], xlab="Month (2009)", ylab="Mean Wind Power (kW)",
type="b", col="green")
axis(1, at=1:12, lab=c("Jan", "Feb", "Mar", "Apr", "May","Jun","Jul","Aug","Sep","Oct","Nov", "Dec"))
legend(10.5, 920, bty="n", x.intersp=0.2, y.intersp = 0.2,legend=c("105","201","311","401"), col =
c("black","red","blue","green"), lty=rep(1,4), pch=rep(21,4), seg.len=rep(0.7,4), lwd=rep(2,4))

plot(monthly(data2009_105,data2010_105)[2,], xlab="Month (2010)", ylab="Mean Wind Power (kW)",
type="b", xaxt="n",  ylim=c(150,850))
lines(monthly(data2009_201,data2010_201)[2,], xlab="Month (2010)", ylab="Mean Wind Power (kW)",
type="b", col="red")
lines(monthly(data2009_311,data2010_311)[2,], xlab="Month (2010)", ylab="Mean Wind Power (kW)",
type="b", col="blue")
lines(monthly(data2009_401,data2010_401)[2,], xlab="Month (2010)", ylab="Mean Wind Power (kW)",
type="b", col="green")
axis(1, at=1:12, lab=c("Jan", "Feb", "Mar", "Apr", "May","Jun","Jul","Aug","Sep","Oct","Nov", "Dec"))
legend(10.5, 920, bty="n", x.intersp = 0.2, y.intersp=0.2,legend=c("105","201","311","401"),col =
c("black","red","blue","green"), lty=rep(1,4), pch=rep(21,4), seg.len=rep(0.7,4), lwd=rep(2,4))

max(monthly(data2009_105,data2010_105)[1,]); max(monthly(data2009_201,data2010_201)[1,])
max(monthly(data2009_311,data2010_311)[1,])
max(monthly(data2009_401,data2010_401)[1,])

min(monthly(data2009_105,data2010_105)[1,])
min(monthly(data2009_201,data2010_201)[1,])
min(monthly(data2009_311,data2010_311)[1,])
min(monthly(data2009_401,data2010_401)[1,])

max(monthly(data2009_105,data2010_105)[2,])
max(monthly(data2009_201,data2010_201)[2,])
max(monthly(data2009_311,data2010_311)[2,])
max(monthly(data2009_401,data2010_401)[2,])

monthly_var = function(data1, data2)
{
```

```
  aux = matrix(numeric(24),2,12)
  for (i in (1:12))
  {
    aux[1,i] = aux[1,i] + sqrt(var(data1[data1[,2]==i,14]))
    aux[2,i] = aux[2,i] + sqrt(var(data2[data2[,2]==i,14]))
  }
  return(aux)
}
monthly_var(data2009_105,data2010_105)[1,]
monthly_var(data2009_201,data2010_201)[1,]
monthly_var(data2009_311,data2010_311)[1,]
monthly_var(data2009_401,data2010_401)[1,]

round(monthly_var(data2009_105,data2010_105)[2,],2)
round(monthly_var(data2009_201,data2010_201)[2,],2)
round(monthly_var(data2009_311,data2010_311)[2,],2)
round(monthly_var(data2009_401,data2010_401)[2,],2)
```

Figure 16. / pag. 18 - Hourly variation of the mean wind speed in 2009 and 2010

```
hourly = function(data1, data2)
{
  aux = matrix(numeric(48),2,24)
  for (i in (1:24))
  {
    aux[1,i] = aux[1,i] + mean(data1[data1[,4]==i-1,11])
    aux[2,i] = aux[2,i] + mean(data2[data2[,4]==i-1,11])
  }
  return(aux)
}

plot(hourly(data2009_105,data2010_105)[1,], xlab="Hour (2009)", ylab="Mean Wind Power (kW)", type="b",
xaxt="n",  ylim=c(300,700))
points(hourly(data2009_201,data2010_201)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (kW)",
type="b", col="red")
points(hourly(data2009_311,data2010_311)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (kW)",
type="b", col="blue")
points(hourly(data2009_401,data2010_401)[1,], xlab="Hour (2009)", ylab="Mean Wind Speed (kW)",
type="b", col="green")
axis(1, at=1:24, lab=c(1:24))
legend(21, 750, bty="n", x.intersp = 0.2, y.intersp = 0.2,legend=c("105","201","311","401"), col =
c("black","red","blue","green"), lty=rep(1,4), pch=rep(21,4), seg.len=rep(0.7,4), lwd=rep(2,4))

plot(hourly(data2009_105,data2010_105)[2,], xlab="Hour (2010)", ylab="Mean Wind Power (kW)", type="b",
xaxt="n",  ylim=c(300,700))
points(hourly(data2009_201,data2010_201)[2,], xlab="Hour (2010)", ylab="Mean Wind Power (kW)",
type="b", col="red")
points(hourly(data2009_311,data2010_311)[2,], xlab="Hour (2010)", ylab="Mean Wind Power (kW)",
type="b", col="blue")
points(hourly(data2009_401,data2010_401)[2,], xlab="Hour (2010)", ylab="Mean Wind Power (kW)",
type="b", col="green")
axis(1, at=1:24, lab=c(1:24))
legend(21, 750, bty="n", x.intersp = 0.2, y.intersp = 0.2,legend=c("105","201","311","401"),col
=c("black","red","blue","green"), lty=rep(1,4), pch=rep(21,4), seg.len=rep(0.7,4), lwd=rep(2,4))

sqrt(var(data2009_105[,14])); sqrt(var(data2009_201[,14])); sqrt(var(data2009_311[,14]))
sqrt(var(data2009_401[,14]))
```

Figure 17. / pag. 18 - wind rose

```
dataHirlam2 = csv.get("dataHirlam2.csv")
```

```
install.packages("openair"); library(openair)
myDatawr = dataHirlam2[,c(8,7)]
names(myDatawr)=c("ws", "wd")
windRose(myDatawr, breaks=12, paddle=F)

## Figure 18. + Figure 19. / pag. 19-20 - delete outliers
turbines = c(101:106, 201:210, 301:311, 401:407)
n = length(turbines)
i=34; cat("Turbine:", turbines[i])
for (i in 1:n)
{
delete_outliers_function(subset(data2009, WindTurbine==turbines[i]),25,paste("data",turbines[i],"no_out.csv",
sep=""))
}
win.graph()
par(mfrow=c(2,2))
plot(data2009_105[,14], data2009_105[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 105")
plot(data2009_201[,14], data2009_201[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 201")
plot(data2009_311[,14], data2009_311[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 311")
plot(data2009_401[,14], data2009_401[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 401")

win.graph()
par(mfrow=c(2,2))
plot(data105no_out[,14], data105no_out[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 105")
plot(data201no_out[,14], data201no_out[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 201")
plot(data311no_out[,14], data311no_out[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 311")
plot(data401no_out[,14], data401no_out[,11], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 401")

## extract negative data
data105no_out = subset(data105no_out, !SpeedMean<0)
data105no_out = subset(data105no_out, !PowerMean<0)
data201no_out = subset(data201no_out, !SpeedMean<0)
data201no_out = subset(data201no_out, !PowerMean<0)
data311no_out = subset(data311no_out, !SpeedMean<0)
data311no_out = subset(data311no_out, !PowerMean<0)
data401no_out = subset(data401no_out, !SpeedMean<0)
data401no_out = subset(data401no_out, !PowerMean<0)

####clean data from t201
win.graph()
plot(data201no_out[,11], data201no_out[,14], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 201")
l=locator(1)
data201no_out = subset(data201no_out,!(SpeedMean<l$y & PowerMean>l$x))
plot(data201no_out[,11], data201no_out[,14], xlab="Mean wind speed", ylab="Mean wind power",
main="Turbine 201")

## Figure 20. / pag. 21 - patterns by month
par(mfrow=c(1,2))
plot(, patterns[,2], ylab="Month", xlab="Power")
plot(patterns[,7], patterns[,2], ylab="Month", xlab="Speed")
dev.off()
```

Figure 21. / pag. 21 - two shapes pattern

```
win.graph()
par(mfrow=c(3,2))
plot(data311no_out[,14],data311no_out[,11], xlab="Mean wind speed", ylab="Mean wind power")
abline(v=8, col="red")
s8 = data311no_out[data311no_out[,14]==8,]
plot(s8[,2], s8[,11], xlab="Month", ylab="Mean wind power", main="Monthly wind power when wind speed is
8 m/s")

plot(data311no_out[,14],data311no_out[,11], xlab="Mean wind speed", ylab="Mean wind power")
abline(v=9, col="red")
s9 = data311no_out[data311no_out[,14]==9,]
plot(s9[,2], s9[,11], xlab="Month", ylab="Mean wind power", main="Monthly wind power when wind speed is
9 m/s")

plot(data311no_out[,14],data311no_out[,11], xlab="Mean wind speed", ylab="Mean wind power")
abline(v=10, col="red")
s10 = data311no_out[data311no_out[,14]==10,]
plot(s10[,2], s10[,11], xlab="Month", ylab="Mean wind power", main="Monthly wind power when wind speed
is 10 m/s")
dev.off()

plot(data311no_out[,14],data311no_out[,11],xlab="Mean wind speed", ylab="Mean wind power")
points(data311no_out[data311no_out[,2]==12,14],data311no_out[data311no_out[,2]==12,11], col="green")
legend (20, 500, bty="n", x.intersp  = 0.3, y.intersp  = 0.3, legend = c("Jan-Nov","Dec"), col = c("black",
"green"), pch=19)
data311no_out_dec = subset(data311no_out, data311no_out[,2]==12)
data311no_out_j_n = subset(data311no_out, data311no_out[,2] %in% c(1:11))
```

Introduce missing values

```
library(chron)
## turbine 105
data105no_out$date=dates(paste(data105no_out[,2],data105no_out[,3],data105no_out[,1],sep="/"))
data105no_out$hour=times(paste(data105no_out[,4],data105no_out[,5],0,sep=":"))
row.names(data105no_out)=paste(data105no_out$date,data105no_out$hour)
a=expand.grid(seq(0,50,by=10),0:23,dates("01/01/2009")+0:364)
d=data.frame(days=a[,3],hours=times(paste(a[,2],a[,1],0,sep=":")))
data105no_out_wmiss=data105no_out[paste(d$days,d$hours),]

data105no_out_wmiss[,1]=rep(2009,365*24*6) ## year
data105no_out_wmiss[,4]=rep(rep(0:23, each=6),365)  ## hour
data105no_out_wmiss[,2] = as.numeric(substr(a[,3],1,2))  ##month
data105no_out_wmiss[,3] = as.numeric(substr(a[,3],4,5)) ##day
data105no_out_wmiss[,5]=rep(seq(0,50,by=10),365*24) ## minute
data105no_out_wmiss[,6]=rep(105,365*24*6)
data105no_out_wmiss[,7]=rep(data105no_out_wmiss[1,7],365*24*6)
data105no_out_wmiss[,8]=rep(data105no_out_wmiss[1,8],365*24*6)
data105no_out_wmiss[,9]=rep(data105no_out_wmiss[1,9],365*24*6)
data105no_out_wmiss[,10]=rep(data105no_out_wmiss[1,10],365*24*6)
```

Figure 22. and table 5. / pag. 25 - missings pattern

```
library(VIM)
NA_mat = matrix(data=0,nrow=4,ncol=12)
for (j in 1:12) NA_mat[1,j] = countNA(data105no_out_wmiss[data105no_out_wmiss[,2]==j,11])
for (j in 1:12) NA_mat[2,j] = countNA(data201no_out_wmiss[data201no_out_wmiss[,2]==j,11])
for (j in 1:12) NA_mat[3,j] = countNA(data311no_out_wmiss[data311no_out_wmiss[,2]==j,11])
for (j in 1:12) NA_mat[4,j] = countNA(data401no_out_wmiss[data401no_out_wmiss[,2]==j,11])
rownames(NA_mat)=c("105", "201", "311", "401")
colnames(NA_mat)=c("1","2","3","4","5","6","7","8","9","10","11","12")
```

```
barplot(NA_mat, beside=T, col = c("lightblue", "mistyrose", "lightcyan","lavender"))

sum(NA_mat[,c(3,4,5)]);sum(NA_mat[,c(6,7,8)]);sum(NA_mat[,c(9,10,11)]);sum(NA_mat[,c(12,1,2)])
dim(subset(data105no_out_wmiss, Month==3 | Month==4 | Month==5))[[1]]
dim(subset(data105no_out_wmiss, Month==6 | Month==7 | Month==8))[[1]]
dim(subset(data105no_out_wmiss, Month==9 | Month==10 | Month==11))[[1]]
dim(subset(data105no_out_wmiss, Month==12 | Month==1 | Month==2))[[1]]

NA_mat1 = matrix(data=0,nrow=4,ncol=31)
for (j in 1:31) NA_mat1[1,j]=countNA(data105no_out_wmiss[data105no_out_wmiss[,3]==j,11])
for (j in 1:31) NA_mat1[2,j]=countNA(data201no_out_wmiss[data201no_out_wmiss[,3]==j,11])
for (j in 1:31) NA_mat1[3,j]=countNA(data311no_out_wmiss[data311no_out_wmiss[,3]==j,11])
for (j in 1:31) NA_mat1[4,j]=countNA(data401no_out_wmiss[data401no_out_wmiss[,3]==j,11])
barplot(NA_mat1, beside=T, col = c("lightblue", "mistyrose", "lightcyan","lavender"))
```

missing imputation by MEDIAN

```
#install.packages("e1071"); library(e1071)
load("final_dataset.RData"); rm(list=setdiff(ls(), "data105no_out_wmiss"))
data105_median_imp = data105no_out_wmiss
data105_median_imp[,14] = impute(data.frame(data105no_out_wmiss[,14]),"median")
colnames(data105_median_imp[,14]) = "SpeedMean"
plot(data105_median_imp[,11], type="l"); lines(data105no_out_wmiss[,11], col=2)
summary(data105no_out_wmiss[,11]); summary(data105_median_imp[,11])
mydata = data.frame(data105no_out_wmiss[,14],data105no_out_wmiss[,11])
data105_median_imp[,11] = powerImputation (mydata, data105_median_imp[,14])
```

missing imputation by NEAREST NEIGHBOR

```
install.packages("SeqKnn"); library(SeqKnn)
data105_knn_imp = data105no_out_wmiss
data105_knn_imp[,c(11,14)] = SeqKNN(data105no_out_wmiss[,c(11,14)],10)
plot(data105_knn_imp[,11], type="l"); lines(data105no_out_wmiss[,11], col=2)
mydata = data.frame(data105no_out_wmiss[,14],data105no_out_wmiss[,11])
data105_knn_imp[,11] = powerImputation(mydata, data.frame(data105_knn_imp[,14]))
```

missing imputation by ARIMA(0,7) + GARCH(1,1)

```
library(Hmisc); library(tseries); library(fGarch)
load("final_dataset.RData");rm(list=setdiff(ls(), "data105no_out_wmiss"));
source("functions_missingImputation.R")
data105_arima_imp = data105no_out_wmiss
data105_arima_imp[,14] = imputationARIMA(data105no_out_wmiss[,14])
mydata = data.frame(data105no_out_wmiss[,14],data105no_out_wmiss[,11])
data105_arima_imp[,11] = powerImputation(mydata, data.frame(data105_arima_imp[,14]))
plot(data105_arima_imp[,11], type="l"); lines(data105no_out_wmiss[,11], col=2)
plot(data105_arima_imp[,14], type="l"); lines(data105no_out_wmiss[,14], col=2)
```

missing imputation by EM modified (mtsdi)

```
library(mtsdi)
f105 = ~data105no_out_wmiss[,11]+data105no_out_wmiss[,14]
f201 = ~data201no_out_wmiss[,11]+data201no_out_wmiss[,14]
f311_1 = ~data311no_out_wmiss1[,11]+data311no_out_wmiss1[,14]
f311_2 = ~data311no_out_wmiss2[,11]+data311no_out_wmiss2[,14]
f401 = ~data401no_out_wmiss[,11]+data401no_out_wmiss[,14]

data105_imp = mnimput(f105, data105no_out_wmiss, eps=1e-3, ts=TRUE, method="spline")
data201_imp = mnimput(f201, data201no_out_wmiss, eps=1e-3, ts=TRUE, method="spline")
data311_imp_1=mnimput(f311_1,data311no_out_wmiss1,eps=1e-3,ts=TRUE, method="spline")
data311_imp_2=mnimput(f311_2,data311no_out_wmiss2,eps=1e-3,ts=TRUE, method="spline")
data401_imp = mnimput(f401, data401no_out_wmiss, eps=1e-3, ts=TRUE, method="spline")

win.graph()  #T105
```

```
par(mfrow=c(2,1))
plot.mtsdi(data105_imp, level=F, leg.loc=c(-100,-100), horiz=T)
summary(data105no_out_wmiss[,11])
summary(data105_imp_values[,1])
summary(data105no_out_wmiss[,14])
summary(data105_imp_values[,2])

plot(data105no_out_wmiss[,14], type="l", col="red"); lines(data105_imp_values[,2])

data105_imp_values = predict(data105_imp)
data201_imp_values = predict(data201_imp)
data311_imp_values_1 = predict(data311_imp_1)
data311_imp_values_2 = predict(data311_imp_2)
data311_imp_values = rbind(data311_imp_values_1,data311_imp_values_2)
data401_imp_values = predict(data401_imp)

win.graph()
par(mfrow=c(2,2))
plot(data105_imp_values[,2],data105_imp_values[,1])
plot(data201_imp_values[,2],data201_imp_values[,1])
plot(data311_imp_values[,2],data311_imp_values[,1])
plot(data401_imp_values[,2],data401_imp_values[,1])
dev.off()

## store results with EM approach
data105_EM_imp = data105no_out_wmiss; data201_EM_imp = data201no_out_wmiss; data311_EM_imp =
data311no_out_wmiss; data311_EM_imp1 = data311no_out_wmiss1; data311_EM_imp2 =
data311no_out_wmiss2; data401_EM_imp = data401no_out_wmiss;
data105_EM_imp[,c(11,14)] = data105_imp_values; data201_EM_imp[,c(11,14)] = data201_imp_values;
data311_EM_imp[,c(11,14)] = data311_imp_values; data311_EM_imp1[,c(11,14)] = data311_imp_values_1;
data311_EM_imp2[,c(11,14)] = data311_imp_values_2; data401_EM_imp[,c(11,14)] = data401_imp_values

## Correct the imputed values ##
#install.packages("mgcv"); library(mgcv)
data105_imp_corr = data105no_out_wmiss
data201_imp_corr = data105no_out_wmiss
data311_imp_corr_1 = data311no_out_wmiss1
data311_imp_corr_2 = data311no_out_wmiss2
data401_imp_corr = data401no_out_wmiss

data105_imp_corr$PowerMean[data105_imp_corr$PowerMean<=0] = 0.5
data201_imp_corr$PowerMean[data201_imp_corr$PowerMean<=0] = 0.5
data311_imp_corr_1$PowerMean[data311_imp_corr_1$PowerMean<=0] = 0.5
data311_imp_corr_2$PowerMean[data311_imp_corr_2$PowerMean<=0] = 0.5
data401_imp_corr$PowerMean[data401_imp_corr$PowerMean<=0] = 0.5

f = PowerMean~s(SpeedMean)
res105 = gam(f,data=data105_imp_corr,family=gaussian(link="log"))
res201 = gam(f,data=data201_imp_corr,family=gaussian(link="log"))
res311_1 = gam(f,data=data311_imp_corr_1,family=gaussian(link="log"))
res311_2 = gam(f,data=data311_imp_corr_2,family=gaussian(link="log"))
res401 = gam(f,data=data401_imp_corr,family=gaussian(link="log"))

pred105 = predict(res105,list("SpeedMean"=data105_imp_values[,2]))
pred201 = predict(res201,list("SpeedMean"=data201_imp_values[,2]))
pred311_1 = predict(res311_1,list("SpeedMean"=data311_imp_values_1[,2]))
pred311_2 = predict(res311_2,list("SpeedMean"=data311_imp_values_2[,2]))
pred401 = predict(res401,list("SpeedMean"=data401_imp_values[,2]))
pred311 = c(pred311_1, pred311_2)
data105_imp_corr[,11] = exp(pred105); data105_imp_corr[,14] = data105_imp_values[,2]
```

```
data201_imp_corr[,11] = exp(pred201); data201_imp_corr[,14] = data201_imp_values[,2]
data311_imp_corr_1[,11]=exp(pred311_1);data311_imp_corr_1[,14]=data311_imp_values_1[,2]
data311_imp_corr_2[,11]=exp(pred311_2);data311_imp_corr_2[,14]=data311_imp_values_2[,2]
data311_imp_corr = rbind(data311_imp_corr_1, data311_imp_corr_2)
data401_imp_corr[,11] = exp(pred401); data401_imp_corr[,14] = data401_imp_values[,2]

## replace missings with the estimated values
data105_EM_GAM_imp = replaceMissingsEM(data105no_out_wmiss, data105_imp_corr)
data201_EM_GAM_imp = replaceMissingsEM(data201no_out_wmiss, data201_imp_corr)
data311_EM_GAM_imp = replaceMissingsEM(data311no_out_wmiss, data311_imp_corr)
data311_EM_GAM_imp1 = replaceMissingsEM(data311no_out_wmiss1, data311_imp_corr_1)
data311_EM_GAM_imp2 = replaceMissingsEM(data311no_out_wmiss2, data311_imp_corr_2)
data401_EM_GAM_imp = replaceMissingsEM(data401no_out_wmiss, data401_imp_corr)

#T105
win.graph()
  plot(data105_EM_imp[,14],data105no_out_wmiss[,11]);
  points(data105_EM_imp[,14],data105_EM_imp[,11],col="red")
win.graph()
  par(mfrow=c(2,1))
  hist(data105no_out_wmiss[,11])
  hist(data105_EM_imp[,11])
win.graph()
  plot(data105no_out_wmiss[,11],data105_EM_imp[,11])
win.graph()
  par(mfrow=c(2,1))
  ts.plot(data105_EM_imp[,11], col=2)
  lines(data105no_out_wmiss[,11])
  ts.plot(data105_EM_imp[,14], col=2)
  lines(data105no_out_wmiss[,14])
```

Misssing imputation with NP methods

```
load("final_dataset.RData"); rm(list=setdiff(ls(), "data105no_out_wmiss"))
data105_NP_imp = data105no_out_wmiss
data105_NP_imp[,14] = imputationNP_Speed(data105no_out_wmiss[,14], lag=3)
mydata = data.frame(data105no_out_wmiss[,14],data105no_out_wmiss[,11])
data105_NP_imp[,11] = powerImputation(mydata, data.frame(data105_NP_imp[,14]))
plot(data105_NP_imp[,14], type="l"); lines(data105no_out_wmiss[,14], col=2)
summary(data105no_out_wmiss[,11]) ; summary(data105_NP_imp[,11])
```

functions_missingImputation.R

```
library(tseries); library(fGarch); #library(audio)
findPatterns = function(data, missingPosition, missingPattern)
{
 i = 1; j=0; n = length(missingPosition)
 while (i<n)
 {
  j=i
  while (((missingPosition[j+1] - missingPosition[j]) == 1))
  {
   j=j+1
   if (j==n) break
  }
  missingPattern = rbind(missingPattern,c(missingPosition[i],missingPosition[j]))
  i = j+1
 }
 if (j != n) missingPattern = rbind(missingPattern,c(missingPosition[n],missingPosition[n]))
 return(missingPattern)
}
```

```
estimateValue =  function (d, step, i_s, n_s)
{
 tm=proc.time()
 model = garchFit(~arma(0,7)+garch(1,1), data=d, trace=F)
 estimate = predict(model,n.ahead = step, se.fit=F)
 cat("Step ",i_s, "of ", n_s,"..... Time elapsed: ",(proc.time()-tm)[3],"\n")
 return(estimate[,1])
}

imputationARIMA = function(data)
{
 d1data = diff(data)
 missingPosition = which(is.na(d1data))
 missingPattern = matrix(0,nrow=0, ncol=2)
 missingPattern = findPatterns(d1data, missingPosition, missingPattern)
 data_imp = d1data; i=1;
 tm = proc.time()
 for (i in (1:dim(missingPattern)[1]))
 {
   data_imp[missingPattern[i,1]:missingPattern[i,2]]=-estimateValue( data_imp[1:(missingPattern[i,1]-
1)],(missingPattern[i,2]-missingPattern[i,1]+1), i,dim(missingPattern)[1])
 }
 cat("Total elapsed time:",(proc.time()-tm)[3])
 missingPosition = which(is.na(data));  data_imp_corr = data
 data_imp_corr[missingPosition] = diffinv(data_imp, xi=data[1])[missingPosition]
 return(data_imp_corr)
}

replaceMissingsEM = function(data_wmiss, data_imp)
{
 missingPosition = which(is.na(data_wmiss[,11]))
 data_imp_corr = data_wmiss;  data_imp_corr[missingPosition,] = data_imp[missingPosition,]
 return(data_imp_corr)
}

imputationNP_Speed = function(data, lag)
{
 missingPosition = which(is.na(data));  missingPattern = matrix(0,nrow=0, ncol=2)
 missingPattern = findPatterns(data, missingPosition, missingPattern)
 delay=0;  n = dim(missingPattern)[1];  newdata = data;
 for (i in (1:n))
 {
  cat("Step ", i, " out of ",n, sep="", "\n");
  dset = newdata[1:(missingPattern[i,1]-1)];
  est = numeric(0);
  if ((missingPattern[i,2]-missingPattern[i,1]+1) > 30)
     sigma = 0.04
  else
     sigma = 1/(missingPattern[i,2]-missingPattern[i,1]+1);
  est = NonParametric(sigma, dset, (missingPattern[i,2]-missingPattern[i,1]+1), delay,lag);
  newdata[1:missingPattern[i,2]] = c(dset, est);
 }
 return(newdata)
}

imputationNP_Power = function(data, lag)
{
 missingPosition = which(is.na(data))
 missingPattern = matrix(0,nrow=0, ncol=2)
 missingPattern = findPatterns(data, missingPosition, missingPattern)
```

```
 delay=0;  n = dim(missingPattern)[1];  newdata = data;
 for (i in (1:n))
 {
  cat("Step ", i, " out of ",n, sep="", "\n");
  dset = newdata[1:(missingPattern[i,1]-1)];
  est = numeric(0);  sigma = 0.5;
  est = NonParametric(sigma, dset, (missingPattern[i,2]-missingPattern[i,1]+1), delay,lag);
  newdata[1:missingPattern[i,2]] = c(dset, est);
 }
 return(newdata)
}
```

Figure 23. and Figure 24. / pag. 30-31 - Results comparison

```
load("imputation_datasets.RData")
##power
win.graph()
par(mfrow=c(5,1))
plot(data105_median_imp[,11], type="l", main="Median imputation")
lines(data105no_out_wmiss[,11], col=2)
plot(data105_knn_imp[,11], type="l", main="KNN imputation")
lines(data105no_out_wmiss[,11], col=2)
plot(data105_arima_imp[,11], type="l", main="ARMA+GARCH imputation")
lines(data105no_out_wmiss[,11], col=2)
plot(data105_EM_GAM_imp[,11], type="l", main="EM modified imputation")
lines(data105no_out_wmiss[,11], col=2)
plot(data105_NP_imp[,11], type="l", main="Nonparametric imputation", col=2, ylab="Wind Power",
xlab="Time")
lines(data105no_out_wmiss[,11], col=1)

##speed
win.graph();par(mfrow=c(3,1))
plot(data105_median_imp[,14], type="l", main="Median imputation", col=2, ylab="Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[,14], col=1)
plot(data105_knn_imp[,14], type="l", main="KNN imputation", col=2, ylab="Wind Speed", xlab="Time")
lines(data105no_out_wmiss[,14], col=1)
plot(data105_arima_imp[,14], type="l", main="ARMA+GARCH imputation", col=2, ylab= "Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[,14], col=1)
win.graph(); par(mfrow=c(3,1))
plot(data105_EM_GAM_imp[,14], type="l", main="EM modified imputation", col=2, ylab= "Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[,14], col=1)
plot(data105_NP_imp[,14], type="l", main="Nonparametric imputation", col=2, ylab="Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[,14], col=1);dev.off()

##speed - 4000 sample
win.graph(); par(mfrow=c(3,1))
plot(data105_median_imp[1:4000,14], type="l", main="Median imputation", col=2, ylab="Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[1:4000,14], col=1)
plot(data105_knn_imp[1:4000,14], type="l", main="KNN imputation", col=2, ylab="Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[1:4000,14], col=1)
plot(data105_arima_imp[1:4000,14], type="l", main="ARMA+GARCH imputation", col=2, ylab="Wind
Speed", xlab="Time")
lines(data105no_out_wmiss[1:4000,14], col=1)
win.graph(); par(mfrow=c(3,1))
```

```
plot(data105_EM_GAM_imp[1:4000,14], type="l", main="EM modified imputation", col=2, ylab="Wind
Speed", xlab="Time")
lines(data105no_out_wmiss[1:4000,14], col=1)
plot(data105_NP_imp[1:4000,14], type="l", main="Nonparametric imputation", col=2, ylab= "Wind Speed",
xlab="Time")
lines(data105no_out_wmiss[1:4000,14], col=1); dev.off()
```

Figure 25, 26, 27, 28, 29 / pag. 34 - time series forecasting –

```
library(tseries)
newSpeed = read.table("newSpeed2.csv")
dnewSpeed=diff(newSpeed[,1])

splot(newSpeed[,1], ylab="Wind speed", main="Wind speed time series", xlab="Time",type="l")
par(mfrow=c(1,2));
acf(newSpeed[,1], main="ACF of the wind speed series");
pacf(newSpeed[,1], main="PACF of the wind speed series")

plot(dnewSpeed, ylab="Diff(wind speed)", main="Differenced wind speed series", xlab= "Time",type="l")
par(mfrow=c(1,2));
acf(dnewSpeed, ylim=c(-0.1,0.1), main="ACF of the diff wind speed series");
pacf(dnewSpeed, ylim=c(-0.1,0.1), main="PACF of the diff wind speed series")

model = arima(newSpeed[,1], order=c(0,1,7))

win.graph(); par(mfrow=c(2,2));
acf(model$residuals, ylim=c(-0.1,0.1), main="ACF of MA(7) residuals");
pacf(model$residuals, ylim=c(-0.1,0.1), main="PACF of MA(7) residuals")
acf(model$residuals^2, ylim=c(-0.1,0.1), main="ACF of MA(7) squared residuals");
pacf(model$residuals^2, ylim=c(-0.1,0.1), main="PACF of MA(7) squared residuals")

model1 = garch(model$residuals, order=c(1,1), trace=F)
win.graph() ; par(mfrow=c(2,2));
acf(model1$residuals[-1], ylim=c(-0.1,0.1), main="ACF of GARCH(1,1) residuals");
pacf(model1$residuals[-1], ylim=c(-0.1,0.1), main="PACF of GARCH(1,1) residuals");
acf(model1$residuals[-1]^2, ylim=c(-0.1,0.1), main="ACF of GARCH(1,1) squared residuals");
pacf(model1$residuals[-1]^2, ylim=c(-0.1,0.1), main="PACF of GARCH(1,1) squared residuals")
```

functions_general.R

```
# Gaussian Kernel
K =  function (x) prod(dnorm(x))

# Nadaraya-Watson estimator
rn = function (x,h,X,Y)
{
  n = length(Y)
  dum = sapply(1:n,function (i) K((x-X[i,])/h))
  sum(Y*dum)/sum(dum)
}

# approximation of the local optimal bandwidth at the point x
bandwidth.x =  function (x,X)
{
  n = length(X[,1]);   d = length(X[1,]) ;   sdv = sd(X[,1])
  f = prod(dnorm(x,sd=sdv)) ;  dum = (sum(x^2)/sdv^4 -d/sdv^2)^2/d
  c0 = (dum*f*(2*sqrt(pi))^2)^(-1/(d+4));  c0*n^(-1/(d+4))
}
```

function_prediction_new.R – one step ahead prediction

```
rollingPrediction =  function (W,n1,ARMAorder,step,cond)
# W = observed values (1:n) [vector]
```

```
# n1 = last observed entry [numeric]
# ARMAorder = c(p,d,q)
# step = # of estimation steps-ahead [numeric]
# cond = vectors of T/F for what method to be estimated
# --> returns <-- a list of the observations and estimation vectors and RMSE values for all methods
{
 ##initilization
 predP=rep(0,(n-n1));predNP1=rep(0,(n-n1));predNP2 = rep(0,(n-n1));predNP3 = rep(0,(n-n1));
 predM1 = rep(0,(n-n1));predM2 = rep(0,(n-n1));predM3 = rep(0,(n-n1));
 errP=0; errNP1=0; errNP2=0; errNP3=0; errM1=0; errM2=0; errM3=0;
 rmseP=0; rmseNP1=0; rmseNP2=0; rmseNP3=0; rmseM1=0; rmseM2=0; rmseM3=0;

 if (cond[1] | cond[5] | cond[6] | cond[7] ) {                  #MA(7) + Mixt
   tm=proc.time();
   ex = execM(W,n1,ARMAorder,step)
   cat("Total time elapsed Parametric + Mixt:", (proc.time()-tm)[3],"\n")
   predP = ex$predP; predM1 = ex$predM1; predM2 = ex$predM2; predM3 = ex$predM3;
   rmseP = ex$rmseP; rmseM1 = ex$rmseM1; rmseM2 = ex$rmseM2; rmseM3 = ex$rmseM3}
 tm=proc.time()
 if (cond[2]) {                          #NP1
   ex = execNP(W, n1, lag=1, step);  predNP1 = ex$predNP; rmseNP1 = ex$rmseNP}
 if (cond[3]) {                          #NP2
   ex = execNP(W, n1, lag=2, step);   predNP2 = ex$predNP; rmseNP2 = ex$rmseNP}
 if (cond[4]) {                          #NP3
   ex = execNP(W, n1, lag=3, step);   predNP3 = ex$predNP; rmseNP3 = ex$rmseNP}
 cat("Total time elapsed Nonparametric:", (proc.time()-tm)[3])
 return(list(obs=W[n1:(n-1)],predP=predP, predNP1=predNP1,predNP2=predNP2, predNP3= predNP3,
predM1=predM1,predM2=predM2,predM3=predM3, rmseP=rmseP,
rmseNP1=rmseNP1,rmseNP2=rmseNP2,rmseNP3=rmseNP3,rmseM1=rmseM1,rmseM2=rmseM2,rmseM3=rms
eM3))
}

execM = function (W,n1,ARMAorder,step)
# W = observed values (1:n) [vector]
# n1 = last observed entry [numeric]
# ARMAorder = c(p,d,q)
# step = # of estimation steps-ahead [numeric]
# --> returns <-- a list of the estimation vectors and RMSE values for all methods
{
 n = length(W)
 if(n1>n) n1 = n
    predDummy = matrix(0,nrow=0,ncol=0);  predDummy = sapply((n1+1):n, function(i)
    predMixt(W[(i-n1):(i-1)],ARMAorder,step,paste("Param + Mixt:: Step",i-n1,"of",n-n1)))
 predDummy = matrix(unlist(predDummy), nrow=4, byrow=T) #convert to matrix
 predP = predDummy[1,]; predM1 = predDummy[2,]; predM2 = predDummy[3,]; predM3 =  predDummy[4,];
 errP=W[(n1+1):n]-predP; errM1=W[(n1+1):n]-predM1; errM2=W[(n1+1):n]-predM2;  errM3=W[(n1+1):n]-
predM3; rmseP=sqrt(mean(errP^2)); rmseM1=sqrt(mean(errM1^2)); rmseM2=sqrt(mean(errM2^2));
rmseM3=sqrt(mean(errM3^2));
return(list(predP=predP, predM1=predM1, predM2=predM2, predM3=predM3,
          rmseP=rmseP, rmseM1=rmseM1, rmseM2=rmseM2, rmseM3=rmseM3))
}

predMixt =  function (W,ARMAorder,step,message)
# W = subset before estimation (1:n1)
# ARMAorder = c(p,d,q)
# step = # of estimation steps-ahead
# message = message to appear in the console while running
# --> returns <-- a list of vectors with Parametric, Mixt(lag=1), Mixt(lag=2) and Mixt(lag=3) estimations
{
 cat(message);tm=proc.time(); n = length(W)
```

```
  model = arima(W,order=ARMAorder)
  predP = predict(model,n.ahead = step, se.fit=F)
  res = model$residuals
  res1 = res; res2 = res; res3 = res; predRes1 = numeric(0); predRes2 = numeric(0); predRes3 = numeric(0);
  for (i in (1:step))
  {
    predRes1 = c(predRes1,predNonparametric(res1[i:length(res1)],1,"","off"))
    predRes2 = c(predRes2,predNonparametric(res2[i:length(res2)],2,"","off"))
    predRes3 = c(predRes3,predNonparametric(res3[i:length(res3)],3,"","off"))
    res1 = c(res,predRes1); res2 = c(res,predRes2); res3 = c(res,predRes3)
  }
  predM1 = predP+predRes1;predM2 = predP+predRes2;predM3 = predP+predRes3;
  tme=(proc.time()-tm)[3];h=trunc(tme/(60*60));m=trunc(tme/60);s=trunc(tme-h*60*60-m*60)
  cat(".....Time elapsed: ",formatC(h, width=2, flag="0"),":", formatC(m, width=2, flag="0"), ":", formatC(s,
width=2, flag="0"),sep="","\n")
  return(list(predP = predP, predM1 = predM1, predM2 = predM2, predM3 = predM3))
}

predNonparametric =  function (W,lag,message,sw)
# W = subset before estimation (1:n1)  [vector]
# lag = lag to be taken into account for the estimation [numeric]
# message = message to appear in the console while running [string]
# sw = "on"/"off" to show or not the message [string]
# --> returns <-- a value for nonparametric prediction [number]
{
  if (sw=="on")
  {
    cat(message); tm=proc.time()
  }
  n = length(W);  x = W[(n):(n-lag+1)]
  X = matrix(nrow=(n-lag),ncol=lag)
  for (j in (1:lag))
    X[,j] = W[(lag+1-j):(n-j)]
  h =  bandwidth.x(x,X)
  pred = rn(x,h,X,W[(lag+1):n])
  if (sw=="on")
  {
    tme=(proc.time()-tm)[3];h=trunc(tme/(60*60));m=trunc(tme/60);s=trunc(tme-h*60*60-m*60)
    cat(".....Time elapsed: ",formatC(h, width=2, flag="0"),":",formatC(m, width=2, flag="0"), ":",formatC(s,
width=2, flag="0"),sep="","\n")
  }
  return(pred)
}

execNP = function(W, n1, lag, step)
# W = observed values (1:n) [vector]
# n1 = last observed entry [numeric]
# lag = lag to be taken into account for the estimation [numeric]
# step = # of estimation steps-ahead [numeric]
# --> returns <-- a list of the NonParametric estimation vector and RMSE value
{
  n = length(W)
  if(n1>n) n1 = n
  predNP = sapply((n1+1):n, function(i)
      predNonparametric( W[(i-n1):(i-1)], lag, paste("Nonparam. d =",lag,":: Step",i-n1,"of",n-n1),"on"))
  errNP=W[(n1+1):n]-predNP;  rmseNP=sqrt(mean(errNP^2));
  return(list(predNP=predNP, rmseNP=rmseNP))
}

rollingPrediction_NP_ARMA =  function (W,n1,step)
```

```
# W = observed values (1:n) [vector]
# n1 = last observed entry [numeric]
# step = # of estimation steps-ahead [numeric]
# --> returns <-- a list of the observatins and estimation vectors and RMSE values for all methods
{
 ##initilization
 predNP1 = rep(0,(n-n1));predNP2 = rep(0,(n-n1));predNP3 = rep(0,(n-n1));
 predM1 = rep(0,(n-n1));predM2 = rep(0,(n-n1));predM3 = rep(0,(n-n1));
 errNP1=0; errNP2=0; errNP3=0; errM1=0; errM2=0; errM3=0;
 rmseNP1=0; rmseNP2=0; rmseNP3=0; rmseM1=0; rmseM2=0; rmseM3=0;
 tm=proc.time()
 ex = execM_NP_ARMA(W,n1,step)
 cat("Total time elapsed NonParametric + Mixt:", (proc.time()-tm)[3],"\n")
 predNP1 = ex$predNP1; predNP2 = ex$predNP2; predNP3 = ex$predNP3;
 predM1 = ex$predM1; predM2 = ex$predM2; predM3 = ex$predM3;
 rmseNP1 = ex$rmseNP1; rmseNP2 = ex$rmseNP2; rmseNP3 = ex$rmseNP3;
 rmseM1 = ex$rmseM1; rmseM2 = ex$rmseM2; rmseM3 = ex$rmseM3
 return(list(obs=W[(n1+1):n],predNP1=predNP1,predNP2=predNP2,predNP3=predNP3, predM1=predM1,
predM2=predM2,predM3=predM3, rmseNP1=rmseNP1,rmseNP2=rmseNP2,
rmseNP3=rmseNP3,rmseM1=rmseM1,rmseM2=rmseM2,rmseM3=rmseM3))
}

execM_NP_ARMA = function (W,n1,step)
# W = observed values (1:n) [vector]
# n1 = last observed entry [numeric]
# ARMAorder = c(p,d,q)
# step = # of estimation steps-ahead [numeric]
# --> returns <-- a list of the estimation vectors and RMSE values for all methods
{
 n = length(W)
 if(n1>n) n1 = n
 predNP1 = execNP(W, n1, lag=1, step)$predNP; predNP2 = execNP(W, n1, lag=2, step)$predNP;  predNP3 =
execNP(W, n1, lag=3, step)$predNP; errNP1 = W[(n1+1):n] - predNP1; errNP2 = W[(n1+1):n] - predNP2;
errNP3 = W[(n1+1):n] - predNP3; predPRes = predARMA(errNP1,c(0,0,2),step,""); predRes1 = errNP1 -
predPRes$residuals; predPRes = predARMA(errNP2,c(0,0,2),step,""); predRes2 = errNP2 -
predPRes$residuals;
 predPRes = predARMA(errNP3,c(0,0,2),step,""); predRes3 = errNP3 - predPRes$residuals;
 predM1 = predNP1+predRes1; predM2 = predNP2+predRes2; predM3 = predNP3+predRes3;

 errM1=W[(n1+1):n]-predM1; errM2=W[(n1+1):n]-predM2; errM3=W[(n1+1):n]-predM3;
 rmseNP1=sqrt(mean(errNP1^2));rmseNP2=sqrt(mean(errNP2^2)); rmseNP3=sqrt(mean(errNP3^2));
rmseM1=sqrt(mean(errM1^2)); rmseM2=sqrt(mean(errM2^2)); rmseM3=sqrt(mean(errM3^2));
 return(list(predNP1=predNP1, predNP2=predNP2, predNP3=predNP3, predM1=predM1, predM2=predM2,
predM3=predM3,  rmseNP1=rmseNP1, rmseNP2=rmseNP2, rmseNP3=rmseNP3, rmseM1=rmseM1,
rmseM2=rmseM2, rmseM3=rmseM3))
}

predARMA =  function (W,ARMAorder,step,message)
# W = subset before estimation (1:n1)  [vector]
# ARMAorder = c(p,d,q)
# step = # of estimation steps-ahead
# message = message to appear in the console while running
# --> returns <-- a list of vectors with Parametric, Mixt(lag=1), Mixt(lag=2) and Mixt(lag=3) estimations
{
 cat(message);tm=proc.time();
 n = length(W)
 model = arima(W,order=ARMAorder)
 predP = predict(model,n.ahead = step, se.fit=F)
 cat("Time elapsed: ", (proc.time()-tm)[3],"\n")
 return(list(predP=predP, residuals = model$residuals))
```

```
}

## Multi-step ahead prediction
library(Hmisc); library(tseries)
##  prepare data and working horizon
newSpeed = read.table("newSpeed2.csv")
source("functions_prediction_new2.R")
n = length(newSpeed[,1])
dT = 355;  ## days of training
dE = 5;  ## days of estimation
n1 = 0; n2 = dT*144; n3 = (dT+dE)*144;
data = newSpeed[(n1+1):n2,];
delay = 0; d = 3; sigma=0.01

##  train nonparametric model
est_NP = NonParametric(sigma,data, dE*24*6, delay,d);
## compute ARMA on nonparametric residuals
res = newSpeed[(n2+1):n3,] - est_NP;
n_res = length(res);dres = diff(res);
par(mfrow=c(1,2));acf(dres);pacf(dres)
dev.off();
model = garchFit(~arma(1,0)+garch(1,1), diff(res), trace=F);
q = predict(model, dE*24*6, se.fit=F)
q = diffinv(c(diff(res),q[,1]),xi=res[1])[(n_res+1):(n_res+dE*24*6)]

## find nonparametric estimates + mixt1 estimates
data = newSpeed[(n1+dE*144+1):n3,];
est_M_NP = numeric(0);mixt1 = numeric(0);
for (i in 1:(dE*24*6))
{
  cat("Step ", i, " out of ", (dE*24*6), sep="", "\n")
  est_M_NP[i] = NonParametric(sigma,data, 1, delay,d);
  mixt1[i] = est_M_NP[i] + q[i];
  data = c(data, est_M_NP[i]);
}
est_M_NP1 = est_M_NP[1:144]; est_M_NP3 = est_M_NP[1:(3*144)]; est_M_NP5 = est_M_NP
mixt1_1 = mixt1[1:144]; mixt1_3 = mixt1[1:(3*144)]; mixt1_5 = mixt1;
rmseNP1 = sqrt(sum((newSpeed[(n3+1):(n3+1*144),] - est_M_NP1)^2)/(1*24*6))
rmseNP3 = sqrt(sum((newSpeed[(n3+1):(n3+3*144),] - est_M_NP3)^2)/(3*24*6))
rmseNP5 = sqrt(sum((newSpeed[(n3+1):(n3+5*144),] - est_M_NP5)^2)/(5*24*6))
rmseMixt1_1 = sqrt(sum((newSpeed[(n3+1):(n3+1*144),] - mixt1_1)^2)/(1*24*6))
rmseMixt1_3 = sqrt(sum((newSpeed[(n3+1):(n3+3*144),] - mixt1_3)^2)/(3*24*6))
rmseMixt1_5 = sqrt(sum((newSpeed[(n3+1):(n3+5*144),] - mixt1_5)^2)/(5*24*6))

##  train ARMA model
data = newSpeed[(n1+1):n2,];
par(mfrow=c(1,2)); acf(diff(data));pacf(diff(data));
dev.off()
est_ARMA = ParametricARMA017(data, dE*24*6, order=c(0,1,7)); ## ARIMA(0,1,7)

## compute NP on ARMA residuals
res = newSpeed[(n2+1):n3,] - est_ARMA

## find ARMA estimates + mixt2 estimates
data = newSpeed[(n1+dE*144+1):n3,];
par(mfrow=c(1,2)); acf(diff(data));pacf(diff(data));
dev.off()
est_ARMA = ParametricARMA017(data, (dE*24*6), order=c(0,1,7)); ## ARIMA(0,1,7)
mixt2 = numeric(0);est_M_ARMA = numeric(0);
for (i in 1:(dE*24*6))
```

```
{
  cat("Step ", i, " out of ", (dE*24*6), sep="", "\n")
  est_M_ARMA[i] = NonParametric(sigma, res, 1, delay,d);
  res = c(res, est_M_ARMA[i]);
}
mixt2 = est_ARMA + est_M_ARMA;
est_ARMA1 = est_ARMA[1:144]; est_ARMA3 = est_ARMA[1:(3*144)]; est_ARMA5 = est_ARMA;
mixt2_1 = mixt2[1:144]; mixt2_3 = mixt2[1:(3*144)]; mixt2_5 = mixt2;

rmseARMA1 = sqrt(sum((newSpeed[(n3+1):(n3+1*144),] - est_ARMA1)^2)/(1*24*6))
rmseARMA3 = sqrt(sum((newSpeed[(n3+1):(n3+3*144),] - est_ARMA3)^2)/(3*24*6))
rmseARMA5 = sqrt(sum((newSpeed[(n3+1):(n3+5*144),] - est_ARMA5)^2)/(5*24*6))
rmseMixt2_1 = sqrt(sum((newSpeed[(n3+1):(n3+1*144),] - mixt2_1)^2)/(1*24*6))
rmseMixt2_3 = sqrt(sum((newSpeed[(n3+1):(n3+3*144),] - mixt2_3)^2)/(3*24*6))
rmseMixt2_5 = sqrt(sum((newSpeed[(n3+1):(n3+5*144),] - mixt2_5)^2)/(5*24*6))

###### plots
#### 1 DAY
plot(est_ARMA1, type="l", ylim = c(-10,30), col=5, lwd=2, lty=5, main="Long-run estimations (1 day-ahead)
::: d=3")
lines(est_M_NP1,col=2, lty=2, lwd=2); lines(mixt1_1,col=3, lty=3, lwd=2);
lines(mixt2_1,col=4, lty=4, lwd=2); lines(newSpeed[(n3+1):(n3+1*144),],col=1)
legend(0,25, bty="n", x.intersp  = 0.15, y.intersp  = 0.15,
      legend=c("ARMA", "NP", "Mixt1 (NP + residuals-ARMA)","Mixt2 (ARMA + residuals-NP)","real data"),
      col=c(5,2,3,4,1), lty=c(5,2,3,4,1), seg.len=rep(0.7,5), lwd=rep(2,5))

#### 5 DAYS
plot(est_ARMA5, type="l", ylim = c(-10,30), col=5, lwd=2, lty=5, main="Long-run estimations (5 days-ahead)
::: d=3")
lines(est_M_NP5,col=2, lty=2, lwd=2);lines(mixt1_5,col=3, lty=3, lwd=2);
lines(mixt2_5,col=4, lty=4, lwd=2); lines(newSpeed[(n3+1):(n3+5*144),],col=1)
legend(0,25, bty="n", x.intersp  = 0.15, y.intersp  = 0.15,
      legend=c("ARMA", "NP", "Mixt1 (NP + residuals-ARMA)","Mixt2 (ARMA + residuals-NP)","real data"),
      col=c(5,2,3,4,1), lty=c(5,2,3,4,1), seg.len=rep(0.7,5), lwd=rep(2,5))
```

function_prediction_new2.R – functions for multistep ahead prediction

```
NonParametric = function (sigma,data, dF, delay,d)
{
  est = numeric(0)
  for (j in (1:dF))
  {
    cat("Step ", j, " out of ",dF, sep="", "\n");
    W = data[j:length(data)];
    nw = length(W);
    x = W[(nw-delay):(nw-delay-d+1)];
    X = matrix(nrow=(nw-d-delay),ncol=d);
    #sigma=numeric(0);
    for (i in 1:d)
    {
       X[,i] = W[(d+1-i):(nw-i-delay)];
       #sigma[i] = bw.ndr(X[,i])
    }
    Y = W[(d+1+delay):nw];
    est[j] = rNW(x,sigma,X,Y);
    data = c(data,est[j]);
  }
  return(est)
}

## Nadaraya-Watson estimator
```

```
rNW = function (x,sigma,X,Y)
{
  n = length(Y);
  dum = numeric(0);
  dum = sapply(1:n,function (i) K((x-X[i,])/sigma))
  s = sum(dum);
  if (s!=0)
   r = sum(Y*dum)/s
  else
   r = 0
  return(r)
}

ParametricARMA017 = function (data, dF, order)
{
  est = numeric(0);
  W = data; nw = length(W);
  model = garchFit(~arma(0,7)+garch(1,1), data=diff(W), trace=F)
  est = predict(model,n.ahead = dF, se.fit=F)
  est = diffinv(c(diff(W),est[,1]),xi=W[1])[(nw+1):(nw+dF)]
  return(est)
}
```

Figure 35. / pag. 44

```
X = W[n1:(n1+144)]; Y = d1_predNP1; ind=numeric(0)
for (i in 1:144)
{
  p = sqrt((X[i+1]-X[i])^2);  m = sqrt((X[i+1]-Y[i])^2);  ind[i] = p/m
}
plot(ind,type="l", ylab="Ratio", xlab="Time"); abline(h=1, col=2)
a = ind[ind<1];length(a)

X = W[n1:(n1+5*144)]; Y = d5_predM1_2; ind=numeric(0)
for (i in 1:720)
{
  p = sqrt((X[i+1]-X[i])^2);   m = sqrt((X[i+1]-Y[i])^2);   ind[i] = p/m
}
plot(ind,type="l", ylab="Ratio", xlab="Time"); abline(h=1, col=2)
a = ind[ind<1];length(a)
```

confidence intervals – Figure 37-36 / pag. 47

```
library(boot)
stat = function(x,i) x[i]   ## statistic use for bootstrap
CI = function(obs, est)
{
  res = obs - est
  b = boot(res,stat,1000)    ### bootstrapped samples of residuals
  L = numeric(0); U = numeric(0);
  for(i in 1:length(est))
  {
    L[i] = quantile(b$t[,i],probs=0.025); U[i] = quantile(b$t[,i],probs=0.975)
  }
  return(list(b=b$t,L=L,U=U))
}

ws_obs_1day = newSpeed2[(52560-5*144+1):(52560-4*144),]
ws_obs_5day = newSpeed2[(52560-5*144+1):52560,]
wp_obs_1day = data105_NP_imp[(52560-5*144+1):(52560-4*144),11]
wp_obs_5day = data105_NP_imp[(52560-5*144+1):52560,11]
```

```
CI_ws_short_1day = CI(ws_obs_1day,ws_short_1day[,1])
CI_ws_short_5day = CI(ws_obs_5day,ws_short_5days[,1])
CI_ws_long_1day = CI(ws_obs_1day,ws_long_1day[,1])
CI_ws_long_5day = CI(ws_obs_5day,ws_long_5day[,1])

CI_wp_short_1day = CI(wp_obs_1day,wp_short_1day[,1])
CI_wp_short_5day = CI(wp_obs_5day,wp_short_5days[,1])
CI_wp_long_1day = CI(wp_obs_1day,wp_long_1day[,1])
CI_wp_long_5day = CI(wp_obs_5day,wp_long_5day[,1])

## WS short 1 day
win.graph()
plot(ts(CI_ws_short_1day$b[1,]+ws_short_1day[,1]),col="grey", ylim=c(0,6), ylab = "Wind Speed")
for(i in 2:144){
        lines(ts(CI_ws_short_1day$b[i,]+ws_short_1day[,1]),col="grey")
}
lines(ts(CI_ws_short_1day$L+ws_short_1day[,1]),col=2)
lines(ts(CI_ws_short_1day$U+ws_short_1day[,1]),col=2)
lines(ws_obs_1day,col=4); lines(ws_short_1day[,1], col=6)
legend(0,6, bty="n", x.intersp  = 0.8, y.intersp  = 0.8,
    legend=c("Lower/Upper bounds", "Observed","Estimated"),
    col=c(2,4,6), seg.len=rep(0.7,3), lwd=rep(2,3))
```

BIBLIOGRAPHY

[1] E. Golding, The Generation of Electricity by Wind Power, New York: Halsted Press, 1976.

[2] G. L. Johnson, Wind Energy Systems, Electronic Version, 2006.

[3] "International Energy Agency," [Online]. Available: http://www.iea.org.

[4] L. Freris and D. Infield, Renewable Energy in Power Systems, United Kingdom: Wiley, 2008.

[5] European Union, *Climate change: Commission welcomes final adoption of Europe's climate and energy package,* 2008.

[6] Commission of the European Communities, *Second Strategic Energy Review: AN EU ENERGY SECURITY AND SOLIDARITY ACTION PLAN,* vol. I, Brussels, 2008.

[7] The European Wind Energy Association, "Wind in power - 2011 European statistics," 2011.

[8] D. Bourlis and J. Bleijs, "A wind speed estimation method using adaptive Kalman filtering for a variable speed stall regulated wind turbine," in *IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems*, Singapore, 2010.

[9] R. G. Kavasseri and K. Seetharaman, "Day-ahead wind speed forecasting using f-ARIMA models," *Renewable Energy,* vol. 34, no. 5, pp. 1388-1393, May 2009.

[10] M. Miranda and R. Dunn, "One-hour-ahead wind speed prediction using a Bayesian methodology," in *IEEE Power Engineering Society General Meeting*, Montreal, Quebec, 2006.

[11] J. Bjørnar Bremnes, "Probabilistic wind power forecasts using local quantile regression," *Wind Energy,* vol. 7, no. 1, pp. 47-54, 2004.

[12] L. McArthur, "Empirical Orthogonal Function Analysis of wind farm power output," in *19th International Congress on Modelling and Simulation (MODSIM)*, Perth, Australia, 2011.

[13] A. More and M. Deo, "Forecasting wind with neural networks," *Marine Structures,* vol. 16, no. 1, pp. 35-49, January 2003.

[14] H. Pousinho, V. Mendes and J. Catalã, "Neuro-Fuzzy Approach to Forecast Wind Power in Portugal," in *International Conference on Renewable Energies and Power Quality*, Granada, Spain, 2010.

[15] "The Wind Power," [Online]. Available: http://www.thewindpower.net.

[16] "AEMET," [Online]. Available: http://www.aemet.es.

[17] I. Sánchez, "Short-term prediction of wind energy production," *International Journal of Forecasting,* vol. 22, no. 1, pp. 43-56, 2006.

[18] B. W. Silverman, Density Estimation for Statistics and Data Analysis, London: Chapman & Hall, 1998.

[19] J. S. Simonoff, Smoothing Methods in Statistics, Springer-Verlag , 1996.

[20] H. A. Sturges, "The choice of a class interval," *Journal of the American Statistical Association,* vol. 21, no. 153, pp. 65-66, March 1926.

[21] D. P. Doane, "Aesthetic Frequency Classifications," *The American Statistician,* vol. 30, no. 4, pp. 181-183, November 1976.

[22] D. W. Scott, "On Optimal and Data-Based Histograms," *Biometrika,* vol. 66, no. 3, pp. 605-610, December 1979.

[23] "On the histogram as a density estimator:L2 theory," *Probability Theory and Related Fields,* vol. 57, no. 4, pp. 453-476, December 1981.

[24] G. Upton and I. Cook, Understanding Statistics, Oxford University Press, 1996, p. p.55.

[25] R. S. TSAY, Analysis of Financial Time Series, Second Edition ed., New Jersey: John Wiley & Sons, 2005.

[26] C. M. Jarque and A. K. Bera, "A Test for Normality of Observations and Regression Residuals," *International Statistical Review,* vol. 55, no. 2, pp. 163-167, August 1987.

[27] M. H. Albuhairi, "Assessment and Analysis of Wind Power Density in Taiz- Republic of Yemen," *Ass. Univ. Bull. Environ. Res,* vol. 9, no. 2, pp. 13-21, October 2006.

[28] "Wind power potential and characteristic analysis of the Pearl River Delta region, China," *Renewable Energy ,* vol. 31, no. 6, pp. 739-756, May 2006.

[29] "RenewableUK," [Online]. Available: http://www.bwea.com.

[30] C. Potter, H. Gil and J. McCaa, "Wind Power Data for Grid Integration Studies," in *Power Engineering Society General Meeting, 2007. IEEE*, Tampa, Florida, 2007.

[31] H. Nielsen, T. Nielsen, H. Madsen, G. Giebel, J. Badger, L. Landbergt, K. Sattler, L. Voulund and J. Tofting, "From wind ensembles to probabilistic information about future wind power production – results from an actual application," in *Probabilistic Methods Applied to Power Systems 2006*, 2006.

[32] A. Kusiak, H. Zheng and Z. Song, "Models for monitoring wind farm power," *Renewable Energy,* vol. 34, no. 3, pp. 583-590, March 2009.

[33] D. Hawkins, Identification of outliers, London: Chapman and Hall, 1980.

[34] H. J. Motulsky, "Detecting outliers when fitting data with nonlinear regression – a new method based on robust nonlinear regression and the false discovery rate," *BMC Bioinformatics,* vol. 7, no. 123, March 2006.

[35] L. A. Osadciw, Y. Yan, X. Ye, G. Benson and E. White, "Wind Turbine Diagnostics Based on Power Curve Using Particle Swarm Optimization," in *Wind Power Systems*, Springer, 2010, pp. 151-165.

[36] C. M. Judd, G. H. McClelland and C. S. Ryan, Data analysis: A model comparison approach, Second Edition ed., San Diego: Routledge, 2008.

[37] J. Graham, "Missing Data Analysis: Making It Work in the Real World," *Annual Review of Psychology,* vol. 60, pp. 549-576, 2009.

[38] A. Gelman and J. Hill, "Chapter 25 Missing-data imputation," in *Data Analysis Using Regression And Multilevel/Hierarchical Models*, Cambridge, Cambridge University Press, 2007, p. 625.

[39] L. Roderick J. A. and R. Donald B., Statistical Analysis with Missing Data, Second Edition ed., John Wiley & Sons, 2002.

[40] J. Fox, Applied regression analysis and generalized linear models, Second edition ed., Los Angeles: SAGE Publications, Inc, 2008.

[41] G. E. P. Box, G. M. Jenkins and G. C. Reinsel, Time series analysis: Forecasting and control, Fourth edition ed., John Wiley & Sons , 2008.

[42] T. Bollerslev, "Generalized Autoregressive Conditional Heteroskedasticity," *Journal of Econometrics,* vol. 31, pp. 307-327, 1986.

[43] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society,* vol. 39, no. 1, pp. 1-38, 1977.

[44] J. Fan and Q. Yao, Nonlinear Time Series - Nonparametric and Parametric Methods, New York: Springer, 2003, p. 551.

[45] T. Hastie and R. Tibshirani, "Generalized Additive Models," *Statistical Science,* vol. 1, no. 3, pp. 297-318, 1986.

[46] S. N. Wood and N. H. Augustin, "GAMs with integrated model selection using penalized regression splines and applications to environmental modelling," *Ecological Modelling,* vol. 157, no. 2-3, pp. 157-177, November 2002.

[47] K. Tharmaratnam, G. Claeskens, C. Croux and M. Salibian-Barrera, "S-estimation for penalized regression splines," *Journal of Computational and Graphical Statistics,* vol. 19, no. 3, pp. 609-625, 2010.

[48] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Annals of Mathematical Statistics,* vol. 27, no. 3, pp. 832-837, 1956.

[49] E. Parzen, "On estimation of a probability density function and mode," *Annals of Mathematical Statistics,* vol. 33, no. 3, pp. 1065-1076, 1962.

[50] W. Härdle, M. Müller, S. Sperlich and A. Werwatz, Nonparametric and Semiparametric Models, Springer, 2004.

[51] E. A. Nadaraya, "On Estimating Regression," *Teor. Veroyatnost. i Primenen.,* vol. 9, no. 1, pp. 157-159, 1964.

[52] P. S. Cowpertwait and A. V. Metcalfe, Introductory Time Series with R, New York: Springer, 2009.

[53] J. D. Cryer and K.-S. Chan, Time Series Analysis With Applications in R, Second edition ed., Springer, 2008.

[54] S. Makridakis, S. C. Wheelwright and R. J. Hyndman, Forecasting. Methods and Applications, Wiley, 1998.

[55] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society,* vol. 26, no. 2, pp. 211-252, 1964.

[56] D. Hill, D. McMillan, K. Bell and D. Infield, "Application of Auto-Regressive Models to U.K. Wind Speed Data for Power System Impact Studies," *Sustainable Energy, IEEE Transactions on,* vol. 3, no. 1, pp. 134-141, 2012.

[57] M. Alexiadis, P. Dokopoulos, H. Sahsamanoglou and I. Manousaridis, "Short-term forecasting of wind speed and related electrical power," *Solar Energy,* vol. 63, no. 1, pp. 61-68, 1998.

[58] J. Taylor, P. McSharry and R. Buizza, "Wind Power Density Forecasting Using Ensemble Predictions and Time Series Models," *Energy Conversion, IEEE Transactions on,* vol. 24, no. 3, pp. 775-782, 2009.

[59] P. Louka, G. Galanis, N. Siebert, G. Kariniotakis, P. Katsafados, I. Pytharoulis and G. Kallos, "Improvements in wind speed forecasts for wind power prediction purposes using Kalman filtering," *Journal of Wind Engineering and Industrial Aerodynamics,* vol. 96, no. 12, pp. 2348-2362, 2008.

[60] J. M. Sloughter, T. Gneiting and A. E. Raftery, "Probabilistic Wind Speed Forecasting Using Ensembles and Bayesian Model Averaging," *Journal of the American Statistical Association,* vol. 105, no. 489, pp. 25-35, 2010.

[61] G. Kariniotakis, G. Stavrakakis and E. Nogaret, "Wind power forecasting using advanced neural networks models," *Energy Conversion, IEEE Transactions on,* vol. 11, no. 4, pp. 762-767 , 1996.

[62] M. Mabel and E. Fernandez, "Estimation of Energy Yield From Wind Farms Using Artificial Neural Networks," *Energy Conversion, IEEE Transactions on,* vol. 24, no. 2, pp. 459-464, 2009.

[63] Y. Liu, J. Shi, Y. Yang and W.-J. Lee, "Short-Term Wind-Power Prediction Based on Wavelet Transform–Support Vector Machine and Statistic-Characteristics Analysis," *Industry Applications, IEEE Transactions on,* vol. 48, no. 4, pp. 1136-1141, 2012.

[64] Q. LIN, G. YAN and C. YU, "Non-parameter kernel estimation of density function of cotton fiber length," *Journal of Textile Research,* vol. 29, no. 11, pp. 22-25, 2008.

[65] Y. Gong, D. Zhang and X. Wu, "Nonparametric Autoregression Prediction Model on Population Growth Rate," *Application of Statistics and Management,* vol. 26, no. 5, pp. 759-764, 2007.

[66] P. C. B. Phillips and J. Y. Park, "Nonstationary Density Estimation and Kernel Autoregression," *Yale University,* 1998.

[67] A. W. Bowman and A. Azzalini, Applied Smoothing Techniques for Data Analysis. The Kernel Approach with S-Plus Illustrations, Oxford: Clarendon Press, 1997.

[68] W. Härdle and R. Chen, "Nonparametric time series analysis, a selective review with examples," in *50th session of International Statistical Institute*, Beijing, 1995.

[69] D. Tjostheim and B. H. Auestad, "Nonparametric Identification of Nonlinear Time Series: Selecting Significant Lags," *Journal of the American Statistical Association,* vol. 89, no. 428, pp. 1410-1419, 1994.

[70] S. Dabo-Niang, C. Francq and J.-M. Zakoian, "Combining Nonparametric and Optimal Linear Time Series Predictions," *Journal of the American Statistical Association,* vol. 105, no. 492, pp. 1554-1565, 2010.

[71] E. Gilleland, "Confidence Intervals for Forecast Verification," Institute for Mathematics Applied to Geosciences (IMAGe) and Research Applications Laboratory, Boulder, Colorado, 2010.

[72] International Energy Agency, "IEA Wind 2010 Annual Report," IEA Wind, 2010.

[73] International Energy Agency, "IEA Wind Energy Annual Report 2009," IEA Wind, 2009.

[74] V. LoBrano, A. Orioli, G. Ciulla and S. Culotta, "Quality of wind speed fitting distributions for the urban area of Palermo, Italy," *Renewable Energy,* vol. 36, no. 3, pp. 1026-1039, 2011.

[75] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *IEEExplore,* vol. 90, no. 7, pp. 1151-1163, 2002.

[76] Y. M. Kantar and I. Usta, "Analysis of wind speed distributions: Wind distribution function derived from minimum cross entropy principles as better alternative to Weibull function,"

Energy Conversion and Management, vol. 49, no. 5, pp. 962-973, 2007.

[77] L. Wenyuan, Q. Zhilong and X. Xiaofu, "Estimating wind speed probability distribution using kernel density method," *Electric Power Systems Research,* vol. 81, no. 12, pp. 2139-2146, 2011.

[78] G. W. Snedecor and W. G. Cochran, Statistical Methods, Eighth Edition ed., Ames, Iowa: Iowa State University Press, 1989.

[79] A. Dempster, N. Laird and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society,* vol. 39, no. 1, pp. 1-38, 1977.

[80] W. Jeffrey C., "Multiple Imputation For Missing Data: What Is It And How Can I Use It?," American Educational Research, Chicago, 2003.

[81] W. Junger and A. Ponce de Leo, "Package mtsdi," February 2012. [Online]. Available: http://cran.r-project.org/web/packages/mtsdi/mtsdi.pdf.

[82] B. W. Silverman, "Spline Smoothing: The Equivalent Variable Kernel Method," *The Annals of Statistics,* vol. 12, no. 3, pp. 898-916, 1984.

[83] M. Holmes, A. Gray and C. Isbell, "Fast Nonparametric Conditional Density Estimation," in *Proceedings of the Twenty-Third Conference Annual Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon, 2007.

[84] M. Rosenblatt, "Conditional probability density and regression estimators," in *Multivariate Analysis II*, New York, 1969.

[85] R. J. Hyndman, D. M. Bashtannyk and G. K. Grunwald, "Estimating and Visualizing Conditional Densities," *Journal of Computational and Graphical Statistics,* vol. 5, no. 4, pp. 315-336, December 1996.

[86] J. Fan, Q. Yao and H. Tong, "Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems," *Biometrika ,* vol. 83, no. 1, pp. 189-206, 1996.

[87] D. M. Bashtannyk and R. J. Hyndman, "Bandwidth Selection for Kernel Conditional Density Estimation," *Computational Statistics & Data Analysis,* vol. 36, pp. 279-298, 2000.

[88] J. Juban, L. Fugon and G. Kariniotakis, "probabilistic short-term wind power forecasting based on kernel density estimators," in *European Wind Energy Conference and exhibition, EWEC 2007* , Milan, 2007.

[89] S. J. Sheather and M. C. Jones, "A reliable data-based bandwidth selection method for kernel density estimation," *Journal of the Royal Statistical Society,* vol. 53, no. 3, pp. 683-690, 1991.

[90] D. W. Scott, Multivariate Density Estimation: Theory, Practice and Visualization, Wiley, 1992.