UNIVERSITAT ROVIRA I VIRGILI

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

E.T.S. d'Enginyeria URV F. Matemàtiques UB F. d'Informatica FIB

MASTER THESIS

Exploring Machine Learning Models for Wind Speed Prediction

Author:
Ing. Maritza PRIETO EMHART

Advisor:
Dr. Miquel SÀNCHEZ-MARRÈ
Department of Computer Science
Universitat Politècnica de Catalunya

A thesis submitted in fulfillment of the requirements for the degree of Master in Artificial Intelligence

January 14, 2018

Abstract

Exploring Machine Learning Models
for
Wind Speed Prediction

by Ing. Maritza PRIETO EMHART

The aim of this work present a comprehensive exploration of machine learning models and compare their performance for wind speed prediction.

The prediction is based on variables from atmospheric reanalysis data from a specific wind farm located in Spain as predictive inputs for the system. The ERA-Interim reanalysis data from the European Center for Medium-Range Weather Forecasts has been the source for obtaining the explanatory variables in this work.

Specifically the experiments include testing and selecting different global models which we later classify as single and combined models, and local models which we construct by means of the creation of clusters of the predictive variables dataset.

The goal is to explore which are the best model among all ones trying to get the estimation of the Wind Speed Prediction, and extract some conclusions from all the study.

Experimental evaluation of the prediction system was performed in real data from the mentioned wind farm , obtaining excellent prediction results when applying combined models.

This work also outlines the combined forecasting approaches and presents an up-to date annotated bibliography of the wind forecasting literature.

Furthermore, the thesis also points out the possible further research directions of combined techniques so as to help the researchers in the field to develop more effective wind speed forecasting methods.

Acknowledgements

I would first like to thank my thesis advisor Dr. Miquel SÀNCHEZ-MARRÈ of the Department of Computer Science at Universitat Politècnica de Catalunya. The door of Prof. Miquel office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right the direction whenever he thought I needed it.

Furthermore, my advisor and myself want to thank the support from Spanish Research Network "Diversificación en Aprendizaje Máquina y Aplicaciones" (DAMA) (TIN2015-70308-REDT, MINECO /FEDER EU).

Finally, I must express my very profound gratitude to my parents, sister and to my husband for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Ing. Maritza PRIETO EMHART

Contents

Abstract iii

Acknowledgements v

1 Introduction 1
- 1.1 Motivation 1
- 1.2 Goal 2

2 State of Art 3
- 2.1 Wind Prediction Problem 3
  - 2.1.1 Wind Farms 3
  - 2.1.2 Wind Prediction 3
    - 2.1.2.1 Time scales of forecasts 3
  - 2.1.3 Reason for wind forecasts 4
  - 2.1.4 Relationship between wind speed and wind power 5
- 2.2 Wind Prediction Methodology 6
  - 2.2.1 Prediction of meteorological variables 6
  - 2.2.2 Main Approaches 8
    - 2.2.2.1 Physical forecasting approach or Mechanistic model-driven 8
    - 2.2.2.2 Statistical forecasting approach or Data-driven 9
    - 2.2.2.3 Comparison 11
  - 2.2.3 Combination Approaches 12
    - 2.2.3.1 Weighting-based combined approaches 14
    - 2.2.3.2 Other combined approaches 14
    - 2.2.3.3 Comparison 16
- 2.3 Atmospheric Reanalysis Data 19
  - 2.3.1 European Centre for Medium-Range Weather Forecasts 19
    - 2.3.1.1 ERA-Interim 20
- 2.4 Machine Learning techniques used 20
  - 2.4.1 Unsupervised methods 21
    - 2.4.1.1 Feature selection/weighting methods 21
    - 2.4.1.2 Clustering methods 23
  - 2.4.2 Supervised methods 24
    - 2.4.2.1 Single predictive methods 24
    - 2.4.2.2 Combined predictive methods 27
- 2.5 Tools and Technologies used 31
  - 2.5.1 Python 31
  - 2.5.2 GESCONDA 32
- 2.6 Evaluation metrics 33
  - 2.6.1 RMSE: Root mean squared error 33
  - 2.6.2 MAE: Mean absolute error 34
  - 2.6.3 R2: Determination Coefficient 34

2.6.4 EV: Explained variance 35

3 Methodology description 37
3.1 General Idea 37
3.2 Data Science Methodology 37
3.3 Business and Data understanding 38
3.4 Data preparation 38
3.4.0.1 Data Cleaning 39
3.4.0.2 Data Transformation 39
3.4.0.3 Data selection 39
3.5 Modeling 40
3.5.1 Global models 40
3.5.1.1 Single Models 41
3.5.1.2 Combined Models 42
3.5.2 Local models 43
3.5.2.1 Clustering model 45
3.6 Evaluation 46

4 Methodology application 47
4.1 Data Understanding 47
4.2 Data Preparation 50
4.2.1 Data cleaning 50
4.2.2 Data Transformation 51
4.2.3 Data Selection 51
4.3 Modeling 52
4.4 Global Models 54
4.4.1 Single Models 54
4.4.1.1 K Nearest Neighbors Model 54
4.4.1.2 Artificial Neural Network Model 56
4.4.2 Combined Models 58
4.4.2.1 Random Forest Model 58
4.4.2.2 Ensemble Model 60
4.5 Local Models 61
4.5.1 Clusters 62
4.5.2 KNN Local models 65
4.5.3 ANN Local models 65
4.5.4 Random Forest Local models 66
4.5.5 Ensemble Local models 66

5 Discussion 69
5.1 Comparison of Single vs Combined Models 69
5.1.1 kNN 69
5.1.2 ANN 69
5.1.3 Random Forest 70
5.1.4 Ensemble 70
5.1.5 Global comparison 71
5.2 Comparison of Global vs Local Models 71
5.2.1 k Nearest Neighbors 71
5.2.2 ANN 72
5.2.3 Random Forest 73
5.2.4 Ensemble 74

5.3 Comparison among all Models 74
5.4 Benchmark comparison 76

6 Conclusions and Future Work 77
6.1 Conclusions 77
6.2 Future work 77

A Complementary Experiment results 79
A.1 Wind Speed Prediction using all 48F+M+H 79
A.1.1 RandomForest Results 79
A.1.2 kNN Results 79
A.1.3 ANN Results 79
A.2 Wind Speed Prediction with feature selection 79
A.2.1 RandomForest Results 80
A.2.2 kNN Results 80
A.2.3 ANN Results 81

Bibliography 83

List of Figures

2.1 Relationship between wind speed and wind power 6
2.2 Physical Approach 8
2.3 Statistical Approach 10
2.4 Artificial Neural Network Approach 11
2.5 Weighting-based combined approach 14
2.6 Preprocessing combined approach 15
2.7 Optimization combined approach 16
2.8 Error processing combined approach 16
2.9 UEB-1 algorithm 22
2.10 UEB-2 algorithm 23
2.11 RandomForest model 29
2.12 Bagging algorithm 30
2.13 Boosting algorithm 31
2.14 Stacking algorithm 31
2.15 Gesconda interface 33

3.1 Data Science methodology 38
3.2 Data preparation 39
3.3 Methodology proposed 40
3.4 Global modeling methodology proposed 41
3.5 Single models 41
3.6 Combined models 42
3.7 Ensemble methodology proposed 43
3.8 Methodology intuition 44
3.9 Local modeling methodology proposed 45

4.1 Wind farm location 47
4.2 Atmospheric Pressure levels 49
4.3 Wind Components 50
4.4 Data Missing values 51
4.5 Training process flowchart 53
4.6 Testing process flowchart 54
4.7 Global training process 54
4.8 KNN Test Prediction 56
4.9 ANN Test Prediction 58
4.10 RandomForest Test Prediction 60
4.11 Ensemble Test Prediction 61
4.12 Local training process 62
4.13 Train set clusters 63
4.14 Test set clusters 64
4.15 Sample of Cluster-1 data 65
4.16 Sample of Cluster-2 data 65
4.17 Sample of Cluster-3 data 65

List of Tables

2.1 Comparison of the main methods 13
2.2 Comparison of the combined approaches 18

4.1 Variables 48
4.2 Predictive Variables weights 52
4.3 RandomForest feature selection results 52
4.4 KNeighbors selection results 52
4.5 KNeighbors CV training results under different k 55
4.6 kNN test results for 20% unseen data 56
4.7 ANN CV training results under different units 57
4.8 ANN test results for 20% unseen data 58
4.9 RandomForest CV training results under different decision trees 59
4.10 RandomForest test results for 20% unseen data 60
4.11 Ensemble test results for 20% unseen data 61
4.12 Clusters partition 64
4.13 Local KNN model test results for 20% unseen data 66
4.14 Local ANN model test results for 20% unseen data 66
4.15 Local RF model test results for each cluster of 20% unseen data 66
4.16 Local Ensemble model test results for each cluster of 20% unseen data 67

5.1 Global KNN model test results for 20% unseen data 69
5.2 Global ANN model test results for 20% unseen data 70
5.3 Global RF model test results for 20% unseen data 70
5.4 Ensemble test results for 20% unseen data 70
5.5 Single vs Combined Models 71
5.6 Global KMM model test results for each cluster 72
5.7 Local KNN model test results for 20% unseen data 72
5.8 Global ANN model test results for each cluster 72
5.9 Local ANN model test results for 20% unseen data 73
5.10 Global RF model test results for each cluster 73
5.11 Local RF model test results for each cluster of 20% unseen data 74
5.12 Ensemble test results for 20% unseen data 74
5.13 Local Ensemble model test results for each cluster of 20% unseen data 74
5.14 Comparison Among all Models 75
5.15 Comparative best results of the wind power estimation by the ELM and GPR 76

A.1 RandomForest results using all 48F+M+H 79
A.2 kNN results using all 48F+M+H 80
A.3 ANN results using all 48F+M+H 80
A.4 Predictive Variables weights 81
A.5 RandomForest results with feature selection 81
A.6 kNN results with feature selection 82

A.7 ANN results with feature selection 82

List of Abbreviations

RF	**R**andom **F**orest
kNN	**K** **N**earest **N**eighbors
ANN	**A**rtificial **N**eural **N**etworks

Dedicated to Iker and Owen who are the engine of my life.

Chapter 1

Introduction

This chapter explains the main motivation of our work, as well as the main goal proposed for the work.

1.1 Motivation

Development of alternative energy sources has become a necessity as fossil energy resources are declining. At the same time, energy demand is rapidly increasing, putting the world on the verge of a global energy crisis. Moreover, the extensive use of conventional energy sources is polluting the environment and causing global warming. On the other hand, wind and other renewable energy sources are viable and clean alternatives to fossil fuels. Low operating cost and extensive availability make wind one of the most advantageous and effective renewable energy sources [1].

Wind energy is of vital importance among the low-carbon energy technologies, which has the potential to achieve sustainable energy supply and constitutes a keystone component for micro-grids in a way towards the smart grid infrastructure. However, stochastic and intermittent wind power generation poses a number of challenges to the large scale penetration of wind power. These wind-related uncertainties can put the system reliability and power quality at risk with the increasing penetration of wind power and thus, the main grid integration issues such as balanced management and reserve capacities can come into question [2, 3, 4]. Reducing the need for balancing energy and making the power generation scheduling and dispatch decisions can be realized with the help of wind speed and power generation forecasts [5]. Furthermore, the forecasts can play a vital role in keeping the costs competitive by reducing the need for wind curtailments and thereby, increasing revenue in electricity market operations [6]. However, the random and unstable characteristics of the wind make it considerably difficult to forecast the wind speed and power accurately. Hence, extensive efforts have been devoted for the developments and improvements of wind speed and power forecasting approaches by numerous energy and environment related research centers and universities [7].

In the literature, many forecasting approaches have been studied and proposed, each utilizing a different technique and performing well with a different prediction horizon. Recent studies in the area of wind prediction are predominantly focused on the short-term wind predictions ranging from minutes to a few days due to the importance of these data on power systems [7]. Especially day-ahead predictions are of significant interest for system operations such as scheduling, unit commitment and load following [8, 9]. However,it is generally difficult to accomplish such a long-term prediction and moreover, the approaches designed for long prediction horizons maybe deficient for shorter terms in terms of prediction performance. Following many studies in the wind forecasting field, it can be indicated that, to date, the targeted performance levels have not been attained with the individual models due to the fact that these models can not give satisfactory results for all situations. For instance, the physical models produce coarse predictions for short-term horizons while mostly outperform the other models in medium and long-term horizons. Also, Artificial Intelligence (AI) based models that rely on a large number of historical data for constructing an input/output mapping function can be less effective than some basic conventional statistical methods for certain application areas in the case of inadequate available information. Therefore, the approaches that incorporate the individually superior features of various forecasting models have emerged, called as hybrid models and combined models, in order to obtain an advanced forecasting method for higher accuracy levels and wider forecast horizons [7].

After evaluating the findings of the studies on the hybrid models, which will be detailed in next chapter, it can be concluded that these models do not generally contribute to the forecasting performance of the individual models considerably and they can even lead to poor performances under some circumstances [10]. On the other hand, combined forecasting methodologies, which follow a different approach and produce the final forecast generally from the weighting of the single approaches, can be a more viable solution for improving the accuracy of the individual models. To that end, the search effort has been recently oriented towards designing new combined algorithms as well as combination methods, which exploit different single prediction models and enhance the prediction performance while providing a reasonable computation time. A study on the classifying and summarizing of the combined methods, which might give an insight about the performance, superiority and application area of various algorithms, was presented in [7] which we will review in order to base this work.

1.2 Goal

With the continuous increase of wind power penetration in power systems, the problems caused by the volatile nature of wind speed and its occurrence in the system operations such as scheduling and dispatching have drawn attention of system operators, utilities and researchers towards the state-of-the-art wind speed and power forecasting methods. These methods have the required capability of reducing the influence of the intermittent wind power on system operations as well as of harvesting the wind energy effectively. In this context, combining different methodologies in order to circumvent the challenging model selection and take advantage of the unique strength of plausible models have recently emerged as a promising research area [7].

In this work, we present a comprehensive exploration about machine learning models including global and local, as well as combined models and how these models are constructed and affect the forecasting performance and final Wind Speed Prediction.

The goal is to explore which are the best models among all ones trying to get the estimation of the Wind Speed Prediction. Furthermore, we want to draw some conclusions from the study.

Chapter 2

State of Art

In this chapter the state of the art related to the work done is presented. First, the domain were the machine learning models has been applied i.e. the Wind Speed prediction problem is presented. Afterwards the different methodologies applied in the Wind Speed prediction problem are explained. Finally, main machine learning techniques are detailed.

2.1 Wind Prediction Problem

2.1.1 Wind Farms

A wind farm is the site where a group of wind turbines are installed for bulk electricity generation. Nameplate capacity of modern wind parks has increased by up to thousands of MWs. Wind farms can be categorized as *onshore* and *offshore* based on the location where wind turbines are installed.

Any wind power system typically comprises of wind turbines, generators, power transformers, and a connection to the power grid [11]. There are usually three kinds of wind systems: the constant speed wind turbine system with a standard squirrel cage induction generator (SCIG), the variable speed wind turbine system with a double-fed induction generator (DFIG), and the variable speed wind turbine with a full rated power electronics conversion system and a synchronous generator [1].

2.1.2 Wind Prediction

A wind power forecast corresponds to an estimate of the expected production of one or more wind turbines referred to as a wind farm. By production is often meant available power for wind farm considered (with units kW or MW depending on the wind farm nominal capacity). Forecasts can also be expressed in terms of energy, by integrating power production over each time interval [12].

2.1.2.1 Time scales of forecasts

The basic role of wind speed and power forecasting is to provide information about the wind speed and power that can be expected in the next few minutes, hours, or days. Based on power system operation requirements, the forecast can be divided into four different horizons: very short-term (few seconds to 30 min), short-term (30 min to 6 h), medium-term (6–24 h), and long-term (1–7 days). [13, 14] Very short-term forecasts are used for turbine control and load tracking. Short-term forecasts are utilized for preload sharing. Medium-term forecasts are used for power system management and energy trading. Long-term forecast are used for maintenance scheduling of the wind turbines.

For the last two possibilities, the temporal resolution of wind power predictions ranges between 10 minutes and a few hours (depending on the forecast length). Improvements of wind power forecasting has focused on using more data as input to the models involved, and on providing uncertainty estimates along with the traditionally provided predictions [12].

2.1.3 Reason for wind forecasts

As detailed in [12], in the electricity grid at any moment balance must be maintained between electricity consumption and generation - otherwise disturbances in power quality or supply may occur. Wind generation is a direct function of wind speed and, in contrast to conventional generation systems, is not easily dispatchable. Fluctuations of wind generation thus receive a great amount of attention. Variability of wind generation can be regarded at various time scales. First, wind power production is subject to seasonal variations, i.e. it may be higher in winter in Northern Europe due to low-pressure meteorological systems or it may be higher in summer in the Mediterranean regions owing to strong summer breezes. There are also daily cycles which may be substantial, mainly due to daily temperature changes. Finally, fluctuations are observed at the very short-term scale (at the minute or intra-minute scale). The variations are not of the same order for these three different timescales. Managing the variability of wind generation is the key aspect associated to the optimal integration of that renewable energy into electricity grids.

The challenges to face when wind generation is injected in a power system depend on the share of that renewable energy. It is a basic concept, the wind penetration which allows one to describe the share of wind generation in the electricity mix of a given power system. For Denmark, which is a country with one of the highest shares of wind power in the electricity mix, the average wind power penetration over the year is of 16-20% (meaning that 16-20% of the electricity consumption is met wind energy), while the instantaneous penetration (that is, the instantaneous wind power production compared to the consumption to be met at a given time) may be above 100%.

The transmission system operator (TSO) is responsible for managing the electricity balance on the grid: at any time, electricity production has to match consumption. Therefore, the use of production means is scheduled in advance in order to respond to load profiles. The load corresponds to the total electricity consumption over the area of interest. Load profiles are usually given by load forecasts which are of high accuracy. For making up the daily schedule, TSOs may consider their own power production means, if they have any, and/or they can purchase power generation from Independent Power Producers (IPPs) and utilities, via bilateral contracts or electricity pools. In the context of deregulation, more and more players appear on the market, thus breaking the traditional situation of vertically-integrated utilities with quasi local monopolies. Two main mechanisms compose electricity markets. The first one is the spot market where participants propose quantities of energy for the following day at a given production cost. An auction system permits to settle the electricity spot price for the various periods depending on the different bids. The second mechanism is the balancing of power generation, which is coordinated by the TSO. Depending on the energy lacks and surplus (e.g. due to power plant failures or to intermittence in the case of wind power installations), the TSO determines the penalties that will be paid by IPPs who missed in their obligations. In some cases, an intra-day market is also present, in order to take corrective actions.

In order to illustrate this electricity market mechanism, let us consider the Dutch electricity market. Market participants, referred to as Program Responsible Parties (PRPs), submit their price-quantity bids before 11 am for the delivery period covering the following day from midnight to midnight. The Program Time Unit (PTU) on the balancing market is of 15 minutes. Balancing of the 15-minute averaged power is required from all electrical producers and consumers connected to the grid, who for this purpose may be organized in subsets. Since these subsets are referred to as Programmes, balancing on the 15-minute scale is referred to as Programme Balance. Programme Balance now is maintained by using the production schedules issued the day before delivery and measurement reports (distributed the day after delivery). When the measured power is not equal to the scheduled power, the Programme Imbalance is the difference between the realised sum of production and consumption and the forecast sum of production and consumption. If only production from wind energy is taken into account, Programme Imbalance reduces to release wind production minus forecast wind production. The programme imbalance is the wind production forecast error.

Programme Imbalance is settled by the System Operator, with different tariffs for negative Programme Imbalance and positive Programme Imbalance. A positive Programme Imbalance indicates more energy actually produced than forecast. by wind energy the realised wind production is bigger than the forecast wind production. And vice versa, in the case of a negative Programme Imbalance by wind energy.

Note that the costs for positive and negative imbalances may be asymmetric, depending on the balancing market mechanism. In general, wind power producers are penalized by such market system since a great part of their production may be subject to penalties.

In parallel to be used for market participation, wind power forecasts may be used for the optimal combined operation of wind and conventional generation, wind and hydro-power generation, or wind in combination with some energy storage devices. They also serve as a basis for quantifying the reserve needs for compensating the eventual lacks of wind production.

2.1.4 Relationship between wind speed and wind power

Wind power is directly related to the wind speed through a so-called power curve 2.1. This is a simplified way of expressing the wind power in terms of atmospheric variables. Other atmospheric fields, such as wind shear, turbulence and air density have also impact on the actual power production for a given wind speed. However, for wind power verification wind speed is the most important parameter, because the bulk of the prediction error is caused by the wind speed prediction errors [15].

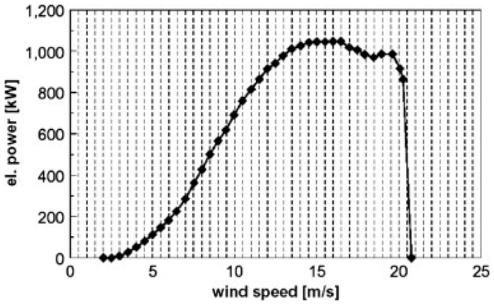


FIGURE 2.1: Relationship between wind speed and wind power (Extracted from [15]).

Given this relationship between wind speed and wind power generation, the efforts in our work will be focused on the prediction of the wind speed variable, because is the fundamental feature for the wind power generation.

2.2 Wind Prediction Methodology

As detailed in [12], several methods are used for short-term prediction of wind generation. The simplest ones are based on climatology or averages of past production values. They may be considered as reference forecasting methods since they are easy to implement, as well as benchmark when evaluating more advanced approaches. The most popular of these reference methods is certainly persistence. This naive predictor — commonly referred to as "what you see is what you get" — states that the future wind generation will be the same as the last measured value. Despite its apparent simplicity, this naive method might be hard to beat for look-ahead times up to 4–6 hours ahead.

Advanced approaches for short-term wind power forecasting necessitate predictions of meteorological variables as input. Then, they differ in the way predictions of meteorological variables are converted to predictions of wind power production, through the so-called power curve. Such advanced methods are traditionally divided into two groups:

- The first group, referred to as *physical approach*, focuses on the description of the wind flow around and inside the wind farm, and use the manufacturer's power curve, for proposing an estimation of the wind power output.
- The second group, referred to as *statistical approach* [1], concentrates on capturing the relation between meteorological predictions (and possibly historical measurements) and power output through statistical models whose parameters have to be estimated from data, without making any assumption on the physical phenomena.

2.2.1 Prediction of meteorological variables

As details in [12] Wind power generation is directly linked to weather conditions and thus the first aspect of wind power forecasting is the prediction of future values of

[1] Although in the literature they are named as statistical forecasting methods they include also machine learning techniques. Therefore, it would be better to name them as data-driven forecasting methods

the necessary weather variables at the level of the wind farm. This is done by using numerical weather prediction (NWP) models. Such models are based on equations governing the motions and forces affecting motion of fluids. From the knowledge of the actual state of the atmosphere, the system of equations allows to estimate what the evolution of state variables, e.g. temperature, velocity, humidity and pressure, will be at a series of grid points. The meteorological variables that are needed as input for wind power prediction obviously include wind speed and direction, but also possibly temperature, pressure and humidity. The distance between grid points is called the spatial resolution of the NWPs. The mesh typically has spacing that varies between few kilometers and up to 50 kilometers for mesoscale models. Regarding the time axis, the forecast length of most of the operational models today is between 48 and 172 hours ahead, which is in adequacy with the requirements for the wind power application. The temporal resolution is usually between 1 and 3 hours. NWP models impose their temporal resolution to short-term wind power forecasting methods since they are used as a direct input.

Predictions of meteorological variables are provided by meteorological institutes. Meteorologists employ atmospheric models for weather forecasts on short and medium term periods. An atmospheric model is a numerical approximation of the physical description of the state of the atmosphere in the near future, and usually is run on a supercomputer. Each computation starts with initial conditions originating from recent measurements. The output consists of the expected instantaneous value of physical quantities at various vertical levels in a horizontal grid and stepping in time up to several hours after initiation. There are several reasons why atmospheric models only approximate reality. First of all, not all relevant atmospheric processes are included in the model. Also, the initial conditions may contain errors (which in a worse case propagate), and the output is only available for discrete points in space (horizontal as well as vertical) and time. Finally, the initial conditions age with time - they are already old when the computation starts let alone when the output is published. Predictions of meteorological variables are issued several times per day (commonly between 2 and 4 times per day), and are available few hours after the beginning of the forecast period. This is because some time is needed for acquiring and analyzing the wealth of measurements used as input to NWP models, then run the model and check and distribute the output forecast series. This gap is a blind spot in the forecasts from an atmospheric model. As an example in the Netherlands, KNMI publishes 4 times per day expected values of wind speed, wind direction, temperature and pressure for the period the between 0 and 48 hours after initialization of the atmospheric model Hirlam with measured data, and then the period before forecast delivery is of 4 hours.

Many different atmospheric models are available, ranging from academic research tools to fully operational instruments. Besides for the very nature of the model (physical processes or numerical schemes) there are some clear distinctive differences between them: time domain (from several hours to 6 days ahead), area (several 10.000 km2 to an area covering half the planet), horizontal resolution (1 km to 100 km) and temporal resolution (1 hour to several hours).

One of the atmospheric models is the High Resolution Limited Area Model, abbreviated HiRLAM, which is frequently used in Europe. HiRLAM comes in many versions, that's why it is better to speak about "a" HiRLAM rather than "the" HiRLAM. Each version is maintained by a national institute such as the Dutch KNMI, the Danish DMI or Finnish FMI. And each institute has several versions under her wing, divided into categories such as: operational, preoperational, semi operational and for research purposes.

Other atmospheric models are UKMO in the UK, Lokalmodell in Germany, Aladin in France (Alladin and Lokalmodell are also used by some other country's within Europe), and MM5 in the USA.

2.2.2 Main Approaches

Research in the area of forecasting wind speed produced by wind farms has been devoted to the development of effective and reliable tools and many different approaches have been proposed. These tools can be classified whether the terrain information at the location is used as an input or not. Two mainstream approaches are the *physical and the statistical approach*. In some models a *combination approach* is used in an attempt to integrate the advantages of both approaches. In this section an overview of existing wind speed forecast approaches is presented [16].

2.2.2.1 Physical forecasting approach or Mechanistic model-driven

The physical approach to forecasting, in contrast to statistical approach, uses the detailed physical description to model the on-site conditions at the location of the wind farm [17, 18]. The basic operation of a physical approach is illustrated in Fig. 2.2.

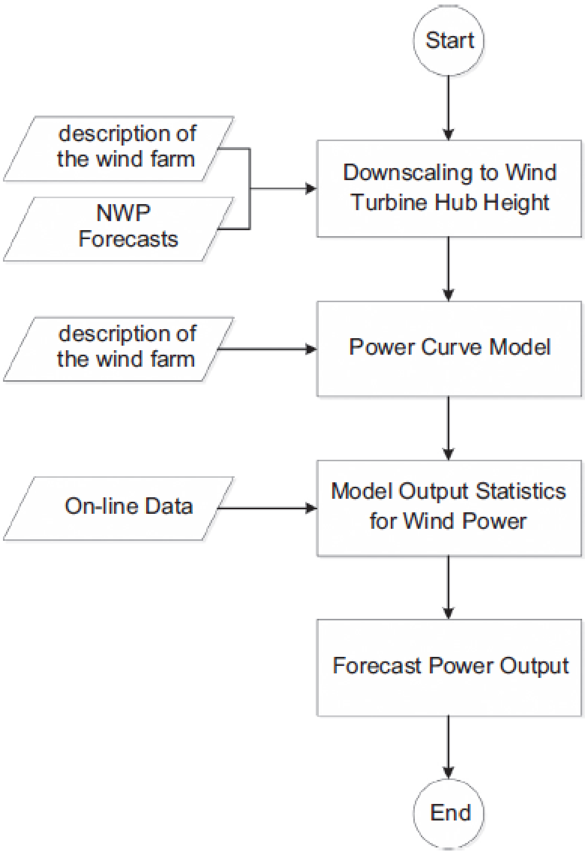

Start
description of the wind farm
NWP Forecasts
Downscaling to Wind Turbine Hub Height
description of the wind farm
Power Curve Model
On-line Data
Model Output Statistics for Wind Power
Forecast Power Output
End

FIGURE 2.2: The physical approach to forecasting wind speed and power (Extracted from [16]).

It carries out the refinement of the Numerical Weather Prediction (NWP) data to take into account the on-site conditions by the downscaling method, which are based on the physics of the lower atmospheric boundary layer. The downscaling method requires the detailed physical descriptions of the wind farms and their surroundings, including: description of the wind farm (wind farm layout and wind turbine power curve, etc.) and description of the terrain (orography, roughness, obstacles, etc.). Then, the refined wind speed data at the hub height of the wind turbines is plugged into the corresponding wind power curve to calculate the wind power production. If the on-line data is available, model output statistics are performed to reduce the error of the forecast. Contrary to the statistical approach, the physical approach does not require training input from historical data. However, acquiring the physical data is one of the main drawbacks of the approach [16].

A number of physical approaches have been introduced in [16]. The Prediktor is developed by the Risoe National Laboratory in Denmark. It uses Wind Atlas Analysis and Application Program (WAsP) and PARK program to take the local conditions into account by using the NWP forecast from High Resolution Limited Area Model (HIRLAM) [19]. The Previento, developed by the University of Oldenburg in Germany has a similar physical approach but uses a different NWP forecast from Lakelmodell of the German Weather Service [20]. The LocalPred is developed by CENER – National Renewable Energy Centre in Spain. It involves adaptive optimization of the NWP forecast, time series modeling, meso-scale modeling with MM5, and power curve modeling [21]. The eWind, developed by AWS TrueWind Inc. in the USA, has a similar physical approach with Prediktor but uses a high-resolution boundary layer model (ForeWind) as a numerical weather model to take the local conditions into account [22].

The physical approaches are based on the models using the fundamental physical principles for conservation of mass, momentum, and energy in air flows. These models address computational fluid dynamics (CFD) for simulating the atmosphere. Although there are many CFD models available, they are all based on the same basic physical principles. They differ in how the grids are structured and scaled, and how the numerical computations are performed [16].

In the majority of cases the statistical approaches provide good results in short-term, medium-term, and long-term forecasting. However, in the very short-term and short-term horizon, the influence of atmospheric dynamics becomes more important, and in these cases the use of physical approaches becomes essential.

2.2.2.2 Statistical forecasting approach or Data-driven

The alternative main approach for wind speed and power forecasting is based on statistical modeling. The statistical approach represents the relation between wind power or speed forecasting and explanatory variables including NWPs and on-line measured data [17]. The general form of the model is illustrated in Fig. 2.3.

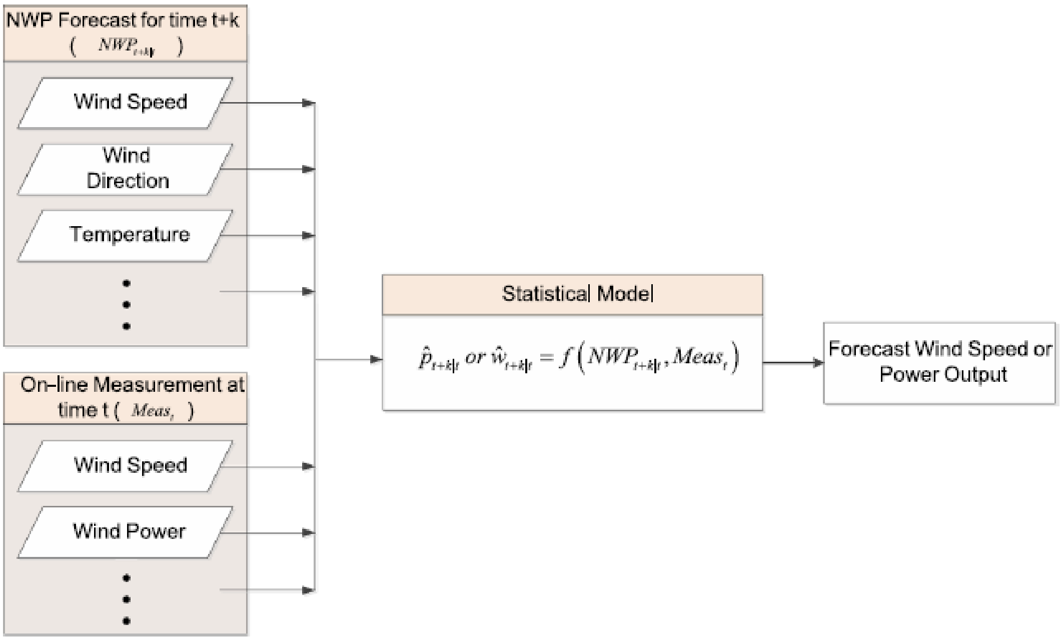

NWP Forecast for time t+k ($NWP_{t+k|t}$)
Wind Speed
Wind Direction
Temperature
On-line Measurement at time t ($Meas_t$)
Wind Speed
Wind Power
Statistical Model
$\hat{p}_{t+k|t}$ or $\hat{w}_{t+k|t} = f(NWP_{t+k|t}, Meas_t)$
Forecast Wind Speed or Power Output

FIGURE 2.3: The statistical approach to forecasting wind speed and power (Extracted from [16]).

The statistical approach generally uses previous history data to build the statistical model. This model uses NWP forecast for time t+k and on-line measurement at time t to forecast the present over the next few hours. It is easy to model and inexpensive. However, contrary to the physical approach, the statistical approach requires historical data to train the statistical model. Many different approaches are employed [16]. Some of the most representative statistical approaches are reviewed in this section.

2.2.2.2.1 Conventional statistical approach

In the conventional statistical approach a time series model is applied to forecast future wind power or speed. According to the forecasting process, which was proposed by Box–Jenkins, this model is divided into four main steps to make a mathematical model of the problem including model identification, model estimation, model diagnostics checking, and forecasting. Several types of time series models may be considered, including autoregressive model (AR), moving average model (MA), autoregressive moving average model (ARMA), and autoregressive integrated moving average model (ARIMA). The general form of the model is:

$$X_t = c + \varepsilon_t + \sum_{i=1}^{p} \varphi X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} \tag{2.1}$$

where X_t represents the forecasting parameter at time t, φ_i is the autoregressive parameter, θ_i is the moving average parameter, c is the constant, and random variable ε_t is the white noise. This model represents the ARMA model having the autoregressive model of order p and the moving average model of order q (ARMA(p,q)). If the order of the moving average model (q) is zero, it represents the autoregressive model of order p (AR(p)). If the order of the autoregressive model (p) is zero, it represents the moving average model of order q (MA(q)). The ARIMA model is a generalization of an ARMA model [16].

2.2.2.2.2 Artificial Neural Network approach

Another common approach is based on the use of Artificial Neural Networks (ANN). The NWP forecasts and further meteorological variables are transformed into the wind power or speed forecast by ANN which has been trained by the large sets of historical data in order to learn the dependence of the output on input variables. The general ANN approach for wind speed and power forecast is shown in Fig. 2.4.

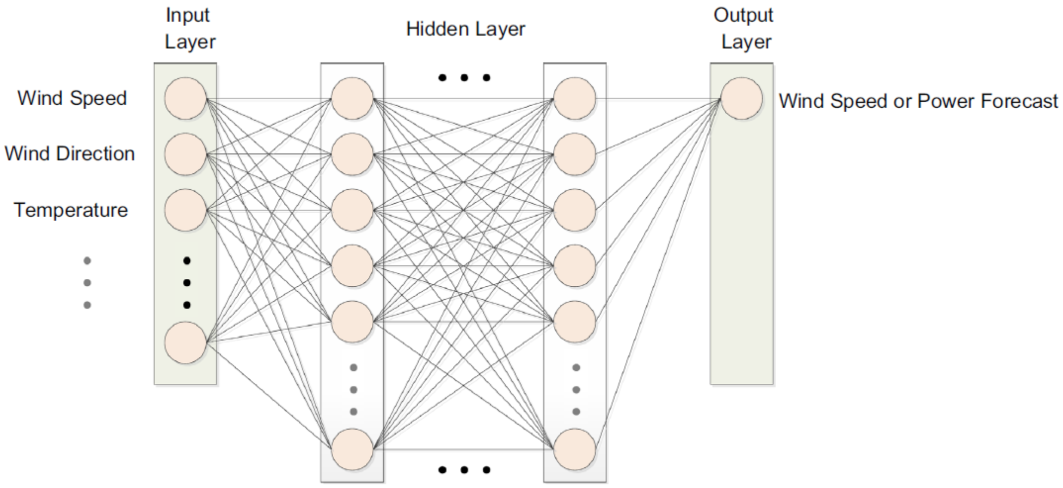


FIGURE 2.4: ANN approach for wind speed and power forecast (Extracted from [16]).

ANN is one of the widely used statistical approaches for wind speed and power forecasts. It consists of an input layer, one or more hidden layers, and an output layer. Each layer has a number of artificial neurons, and it uses a connectionist approach to connect the neurons to the neurons of the previous layer. This approach is able to model the complex non-linear relationship between the input and output layers through a training and learning process. This approach does not require explicit mathematical expressions as used in the physical and statistical approaches reviewed previously. Furthermore, it has the ability of self-learning, self-organizing and self-adaption [16].

2.2.2.3 Comparison

As indicated above, the underlying idea behind combining models can be described as the utilization of the features of different forecasting methods. In this context, it is reasonable to first briefly mention the most widely used forecasting methods in the literature and their characteristics. To this end, the forecasting methods are classified according to the common terminology criteria for wind forecasting methods and inspected by several studies from the literature [7].

The majority of the wind forecasting techniques can be clustered into two main groups, namely *physical methods* and *statistical methods*. In short, the first group takes into account the physical considerations such as local terrain, wind farm layout and temperature to reach the estimate and utilizes the output from Numerical Weather

Prediction (NWP) models which provide weather forecasts by using the mathematical model of the atmosphere. The concept of utilizing the NWP models as an input was taken into account by Landberg and then corrections on the wind speed predictions were applied by making use of various programs such as Wind Atlas Analysis and Application Program (WAsP) and PARK [23]. Furthermore, the NWP model output can be used directly for wind speed predictions, as demonstrated in [24]. Likewise, another NWP model, called as Eta Model, was utilized for wind prediction up to 36 h in [15] and it was shown that Eta model is quite effective in predicting wind energy.

The latter aims at describing the relation between historical time series of wind speed (or power) at the location of interest by generally recursive techniques and it can be stated that short term forecasting models are generally based on statistical approaches due to the fact that NWP models require long operation time and large amount of computational resources.

As can be seen from the mentioned research and review studies, each forecasting model has its own strengths and weaknesses over the other models. In order to gather the present knowledge on the widely used wind forecasting models in the literature, a concise comparison of these basic approaches is shown in Table 2.1.

2.2.3 Combination Approaches

The basic idea of the *combination model* is to combine different strategies or models, retaining advantages of each approach. The desire is to improve the forecast accuracy, but combining forecasts does not always perform better than the best individual forecasts. However, in some cases it is viewed as less risky to combine forecasts than to select just an individual forecast [25].

Several approaches have been developed based on the combination of various models [16]. As stated earlier, combined forecasting methodologies can improve the final forecasting performance taking advantages of individual forecasting methods which have different performances depending on the data sets, forecast horizons as well as their capability of capturing nonlinearity, and provide some advantages compared to the individual methods. Among the advantages, the potential of utilizing the combined methods in a wider range of application area has a special importance due to the fact that individual models perform well only in a certain situation and therefore different models have to be tested for deciding the most suitable one. The mentioned time consuming drawback of individual methods can be overcome while using combined models, particularly in the case that the determination of the best performing model is complicated.

There is controversy and confusion in the literature about the definition and structure of the combined models. Notwithstanding, it can be indicated that the most widely accepted procedure for combination of models taking place in the literature is to assign a weighting coefficient to each method proportional to their past forecasting performance. Besides, some other approaches are presented as combined models in the literature of wind forecasting by utilizing different methodologies. The mentioned weighting and other combination methods are elucidated and exemplified in detail according to a chronological sequence and by grouping the similar studies, correspondingly in the following sections [7].

TABLE 2.1: Brief comparison of the main methods used for forecasting of wind speed in the literature (Extracted from [7].)

Wind speed forecasting approach	**Advantages**	**Disadvantages**
NWP models	Applicable for longer prediction horizons	Weakness in handling smaller scale phenomena, not suitable for short forecast times, requires large computational resource and time
Time series models (AR, ARMA, ARIMA, f-ARIMA, etc.)	Easy to find tools, comparatively basic structure, capability of correcting local trends in data, provides confidence intervals for predictions	Requires a great deal of historic records, difficult to model nonlinear problems and decide the best structure
ANN-based models	Gains knowledge from training data, no need to specify any mathematical model a priori, high data error tolerance, higher adaptability to online measurements	Requires a training procedure and a large number of training data
SVM-based models	High generalization performance	Depends on the tuning of parameters appropriately, complex optimization process and longer training time
Fuzzy logic models	Suitable for systems which are difficult to model exactly, relatively less complex	High complexity and a long process time in the case of many rules
Bayesian networks	Ability to handle missing observations and to avoid the overfitting of data, suitable for small training data sets, suitable for various input data	Requires relatively more effort, depends on the user's expertise level
Kalman filter models	Does not require to store all historic data because of its recursive form	Requires previous knowledge about the system

2.2.3.1 Weighting-based combined approaches

From a conceptual point of view, the process of weighting used for the combination of the wind forecasting models can be defined as determining the relative effectiveness of each model and providing them an appropriate value that reflects their special importance in the combined model. The general prediction process of these combined models is depicted in Fig. 2.5.

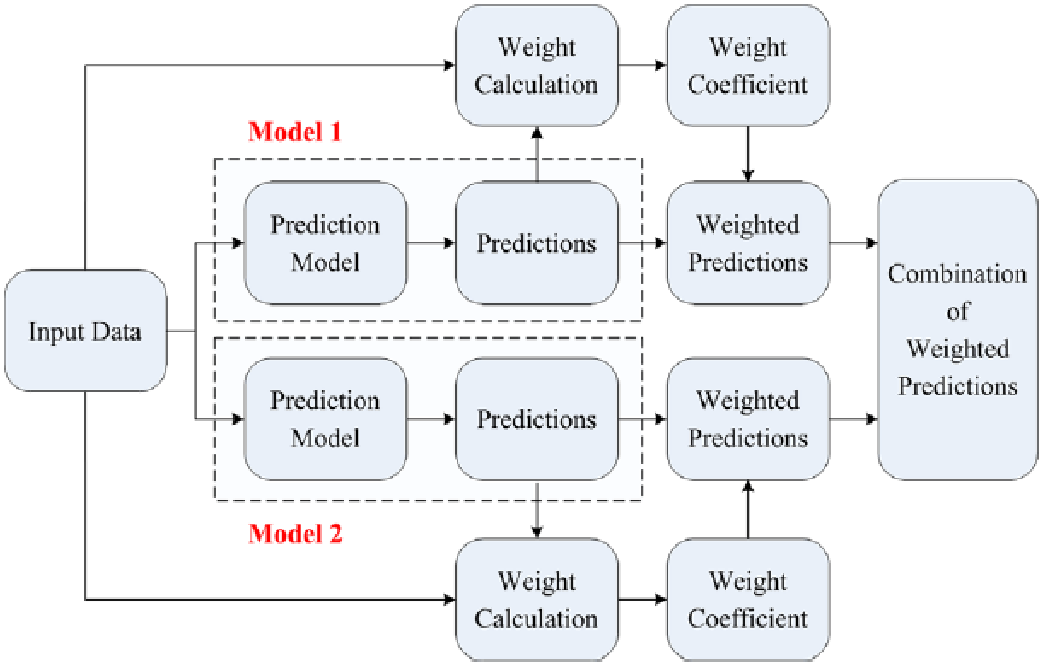


FIGURE 2.5: Flowchart of the weighting-based combined approaches. (Extracted from [7])

2.2.3.2 Other combined approaches

In the literature, these approaches generally consist of two different models, one for main forecasting task and the other for the auxiliary processes such as data filtering, data decomposition, selection of the best parameter and residual error evaluation. Depending on the function of the models used for the auxiliary processes in the final model, "the other combined approaches" can be gathered into three groups:

- combined approaches including data preprocessing techniques,
- combined approaches including parameter selection and optimization techniques, and
- combined approaches including data post-processing techniques [7].

2.2.3.2.1 Combined approaches including data preprocessing techniques

In the first group, the main objective of the data pre-processing models is, as the name implies, to realize a preliminary process on data sets by decomposing the nonlinear wind speed time series into more stationary and regular subseries which are generally easier to analyze and/or by filtering out the irrelevant and redundant features of the data set. Thus, more stable subseries are obtained as well as the most informative training data is determined enabling to improve the quality of the data and avoiding the unnecessary computation burden. The basic flowchart followed in a large part of the mentioned approaches is illustrated in Fig. 2.6.

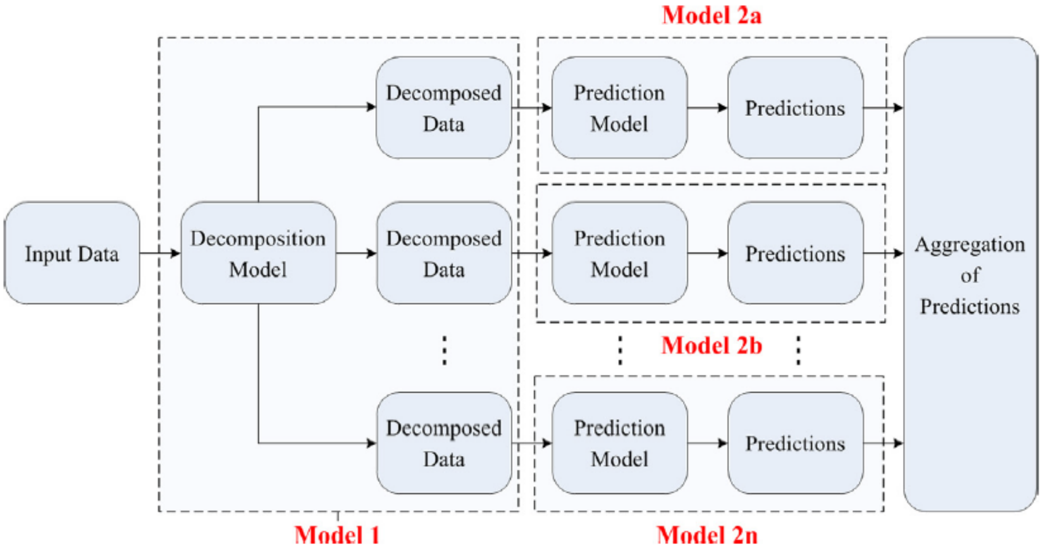


FIGURE 2.6: Flowchart of combined approaches including data preprocessing techniques (Extracted from [7]).

2.2.3.2.2 Combined approaches including parameter selection and optimization techniques

In a large numbers of papers on wind forecasting, it has been reported that certain parameter selection and optimization approaches also can make a considerable contribution to the prediction performance during the training process. The selection of explanatory variables and determination of model parameters while using the mentioned approaches can allow avoiding the time consuming process of the optimization of the prediction method, which is usually carried out by testing the method over a large number of candidate parameters and deriving the structure of the model heuristically. The approaches in question perform the predictions after a parameter evaluation process with a suitable method, as shown in Fig. 2.7.

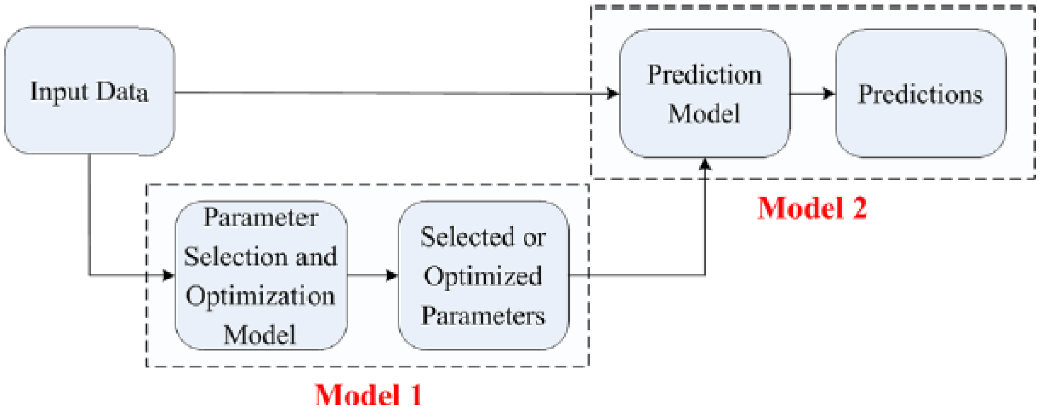

Input Data
Prediction Model
Predictions
Model 2
Parameter Selection and Optimization Model
Selected or Optimized Parameters
Model 1

FIGURE 2.7: Flowchart of combined approaches including parameter selection and optimization techniques. (Extracted from [7])

2.2.3.2.3 Combined approaches including error processing techniques

Apart from the approaches mentioned before, there exists a few combined approaches in the literature that have a structure that takes into account the residual error values obtained from a forecasting model, as denoted in Fig. 2.8.

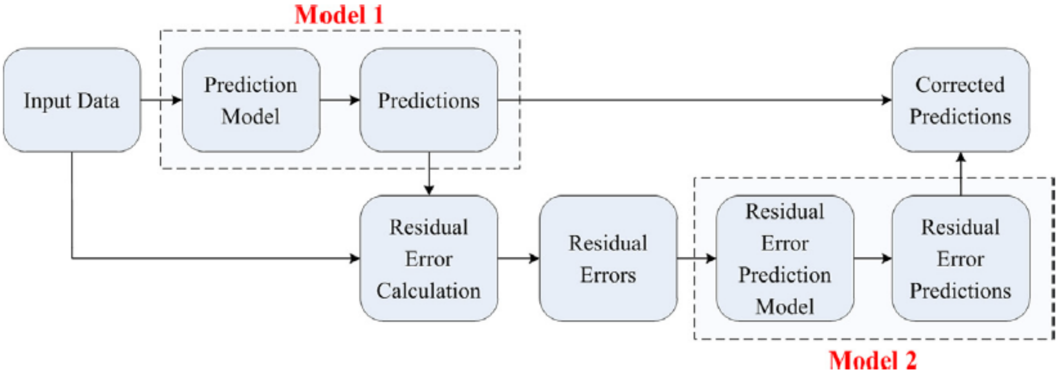


FIGURE 2.8: Flowchart of combined approaches including error processing techniques (Extracted from [7]).

2.2.3.3 Comparison

In the literature, there exist a lot of studies on the predictions of wind speed and power with various methods, which mostly generate reasonable values and each method has its own advantages and disadvantages. However, due to the fact that the prediction models developed are generally site-specific and considerably influenced by the changing of required prediction times, the suitable model is selected regarding the specific data characteristic of the site and the application area of the method with a time-consuming and specialist process. In order to address this problem by avoiding the ambiguity of model selection and to further improve the prediction performance, combining several methods have been proposed, especially at the last years and the results obtained have verified the effectiveness of these approaches. The subject of combined wind speed and power forecasting methods can be deemed as a novel research area. Therefore, a consensus has not yet been reached concerning the fundamentals of the subject and the basic definitions about the related terms. For instance, it can be readily indicated that predictions errors are always proportional to the prediction time; however, it is not possible to say exactly which method is the most appropriate candidate for a certain prediction time. In this issue, only some suggestions and assumptions are available at this stage. Likewise, the classification of the methods according to the prediction time is another subject that is suggested in various ways in the literature. For instance, the short term prediction is defined as the predictions up to six-hour-ahead in [26]; however, it corresponds to 24-h-ahead predictions in [27] and 72-h-ahead predictions in [9]. Besides, comparison of the forecasting performance of the models is another troublesome topic evidently due to the fact that no single model provides the best predictions in terms of all performance metrics, meaning that a universal standard for a fair comparison of prediction performances is not yet present [7].

Similar to the above mentioned problems, it is important to highlight that there still exist discrepancies in the definitions and structures of the combined and hybrid forecasting approaches and this situation complicates the classification and comparison of the approaches. For this purpose, the approaches were divided into several main and sub-categories and the effects of different models on the forecasting performance are briefly described and discussed by pursuing an extensive body of the literature in this direction [7]. According to the mentioned studies, firstly there is no doubt that combination of proper prediction techniques is of significant importance in terms of improving the accuracy. Then, the preliminary processing of the input data, particularly with decomposing and filtering methods such as a Wavelet Transform (WT) and EMD (Empirical Mode Decomposition) methods, has been found valuable for facilitating the forecasting process and thereby, improving the prediction quality with the cost of spending more time on final model building. Likewise, it was observed that evolutionary algorithms such as GA (Genetic Algorithms), EP (Evolutionary Programming) and DE (Differential Evolution) methods and Kalman filtering method tend to perform well for the selection and optimization of the prediction system parameters. Moreover, it was realized that the total performance of the prediction methods can be augmented by also predicting and then including the residual error values with a variety of models. A brief evaluation about each class of the combined approaches which is determined according to the above mentioned criterion is shown in Table 2.2 [7].

By examining the characteristics of the combined approaches summarized in Table 2, it can be concluded that the approaches with a weighting model and with a data preprocessing model are very suitable for longer and harder prediction tasks, which are required for power system operations such as scheduling, unit commitment and load following. Weighting-based approaches consisting of an NWP model should also be utilized effectively for the much longer term objectives such as maintenance of wind turbines or conventional power plants. Besides, the approaches including parameter selection and data optimization techniques may present high-accuracy predictions required mainly for energy trading and marketing. However, the approaches with an error processing method may only give reasonable results in the case of systematic errors and hence are not of a specific application area. Lastly, it is worth mentioning that the combined models are ineffective in very short term forecasts, ranging from milliseconds up to a few minutes and used for wind turbine active control, due to the fact that these approaches have generally computational time inefficiency compared to the individual methods [7].

TABLE 2.2: Brief evaluation of combined approaches applied for forecasting of wind speed in the literature (Extracted from [7]).

Combined wind speed forecasting approach	**Strategy**	**Advantages**	**Disadvantages**
Weighting-based combined approaches	Assigning weight factors to models according to their performance	Easy to implement and code, suitable for a wide range of prediction time, adaptive to new data	Does not guarantee the best predictions along the prediction horizon, requires an extra model for determining the weights
Combined approaches including data preprocessing techniques	Forecasting of the subseries obtained by decomposition models	Higher performance compared to other approaches, easy to find literature examples, robustness to rapid changes in wind speed	Requires a detailed mathematical knowledge on decomposition models, provides slow response to new data
Combined approaches including parameter selection and optimization techniques	Optimization of the parameters of forecasting model	Easy to find literature examples, a relatively basic structure	Harder to code, dependent on designer's knowledge about the optimization problems, computationally intensive
Combined approaches including error processing techniques	Forecasting of residual error caused by forecasting model	High accuracy, effective in reducing systematic error	Computational time inefficiency

2.3 Atmospheric Reanalysis Data

Reanalysis is a systematic approach to produce data sets for climate monitoring and research. Reanalyses are created via an unchanging ("frozen") data assimilation scheme and model(s) which ingest all available observations every 6-12 hours over the period being analyzed. This unchanging framework provides a dynamically consistent estimate of the climate state at each time step. The one component of this framework which does vary are the sources of the raw input data. This is unavoidable due to the ever changing observational network which includes, but is not limited to, radiosonde, satellite, buoy, aircraft and ship reports. Currently, approximately 7-9 million observations are ingested at each time step. Over the duration of each reanalysis product, the changing observation mix can produce artificial variability and spurious trends. Still, the various reanalysis products have proven to be quite useful when used with appropriate care [28].

Key Strengths

- Global data sets, consistent spatial and temporal resolution over 3 or more decades, hundreds of variables available; model resolution and biases have steadily improved
- Reanalyses incorporate millions of observations into a stable data assimilation system that would be nearly impossible for an individual to collect and analyze separately, enabling a number of climate processes to be studied
- Reanalysis data sets are relatively straightforward to handle from a processing standpoint (although file sizes can be very large)

Key Limitations

- Observational constraints, and therefore reanalysis reliability, can considerably vary depending on the location, time period, and variable considered
- The changing mix of observations, and biases in observations and models, can introduce spurious variability and trends into reanalysis output
- Diagnostic variables relating to the hydrological cycle, such as precipitation and evaporation, should be used with extreme caution

2.3.1 European Centre for Medium-Range Weather Forecasts

The European Centre for Medium-Range Weather Forecasts (ECMWF) is an independent intergovernmental organisation supported by 34 states.

ECMWF is both a research institute and a 24/7 operational service, producing and disseminating numerical weather predictions to its Member States. This data is fully available to the national meteorological services in the Member States. The Centre also offers a catalogue of forecast data that can be purchased by businesses worldwide and other commercial customers. The supercomputer facility (and associated data archive) at ECMWF is one of the largest of its type in Europe and Member States can use 25% of its capacity for their own purposes.

The organisation was established in 1975 and now employs around 350 staff from more than 30 countries. ECMWF is one of the six members of the Co-ordinated Organisations, which also include the North Atlantic Treaty Organisation (NATO), the Council of Europe (CoE), the European Space Agency (ESA), the Organisation for Economic Co-operation and Development (OECD), and the European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT).

ECMWF is based in Reading, UK.

What they produce:

- produce numerical weather forecasts and monitor the Earth-system;
- carry out scientific and technical research to improve forecast skill;
- maintain an archive of meteorological data.

To do so the Centre provides:

- twice-daily global numerical weather forecasts;
- air quality analysis;
- atmospheric composition monitoring;
- climate monitoring;
- ocean circulation analysis;
- hydrological prediction.

2.3.1.1 ERA-Interim

ERA-Interim is a global atmospheric reanalysis data from 1979, continuously updated in real time.

The data assimilation system used to produce ERA-Interim is based on a 2006 release of the IFS (Cy31r2). The system includes a 4-dimensional variational analysis (4D-Var) with a 12-hour analysis window. The spatial resolution of the data set is approximately 80 km (T255 spectral) on 60 vertical levels from the surface up to 0.1 hPa.

ERA-Interim data can be downloaded from the ECMWF Public Datasets web interface or from MARS (class=ei, expver=1).

For a detailed documentation of the ERA-Interim Archive see [29] .

2.4 Machine Learning techniques used

In this section we proceed to explain the algorithms that are going to be used on our proposed methodology.

Firstly, we will detail the unsupervised techniques which are the ones used for feature weighting called Unsupervised Entropy-base method and also the one used to create the clusters which will be used to train and test our local models.

Later, we will detail the supervised techniques which we separate in two groups. The first group refers to the simple models which are the k-Nearest Neighbor and the Artificial Neural Network and the second group refers to the combined models which are the Random Forest and the Ensemble model. Finally, we present the algorithms used for the combined approaches that we proposed: the local models and the ensemble approach.

2.4.1 Unsupervised methods

2.4.1.1 Feature selection/weighting methods

2.4.1.1.1 Instance-Based Learning Feature Weighting

The results in [30] show in an empirical way that you can use unsupervised weighting algorithms to determine the feature relevance in unsupervised databases, as a first approach. They think that this is due to the fact that similarity computations between instances in unsupervised methods capture the intrinsic distribution of different instances (different "classes") in a similar way than supervised methods do.

Unsupervised Entropy-Based methods

In [31], the authors present a feature selection method for unsupervised domains based in entropy computations. Starting from this approach (UEB), in [30] they made an extension to obtain two feature weighting algorithms (UEB-1 and UEB-2), trying to obtain a superior performance assigning real-valued weights instead of binary-value weights. The underlying idea is that data have orderly configurations if they have distinct clusters, and have disorderly or chaotic configurations otherwise [31]. From entropy theory, it can be stated that entropy is lower for ordered configurations, and higher for disordered configurations. The feature selection method is based on the observation that removing an irrelevant feature from the feature set may not change the underlying concept of the data, but not so otherwise. Following this idea, the first step consist in compute the entropy between two instances:

$$E = -Slog_2S - (1-S)log_2(1-S) \tag{2.2}$$

where S is the similarity measure based on a distance concept, and assumes a very small value (close to 0.0) for very close pairs of instances, and a very large (close to 1.0) for very distant pairs. For the entire data set of N instances the entropy measure is given as:

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ij} * log_2S_{ij} + (1-S_{ij}) * log_2(1-S_{ij}) \tag{2.3}$$

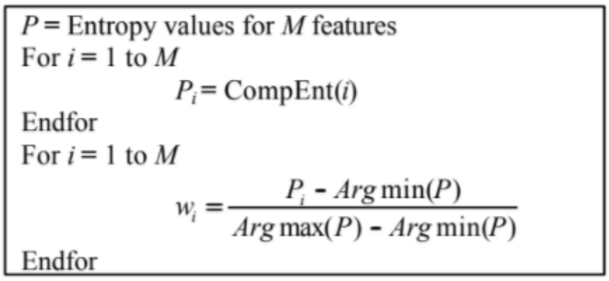
where Sij is the similarity value between the instance i and the instance j normalized to [0,1]. When all features are numeric or ordinal, the similarity of two instances is: $S_{ij} = e^{-a*D_{ij}}$ where D_{ij} is the distance between the instances i and j. The value of α is computed automatically by: $a = -ln0.5/\bar{D}$, where $\bar{D}$ is the average distance among all the instances. Euclidean distance is used to compute the distance Dij. If all the attributes are nominal, the similarity between two instances is: $S_{ij} = \sum_{k=1}^{M} |x_{ik} = x_{jk}|/M$ where $|x_{ik} = x_{jk}|$ is 1 if x_{ik} equals x_{jk} and 0 otherwise, and M is the number of features.

-Unsupervised Entropy-Based method 1 (UEB-1)

The algorithm computes the entropy of data by removing a feature. For M features this is repeated M times. Features are ranked in descending order of relevance by finding the descending order of the entropy after removing each of the M features one at a time. Feature selection algorithms focuses on deciding if one attribute is relevant or not. On the other hand, feature weighting algorithms focus on giving a relevance measure for each attribute. In our method (UEB-1) that is the first extension of the UEB algorithm, to obtain feature weights instead of feature selection, the approach takes the entropy values computed for each one of the attributes, and applies a scaling process to assign weights. To obtain weights in [0,1] range for each attribute k, the following computation is done:

$$W_k = \frac{Entropy_k - Argmin(Entropy)}{Argmax(Entropy) - Argmin(Entropy)} \tag{2.4}$$

In 2.9, an outline of the algorithm is described. $CompEnt(i)$ computes the entropy of the data after discarding the ith feature.

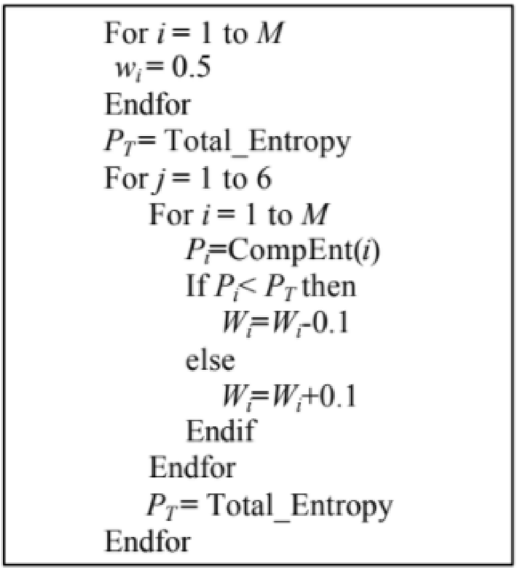


```
$P$ = Entropy values for $M$ features
For $i = 1$ to $M$
        $P_i$ = CompEnt($i$)
Endfor
For $i = 1$ to $M$
        $w_i = \frac{P_i - Arg\min(P)}{Arg\max(P) - Arg\min(P)}$
Endfor
```

FIGURE 2.9: UEB-1 Algorithm outline (Extracted from [30]).

-Unsupervised Entropy-Based method 2 (UEB-2)

The second extension (UEB-2) performs a wrapper approach in the sense that implements an update of weights in each step of the cycle taking into account the last values of computed weights. Taking this into account, UEB-1 can be seen as a filter approach. In UEB-2, an initial weight of 0.5 is assigned to each attribute and the entropy for the entire database is computed. Then, it computes the new entropy value after removing one attribute at a time. If the new entropy value after removing one attribute is less than the entropy of the entire database, then the weight of that attribute is decreased by 0.1. If the new entropy value after removing one attribute is greater than the entropy of the entire database, the weight of that attribute is increased by 0.1. This increasing/decreasing parameter was set to 0.1 after an empirical study. This cycle is performed several times allowing the weights to reach a minimum or maximum value in the [0,1] range. After an empirical evaluation, this parameter was set to 6. An outline of the UBE-2 algorithm is presented in 2.10. Total Entropy is the entropy for all the database taking into account all the features. $CompEnt(i)$ computes the entropy of the data after discarding the ith feature.



```
For $i$ = 1 to $M$
 $w_i$ = 0.5
Endfor
$P_T$= Total_Entropy
For $j$ = 1 to 6
    For $i$ = 1 to $M$
        $P_i$=CompEnt($i$)
        If $P_i$< $P_T$ then
            $W_i$=$W_i$-0.1
        else
            $W_i$=$W_i$+0.1
        Endif
    Endfor
    $P_T$= Total_Entropy
Endfor
```

FIGURE 2.10: UEB-2 Algorithm outline (Extracted from [30]).

2.4.1.2 Clustering methods

A partition clustering technique generates a single partition of the data in an attempt to recover natural groups present in the data. It tries to obtain a good partition of the observations. The partition is composed by a set of groups or clusters. Thus, this kind of techniques assign each observation to the "best" cluster. This "best" cluster is the one optimizing certain criterion (minimization of the square sum of distances of the observations to the centroids of the clusters, etc.). Either these algorithms require the number of clusters to be obtained, namely k, or some threshold value (classification distance) used to decide whether an observation belongs to a forming cluster or not.

Partitional clustering methods are especially appropriate for the efficient representation and compression of large databases, and when just one partition is needed.

2.4.1.2.1 K-means Clustering

This clustering method was utilized for power prediction by Kusiak and Li with various data-mining algorithms [15].

One of the most popular partitional clustering algorithms is the K-means clustering algorithm described for MacQueen in 1967. Starting with a randomly initial partition, it explodes the idea of changing the current partition to another one decreasing the sum of squares of distances of the observations to the centroids of the clusters. It converges, possibly to a local minimum, but in general can converge fast in a few iterations. It has a main parameter k, which is the number of desired clusters.

The K-Means Algorithm

The main goal of clustering is to generate compact groups of objects or data that share similar patterns within the same cluster, and isolate these groups from those which contain elements with different characteristics. In the field of renewable energy forecasting, this technique allows handling groups of data separately, which provides a better understanding of the collected information and improves the accuracy of the final forecast results. The K - means is a well-known, low complexity algorithm utilized for data-partitioning in this scenario. The algorithm starts running after an input of K clusters is given, and outputs the cluster centroids through iterations. Let $X = x_1, x_2, ..., x_N$ be the set of N points to be grouped into K different clusters, set as $C = C_k k = 1, 2, ..., K$. By means of the Euclidean distance, the algorithm assigns each data point to its closest centroid C_k,, calculated by:

$$C_k = (\frac{1}{N_k}).\sum_{i=1}^{N_k} X_i^k \tag{2.5}$$

where X_i^k is the i-th data point in the cluster k, and Nk is the number of data points in the respective cluster.

After the first run, the algorithm calculates the mean of the data points in each cluster Ck and selects this value as a new cluster centroid, starting a new iteration. As new clusters are selected, a new mean value is also obtained. The algorithm halts once the sum of the squared error over K clusters is minimized. However, with respect to its computational complexity and the initialization step, the K-means approach presents some limitations [32].

The number of clusters, k, must be known before the first iteration of the algorithm.

The K-means algorithm is very sensitive to the initial cluster centroids. The more the selected initial clusters are distant from the optimal cluster centroid, the more iteration will take for the algorithm to converge.

K-means is also strongly sensitive to noisy data, which may affect the accuracy of the final forecasts [33].

2.4.2 Supervised methods

2.4.2.1 Single predictive methods

2.4.2.1.1 k Nearest Neighbors

As a first single model we propose the k-nearest neighbors algorithm [34]. This choice is made for the following reasons:

- Interpretability of the model. The results of the predictive algorithm using the k-nearest neighbors approach are based upon the occasions in the past that are closest to the current state (according to a given distance metric). Prediction is fulfilled by a simple averaging of the output values of the k nearest neighbors, or by some weighted averaging. The k-nearest neighbors algorithm allows its results to be interpreted by experts [35].

- Cyclic factors treatment. The factors used include some cyclic ones (year, month etc.). The k-nearest neighbors algorithm can be tuned to work with them (unlike tree methods, for example, which are not able to deal with cyclic factors).

- No multiple learning is needed with the k-nearest neighbors algorithm when new portions of data are introduced. In this case, when adding samples, we expand the search instances without a need to recalculate the model. Removing old data is also done without repeated learning [35].

Nearest Neighbors Algorithm

In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method used for classification and numerical prediction. [36] In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or numerical prediction:

In k-NN numerical prediction, the predictable variable for the object. This value is the average of the values of its k nearest neighbors. k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The k-NN algorithm is among the simplest of all machine learning algorithms but is also very powerful.

Both for classification and numerical prediction, a useful technique can be to assign weight to the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. For example, a common weighting scheme consists in giving each neighbor a weight of 1/d, where d is the distance to the neighbor.

The neighbors are taken from a set of objects for which the class (for k-NN classification) or the predictable variable (for k-NN numerical prediction) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required.

A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data [37].

The main idea of k-NN algorithm is that whenever there is a new point to predict, its k nearest neighbors are chosen from the training data. Then, the prediction of the new point can be the average of the values of its k nearest neighbors. Wind prediction using the k-NN algorithm has been developed successfully in recent years [35], and resembles the similar-day approach for electrical load forecasting [38]. The similar-day approach is still used by many utilities, and derives the future power load using historical days with similar temperatures and day types. In this work, the k-NN algorithm is used to find the appropriate historical examples with characteristics similar to the future weather condition (provided by the NWP model). Then, wind speed or power observations of these historical examples will be extracted and used to construct the wind speed or power predictive density. The main process of probabilistic wind speed or power forecasting using the k-NN algorithm can be boiled down to three steps:

- Calculating the predefined distance between the testing example and the training example;
- Choosing k nearest neighbors from the training examples with the k smallest distances;
- Predicting the wind speed or power output based on a averaging technique

Firstly, a distance measure is required to characterize the similarity of any two instances (each record in the dataset is called an instance). The most commonly used distance metrics are the Euclidean distance, the Manhattan distance and the Mahalanobis distance. In this work, the original Euclidean distance was used.

The Euclidean distance is calculated as:

$$D_{X,Y} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.6}$$

where X and Y are two instances from the training and testing datasets, respectively; x_i and y_i are input variables; and n is the number of input variables. w_i is the weight assigned to the ith input variable. In the original Euclidean distance, the weight w_i is equal to 1, which means that each input variable is assumed to make the same contribution to the distance D. Secondly, the instances with the k smallest distances are chosen as the k nearest neighbors. Let $X_1, X_2, ..., X_K$ denote the k nearest neighbors, and their corresponding wind speed or power observations be represented by $p_1, p_2, ..., p_K$. The distance d_k from the training instance X_k to the testing instance Y is calculated using the euclidean distance, and follows an ascending order $d_1 \leq ... \leq d_k \leq ... \leq d_K$, where $d_k = D[X_k, Y]$. Once k nearest neighbors have been determined, wind power point prediction is derived using the average.

2.4.2.1.2 Artificial Neural Networks

Artificial neural network is a predictive model and classification technique employed to model complex relationships between cause and effect variable sets or to identify patterns within data. They have broad applications in a variety of fields including transient detection, pattern recognition, approximation, and time-series prediction [39, 40]. The network topology includes an input layer, one or several hidden layers and an output layer. The neurons, which form part of the individual layers, are connected by unknown parameters. These parameters can be adjusted using a variety of algorithms, e.g., the Levenberg–Marquardt algorithm [40, 41].

ANN Algorithm

ANNs are widely accepted as a technology offering an alternative way to tackle complex and ill-defined problems. An ANN is an information processing pattern which works in a way that a human brain processes information. The structure of this information processing system is composed of highly interconnected processing elements, called neurons working in parallel to solve problems. A neural network helps when it is highly complex to formulate an algorithmic solution and also where there is a need to pick out the structure from the existing data. Neural networks learn by example and they cannot be programmed to perform a specific task. They are fault-tolerant, that is, they are able to handle noisy and incomplete data, are able to deal with nonlinear problems and once trained can assist in prediction and generalization at high speed. In more practical terms, neural network is a non-linear statistical data modeling tool. The tasks for which ANNs are useful fall into various applications such as control, pattern recognition, forecasting, optimization, etc. In this study, ANN is applied for the prediction of wind generation from a specific wind farm [42].

An (artificial) neural network is a network of simple elements called neurons, which receive input, change their internal state (activation) according to that input, and produce output depending on the input and activation. The network is built by connecting the output of certain neurons to the input of other neurons forming a directed, weighted graph. The weights as well as the functions that compute the activation can be modified by a process called learning which is governed by a learning rule [43].

Components of an artificial neural network

-Neurons

A neuron with label j receiving an input pj(t) from predecessor neurons consists of the following components [43]:

- an activation $a_j(t)$, depending on a discrete time parameter,
- possibly a threshold $\theta_j(t)$, which stays fixed unless changed by a learning function,

- an activation function f that computes the new activation at a given time $t+1$ from $a_j(t), \theta_j$ and the input $p_j(t)$ giving rise to the relation $a_j(t+1) = f(a_j(t), p + j(t), \theta_j)$ and an output function f_{out} computing the output from the activation $o_j(t) = f_{out}(a_j(t))$. Often the output function is simple the Identity function.

An input neuron has no predecessor but serves as input interface for the whole network. Similarly an output neuron has no successor and thus serves as output interface of the whole network [44].

-Connections and weights

The network consists of connections, each connection transferring the output of a neuron i to the input of a neuron j. In this sense i is the predecessor of j and j is the successor of i. Each connection is assigned a weight w_{ij}. [43]

-Propagation function

The propagation function computes the input $p_j(t)$ to the neuron j from the outputs $o_i(t)$ of the predecessor neurons and typically has the form [43]:

$$p_j(t) = \sum_i o_i(t) w_{ij} \tag{2.7}$$

-Learning rule

The learning rule is a rule or an algorithm which modifies the parameters of the artificial neural network, in order for a given input to the network to produce a favored output. This learning process typically amounts to modifying the weights and thresholds of the variables within the network [43] [44].

2.4.2.2 Combined predictive methods

2.4.2.2.1 RandomForest

The algorithm of Random forests has many advantages such as less adjustable parameters, higher precision of prediction and better generalization ability [45].

RandomForest Algorithm

Random Forest (RF) was proposed by Leo Breiman [4] in 2001. The Random Forest algorithm is based on statistical learning theory, by using bootstrap as sampling method extracting multiple samples from the original sample, building decision tree modeling according to each bootstrap sample, then integrating the prediction of multiple decision trees, and coming up with the final results by voting ultimately. The essence of RF is a classifier containing a number of decision trees, which are formed by adopting random method. Random Forest Regression (RFR) can be regarded as a strong predictor integrating a lot of weak predictors (decision trees). The realization process of random forest is as follows:

- The original training set is N, the application of bootstrap method has been put back to the random extraction of K as a new self-help samples, and the resulting the classification trees, each time has not been drawn out of the sample composed of the out-of-bag data;

- From all available variables chose the best of the variables according to the decision tree growth criteria (entropy gain, gain ratio or impurity)

- Each tree growth the maximum and do not apply any pruning technique;

- The random forest is composed of trees, and the new data are identified and classified according to the random forest classifier. Supposing the training set is extracted independently from the distribution of random vector X, Y; as a result, any numerical prediction value H (X) of the mean square generalization error is [46]:

$$E_{x,y} = [Y - h(X)]^2 \tag{2.8}$$

The predicted value of the random forest numerical prediction is the average of the K decision trees:

$$h(\theta, X_k) \tag{2.9}$$

It is similar to random forest classification. The theorem can be seen here [46]: Theorem 1: $k \to \infty$

$$E_{X,Y} = [Y - avh_k(X, \theta_k)]^2 \to E_{X,Y}[Y - E_\theta(X, \theta_k)]^2 \tag{2.10}$$

Record type is the PE^{**} on the right side. It means the generalization error of random forest. The average generalization error of each decision tree PE^{**} can be defined as:

$$PE^{**} = E_\theta E_{X,Y}[Y - h(X, \theta)]^2 \tag{2.11}$$

Theorem 2: For all of the $\theta, EY = E_x h(X, \theta)$

$$PE^{**} \leq \bar{p}PE^{*} \tag{2.12}$$

In this type, $\bar{p}$ is the residual error $Y - h(X, \theta$ and $Y - h(X, \theta')$'s weighted correlation coefficient and also θ and $\theta'$are mutually independent.

The theorem 2 gives the exact regression forest condition: low correlation between residual errors and decision tree with low error. Random forest reduces the average error of decision tree by weighted correlation coefficient $\bar{p}$.

Random forest prediction can be viewed as an adaptive neighborhood classification and numerical prediction process. For each one X = x , both of them can get the original n observed value of the weight set. The estimation of random forest prediction or conditional mean is equivalent to the weighted mean of dependent variables [45].

In accordance with the random forest algorithm mentioned, the algorithm has fewer parameters to adjust and need not worrying about the features such as over fitting, speedy classification and high-efficient processing large sample data, estimable characteristic factor importance, strong ability to resist noise and so on. Hence, random forest can fully reflect the advantages of data mining and does not need to assume the implementation of function form in order to avoid the hypothesis error. In wind power prediction, using random forest numerical prediction method can effectively analyze the nonlinear and interaction data. It does not need to assume the provided model of mathematical form in advance. It has good numerical prediction analysis results.

The study of wind farm power prediction by using random forest method is based on the learning rule of high precision fitting in samples and the ability of the high confidence to promote knowledge out of samples. Random forest has two sorts of techniques, namely, classification and numerical prediction.

The following part shows the calculation procedure of the wind farm power prediction model [47]based on random forest: Wind power prediction of random forests is a collection of B trees $T_1(X), ..., T_B(X)$. Among them, $X = x1, ...xp$ is the dimension p characteristic vectors of the wind power. The collection will produce B results $\hat{Y}_1 = T_1(X), ...\hat{Y}_B = T_B(X)$. The $\hat{Y}_b(b = 1, ...B)$ is the predicted value of wind power about the tree b. For numerical prediction problem, $\hat{Y}$ is the mean of all trees prediction.

A complete random forest training set is established in line with the process of stochastic algorithm, and then put the independent variables into the test set, and the result of wind power prediction comes out. The basic idea of establishing the random forest model is shown in Figure 2.11.

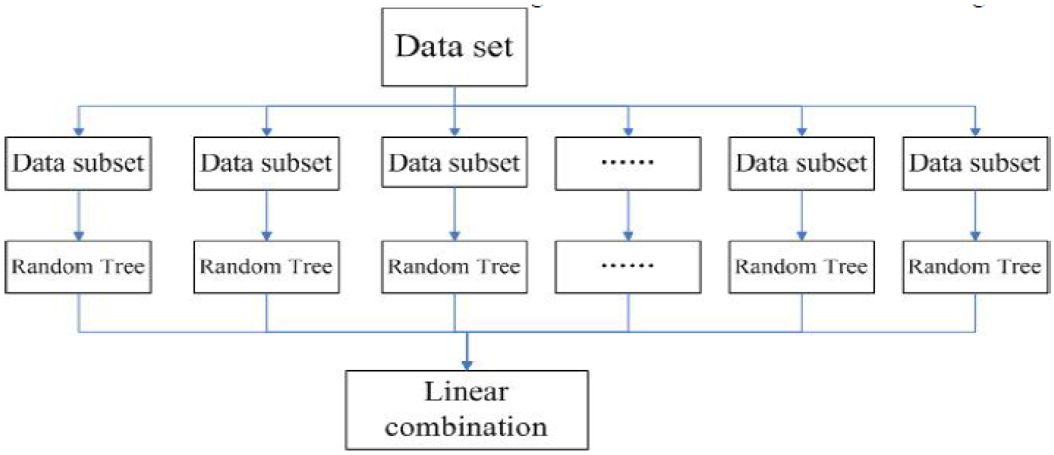


FIGURE 2.11: The establishing of a random forest model (Extracted from [45]).

2.4.2.2.2 Ensemble method

In the literature, there are several works proposing the use of a set of predictive models. The aim is to build a predictive model by combining the strengths of a collection of single base models. There are several ways of implementing this idea. Some approaches are based on re-sampling the training set, others on using different predictive methods, others on varying some parameters of the predictive methods, etc. Finally, the ensemble of methods is used to combine the output of each prediction, i.e., the predicted value, by means of a (weighted) majority voting.

Ensemble Algorithm

An ensemble model combines multiple 'individual' (diverse) models together and delivers superior prediction power.

-Bagging (Bootstrap Aggregating)

It is an ensemble method. First, we create random samples of the training data set (sub sets of training data set). Then, we build a classifier for each sample. Finally, results of these multiple classifiers are combined using average or majority voting. Bagging helps to reduce the variance error.

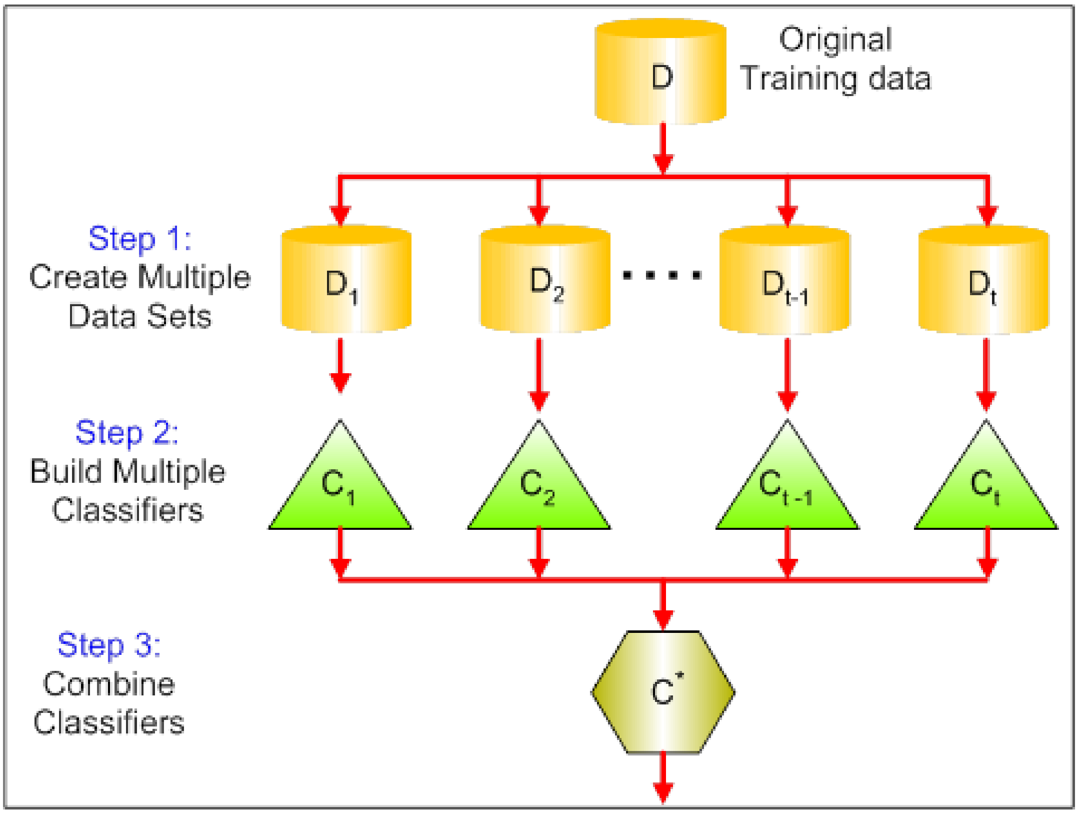


FIGURE 2.12: Bagging Algorithm.

-Boosting

It provides sequential learning of the predictors. The first predictor is learned on the whole data set, while the following are learnt on the training set based on the performance of the previous one. It starts by classifying original data set and giving equal weights to each observation. If classes are predicted incorrectly using the first learner, then it gives higher weight to the missed classified observation. Being an iterative process, it continues to add classifier learner until a limit is reached in the number of models or accuracy. Boosting has shown better predictive accuracy than bagging, but it also tends to over-fit the training data as well.

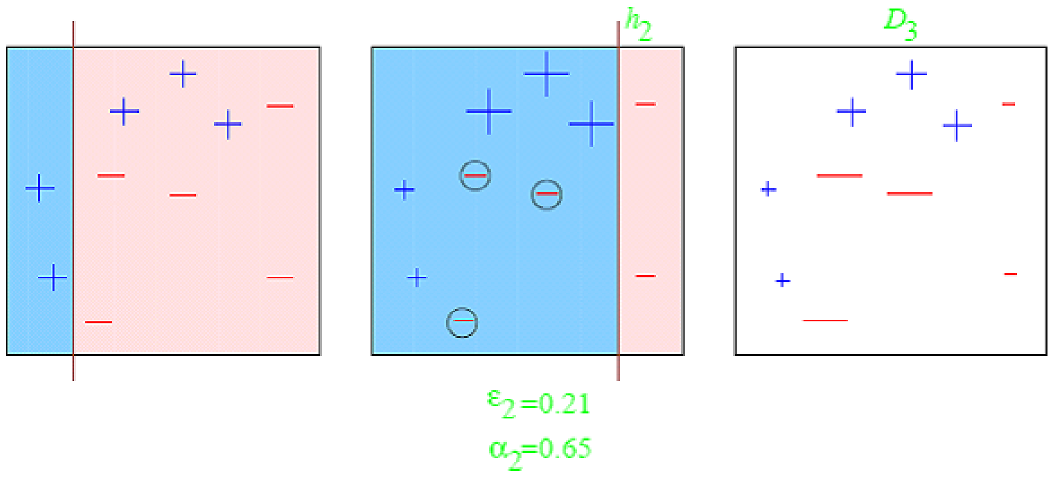


FIGURE 2.13: Boosting Algorithm.

-Stacking

It works in two phases. First, we use multiple base classifiers to predict the class. Second, a new learner is used to combine their predictions with the aim of reducing the generalization error.

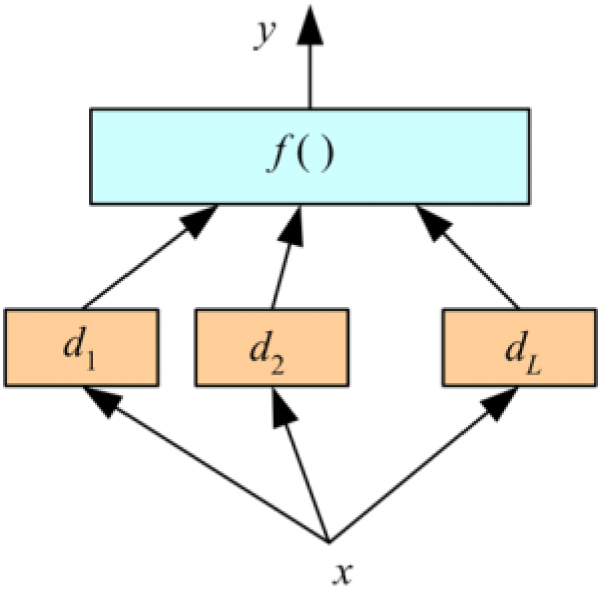


FIGURE 2.14: Stacking Algorithm.

2.5 Tools and Technologies used

2.5.1 Python

Python is a general-use high-level programming language that bills itself as powerful, fast, friendly, open, and easy to learn. Python "plays well with others" and "runs everywhere" (according to the language's About page).

Conceived in the late 1980s (and named for comedy group Monty Python), Python didn't make inroads into data science until recently. For a long time, as Tal Yarkoni of UT Austin says, “you couldn’t really do statistics in Python unless you wanted to spend most of your time pulling your hair out.”

Now, however, tools for almost every aspect of scientific computing are readily available in Python. (Thanks in part, no doubt, to the $3 million the Defense Advanced Research Projects Agency (DARPA) put toward the development of data analytics and data processing libraries for Python in late 2012.)

Bank of America uses Python to crunch financial data. The Theoretical Physics Division of Los Alamos National Laboratory chose Python to not only control simulations, but also analyze and visualize data. Facebook turns to the Python library Pandas for its data analysis because it sees the benefit of using one programming language across multiple applications.

Python’s increased use in data science applications has situated it in opposition to R, a programming language and software environment specifically designed to execute the sorts of data analysis tasks Python can now handle. As speculation mounts about whether one of the languages will eventually replace the other in the data science sphere, individuals have to decide which language to learn or which to use for a specific project.

While there are many libraries available to perform data analysis in Python, here’s a few to get you started:

- NumPy is fundamental for scientific computing with Python. It supports large, multi-dimensional arrays and matrices and includes an assortment of high-level mathematical functions to operate on these arrays.

- SciPy works with NumPy arrays and provides efficient routines for numerical integration and optimization.

- Pandas, also built on top of NumPy, offers data structures and operations for manipulating numerical tables and time series.

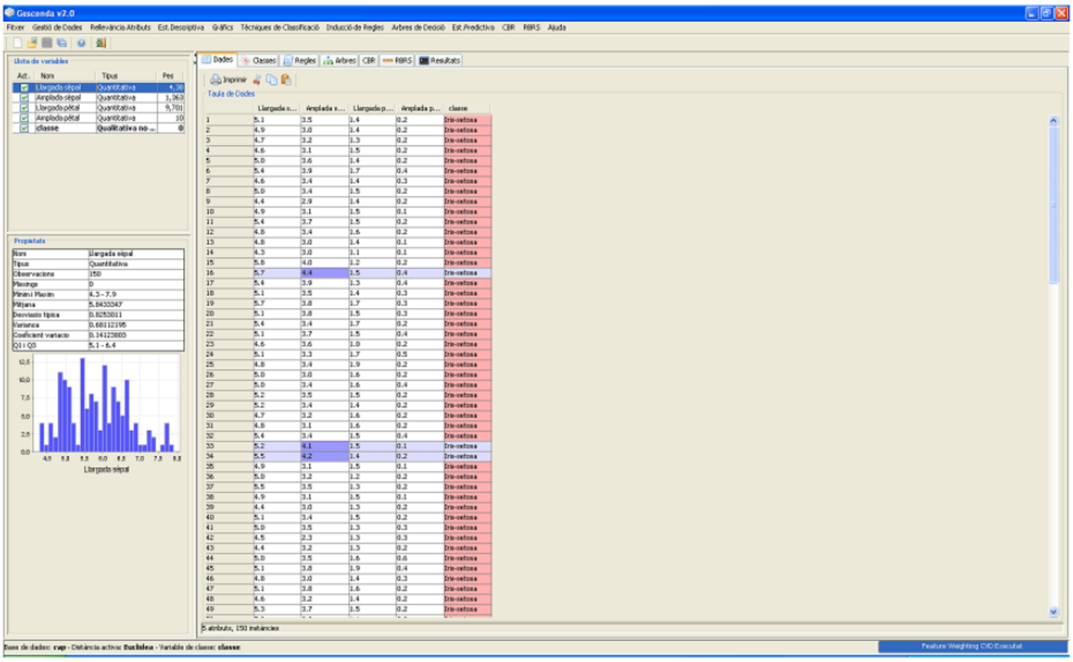
- Matplotlib is a 2D plotting library that can generate such data visualizations as histograms, power spectra, bar charts, and scatterplots with just a few lines of code.

- Built on NumPy, SciPy, and Matplotlib, Scikit-learn is a machine learning library that implements classification, regression, and clustering algorithms including support vector machines, logistic regression, naive Bayes, random forests, and gradient boosting.

2.5.2 GESCONDA

GESCONDA was designed and developed for intelligent data analysis and management of implicit knowledge from databases and also for providing the users with reasoning capabilities, with special focus on environmental databases and environmental modeling.

GESCONDA provides a set of mixed techniques that will be useful to acquire relevant knowledge from environmental systems, through available databases. This knowledge will be used afterwards in the implementation of reliable IEDSS. The portability of the software is provided by a common Java platform. In figure 2.15 there is a snapshot of the GESCONDA interface [48].



Act.	Nom	Tipus	Pes
☑	Llargada sèpal	Quantitativa	4,38
☑	Amplada sèpal	Quantitativa	1,363
☑	Llargada pètal	Quantitativa	9,701
☑	Amplada pètal	Quantitativa	10
☑	classe	Qualitativa no ...	0

Nom	Llargada sèpal
Tipus	Quantitativa
Observacions	150
Missings	0
Mínim i Màxim	4.3 - 7.9
Mitjana	5.8433347
Desviació típica	0.8253011
Variança	0.68112195
Coeficient variació	0.14123803
Q1 i Q3	5.1 - 6.4

	Llargada s...	Amplada s...	Llargada p...	Amplada p...	classe
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa
21	5.4	3.4	1.7	0.2	Iris-setosa
22	5.1	3.7	1.5	0.4	Iris-setosa
23	4.6	3.6	1.0	0.2	Iris-setosa
24	5.1	3.3	1.7	0.5	Iris-setosa
25	4.8	3.4	1.9	0.2	Iris-setosa
26	5.0	3.0	1.6	0.2	Iris-setosa
27	5.0	3.4	1.6	0.4	Iris-setosa
28	5.2	3.5	1.5	0.2	Iris-setosa
29	5.2	3.4	1.4	0.2	Iris-setosa
30	4.7	3.2	1.6	0.2	Iris-setosa
31	4.8	3.1	1.6	0.2	Iris-setosa
32	5.4	3.4	1.5	0.4	Iris-setosa
33	5.2	4.1	1.5	0.1	Iris-setosa
34	5.5	4.2	1.4	0.2	Iris-setosa
35	4.9	3.1	1.5	0.1	Iris-setosa
36	5.0	3.2	1.2	0.2	Iris-setosa
37	5.5	3.5	1.3	0.2	Iris-setosa
38	4.9	3.1	1.5	0.1	Iris-setosa
39	4.4	3.0	1.3	0.2	Iris-setosa
40	5.1	3.4	1.5	0.2	Iris-setosa
41	5.0	3.5	1.3	0.3	Iris-setosa
42	4.5	2.3	1.3	0.3	Iris-setosa
43	4.4	3.2	1.3	0.2	Iris-setosa
44	5.0	3.5	1.6	0.6	Iris-setosa
45	5.1	3.8	1.9	0.4	Iris-setosa
46	4.8	3.0	1.4	0.3	Iris-setosa
47	5.1	3.8	1.6	0.2	Iris-setosa
48	4.6	3.2	1.4	0.2	Iris-setosa
49	5.3	3.7	1.5	0.2	Iris-setosa

5 atributs, 150 instàncies

Base de dades: cvp - Distància activa: Euclidea - Variable de classe: classe

FIGURE 2.15: Gesconda interface.

2.6 Evaluation metrics

The models with simple structure provide better generalization ability, but it may not be easy to learn the issue. While in complex structure models learning can be done easily, but it is slow and achieves poor generalization performance due to overfitting. The proposed models confirms that even though we keep the structure of our models simple they obtain stable performance on training and prediction. The ultimate aim was to fix the models for wind speed forecasting application with better accuracy and minimal statistical error. The below given evaluation metrics are usual evaluation metrics used in numerical predictions to select the models. The proposed approach performance is analyzed based on the mean absolute error (MAE) and the root mean square error (RMSE) evaluation metrics as well as the determination coefficient (R2) and explained variance (EV). Among the proposed different criterias, a suitable of parameters for each model are proposed based on the minimum error performance. The statistical error metrics employed for the best models are:

2.6.1 RMSE: Root mean squared error

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors or deviations—that is, the difference between the estimator and what is estimated. MSE is a risk function, corresponding to the expected value of the squared error loss or quadratic loss. The difference occurs because of randomness or because the estimator doesn't account for information that could produce a more accurate estimate [49].

The MSE is a measure of the quality of an estimator it is always non-negative, and values closer to zero are better.

The MSE is the second moment (about the origin) of the error, and thus incorporates both the variance of the estimator and its bias. For an unbiased estimator, the MSE is the variance of the estimator. Like the variance, MSE has the same units of measurement as the square of the quantity being estimated. In an analogy to standard deviation, taking the square root of MSE yields the root-mean-square error or root-mean-square deviation (RMSE or RMSD), which has the same units as the quantity being estimated; for an unbiased estimator, the RMSE is the square root of the variance, known as the standard deviation.

$$RMSE_{y,\hat{y}} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{2.13}$$

In numerical prediction analysis, the term mean squared error is sometimes used to refer to the unbiased estimate of error variance: the residual sum of squares divided by the number of degrees of freedom. This definition for a known, computed quantity differs from the above definition for the computed MSE of a predictor in that a different denominator is used. The denominator is the sample size reduced by the number of model parameters estimated from the same data, (n-p) for p regressors or (n-p-1) if an intercept is used.[50]

2.6.2 MAE: Mean absolute error

In statistics, mean absolute error (MAE) is a measure of difference between two continuous variables. Assume X and Y are variables of paired observations that express the same phenomenon. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement.

The Mean Absolute Error is given by:

$$MAE_{y,\hat{y}} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{2.14}$$

The mean absolute error is one of a number of ways of comparing forecasts with their eventual outcomes.

2.6.3 R2: Determination Coefficient

In statistics, the coefficient of determination, denoted R^2 or r^2 and pronounced "R squared", is the proportion of the variance in the dependent variable that is predictable from the independent variable(s) [51].

It is a statistic used in the context of statistical models whose main purpose is either the prediction of future outcomes or the testing of hypotheses, on the basis of other related information. It provides a measure of how well observed outcomes are replicated by the model, based on the proportion of total variation of outcomes explained by the model [**59**, 50, 52].

There are several definitions of R2 that are only sometimes equivalent. One class of such cases includes that of simple linear regression where r2 is used instead of R2. When an intercept is included, then r2 is simply the square of the sample correlation coefficient (i.e., r) between the observed outcomes and the observed predictor values. If additional regressors are included, R2 is the square of the coefficient of multiple correlation. In both such cases, the coefficient of determination ranges from 0 to 1.

$$FVU = \frac{MSE(y, \hat{y})}{var[Y]} \tag{2.15}$$

$$R^2 = 1 - FVU \tag{2.16}$$

2.6.4 EV: Explained variance

In statistics, explained variation measures the proportion to which a mathematical model accounts for the variation (dispersion) of a given data set. Often, variation is quantified as variance; then, the more specific term explained variance can be used.

$$FVU = \frac{\sum(X - \bar{X})^2}{n - 1} \tag{2.17}$$

Chapter 3

Methodology description

In this chapter we present the methodology we propose to solve the wind speed prediction problem we presented in the previous chapters for a specific atmospheric reanalysis data of a wind farm in Spain. We will start by clearly explaining the main idea and the intuition behind it. Then we will explain the data science methodology we will apply such as the data understanding, data preparation steps that are to be performed and finally the modeling proposals for the methodology.

3.1 General Idea

As stated in the previous chapter, there are several machine learning methods proposed in the literature to solve the wind prediction problem.

We believe that combined forecasting methodologies, which follow a different approach and produce the final forecast generally from the combination of single approaches, can be a more viable solution for improving the accuracy of the individual single models.

Also we know that the wind data used in the training process of the forecasting models is of significant importance in terms of accuracy of forecasting model and computational time. In the literature, prediction accuracy is often reported as proportional to the amount of input data and computational time. However, it can be possible to obtain good prediction accuracy while utilizing only a subset of the data, following by a reduction in required time of prediction process. In this sense, grouping the wind data according to the similarities between the observations and filtering the similar groups in order to determine the most relevant input parameters is a widely preferred method for this objective.

Most applied methods to Wind Speed prediction found on the literature used physical model approaches or some statistical approaches. Here in this work several data-driven approaches will be explored using several machine learning techniques. Therefore a data science project emerge in our work.

3.2 Data Science Methodology

In this chapter the use of the standard data science methodology is proposed for the Wind Speed prediction.

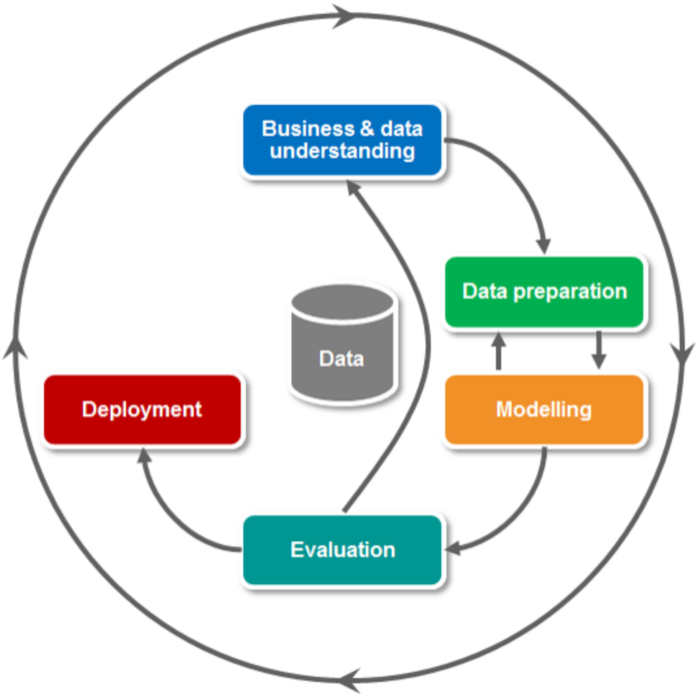


FIGURE 3.1: Data Science methodology.

As shown in the figure 3.1 the main steps are: Business and data understanding, data preparation, modeling, evaluation and finally results interpretation or deployment.

3.3 Business and Data understanding

Before working with the data we need to understand the problem which was deeply analyzed in previous chapters. Thus, we will use some visualization techniques to help us understand the data content, assess the quality of the data and discover some initial insights into the data. We will describe the amount of data we have to work, verify data types, data quality, check if we have some missing values, error values, inconsistence, etc. We will perform all this data understanding tasks by means of a data science tool developed in the UPC called GESCONDA previously described in chapter 2.

3.4 Data preparation

The data preparation stage comprises all activities used to construct the data set that will be used in the modeling stage. These include data cleaning, combining data from multiple sources and transforming data into more useful variables. Moreover, feature engineering and text analytics may be used to derive new structured variables, enriching the set of predictors and enhancing the model's accuracy.

In this stage, as we can see in the figure 3.2 we will prepare the data by selecting the right data to work with, selecting features, performing some cleaning, filling the missing values and correcting the data errors to be able to proceed to the modeling stage.

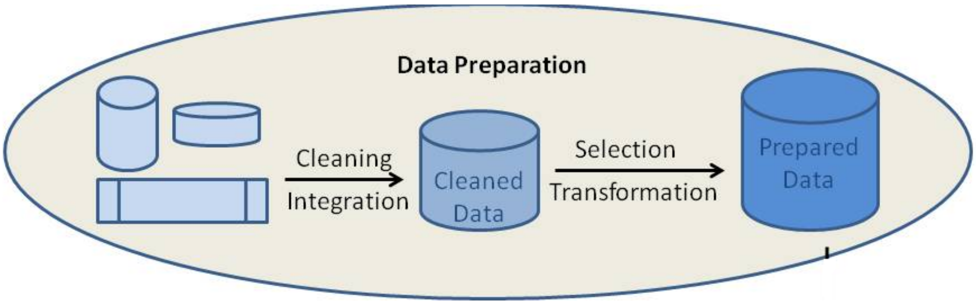


FIGURE 3.2: Data preparation.

3.4.0.1 Data Cleaning

Missing data may be due to some equipment malfunctioning or inconsistency with other recorded data, and thus can exist deleted data or not entered ones. Due to misunderstanding certain data may not be considered important at the time of entry. It is important to note that, a missing value may not always imply an error.

In this work, we propose to complete the missing values by replacing them with the mean estimated by the 2 previous and 2 past observation for the same time range, this in order to not discard these available information about the pressure temperature and wind components.

3.4.0.2 Data Transformation

Data Transformation can involve the following tasks:

- Smoothing: remove noise from the data, including binning, regression and clustering
- Aggregation
- Generalization
- Normalization
- Attribute construction.

In this work we will normalize the data by standardization. Normalization of a dataset is a common requirement for many machine learning estimators: they might behave badly if the individual feature do not look like standard normally distributed data (e.g. Gaussian with 0 mean and unit variance).

For instance many elements used in the objective function of a learning algorithm (such as the RBF kernel of Support Vector Machines or the L1 and L2 regularizers of linear models) assume that all features are centered around 0 and have variance in the same order. If a feature has a variance that is orders of magnitude larger that others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected.

3.4.0.3 Data selection

Since the data provided comes from a data reanalysis and knowing that we have data from the 4 points surrounding one wind farm we will perform some feature weighting techniques to find our whether there exists some irrelevant variables in an attempt to reduce computational cost when performing our experiments.

3.5 Modeling

Starting with the first version of the prepared data set, data scientists use as training set historical data in which the outcome of interest is known to develop predictive models using the analytic approach already described. The modeling process is highly iterative. The steps of the modeling phase will be selecting the modeling techniques, selecting the data types available for analysis, selecting an algorithm or a model, defining the modeling goals, stating specific modeling requirements and finally building the model and training the model.

As we explained before in our general idea we plan to explore several methods used for Wind speed prediction divided in two main groups, global models and local models.

In the following figure 3.3 we can see the high level diagram of the proposed methodology, how we plan to define our global models and our local models.

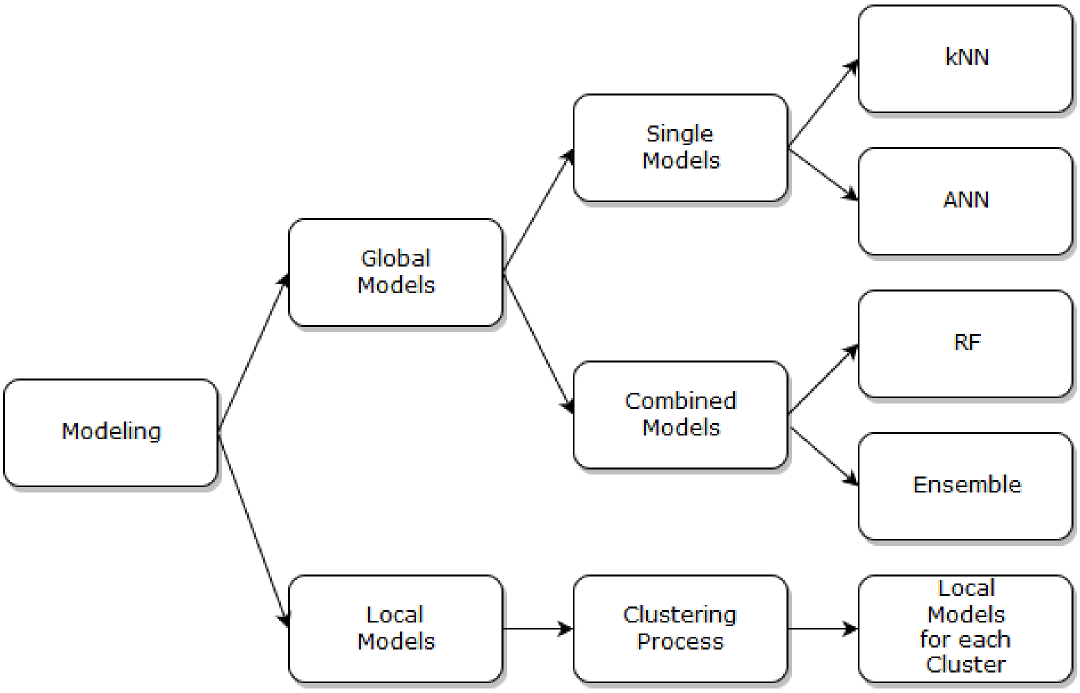


FIGURE 3.3: High Level design of the proposed methodology.

3.5.1 Global models

In the figure 3.4 we can see the proposed Global modeling.

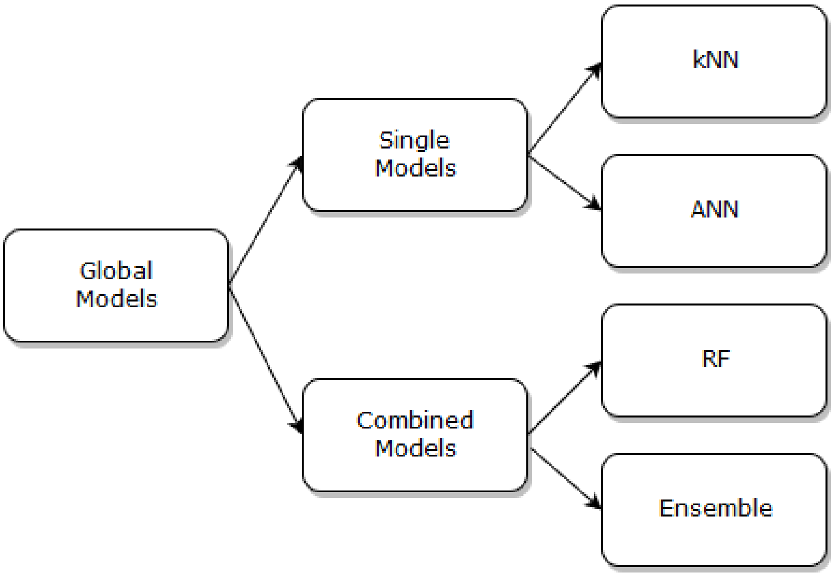


FIGURE 3.4: Global modeling methodology proposed.

After different tests to select the best global models we decided to work with specifically Random Forest, K-nearest neighbors, Artificial Neural Networks and Ensemble of the three mentioned for the numerical prediction of the wind speed.

3.5.1.1 Single Models

As single models we selected to work with ones shown in figure 3.5.

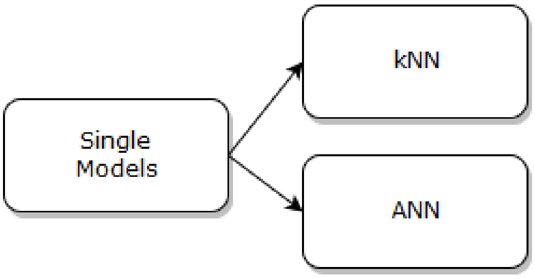


FIGURE 3.5: Single models.

3.5.1.1.1 K Nearest Neighbor model

K Nearest Neighbor is one of those algorithms that are very easy to understand but works incredibly well in practice. It is also one of the top 10 data mining algorithms.

K nearest neighbors is an algorithm that stores all available cases and predict the numerical target based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique.

For the selection of the KNN algorithm for our second global model we thought about following reasons:

- It's 100% nonparametric

- It's local, so it can pick up on weird systematic but localized features of data without structural breaks or mixtures

For the same reason, it works on arbitrarily nonlinear data, is a simple model but yet powerful.

3.5.1.1.2 ANN model

Artificial Neural Networks can model nonlinearities, which one would need to explicitly model using transformations in linear regression. ANNs are able to model complex relationships, and thus, it could be a good candidate approach to model the complex wind speed prediction problem.

3.5.1.2 Combined Models

As combined models we selected to work with ones shown in figure 3.6.

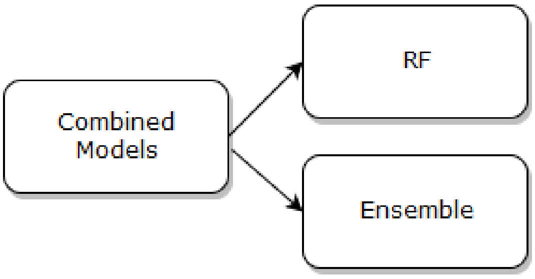


FIGURE 3.6: Combined models.

3.5.1.2.1 Random Forest model

We know decision trees do a great job at capturing the non-linearity in the data by dividing the space into smaller sub-spaces depending on the problem and in this case since we have a numeric dataset we find very useful to have the decision trees capabilities.

- We want to use something more interpretable, something that trains faster and performs pretty much just as well as the Logistic Regression or even Neural Networks.
- Much faster to train versus simple neural networks for comparable performance.
- Easily interpretable, suitable for variable selection
- Fairly robust on smaller datasets
- Decision Trees and Decision Tree Learning are simple to understand

3.5.1.2.2 Ensemble model

In the following figure 3.7 we can see the proposed methodology for the combined ensemble modeling.

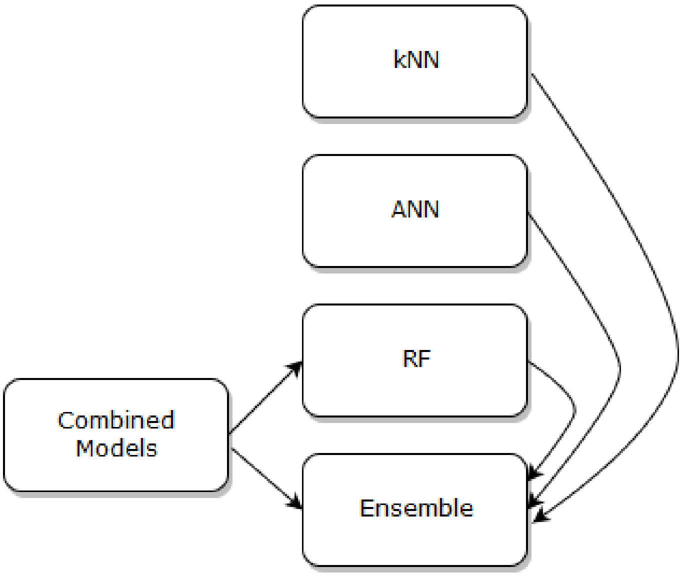


FIGURE 3.7: Ensemble methodology proposed.

Ensemble model combines multiple 'individual' (diverse) models together and delivers superior prediction power.

Basically, an ensemble is a supervised learning technique for combining multiple weak learners/ models to produce a strong learner. Ensemble model works better, when we ensemble models with low correlation.

Bagging (Bootstrap Aggregating) is an ensemble method. First, we create random samples of the training data set (sub sets of training data set). Then, we build a classifier for each sample. Finally, results of these multiple classifiers are combined using average or majority voting. Bagging helps to reduce the variance error. We will use a form of bagging ensemble method except for the random samples part where we will use the whole training set to build and train each of our global models and then finally average the output.

There are two major benefits of Ensemble models:

- Better prediction
- More stable model

Combining multiple predictions generated by different algorithms would normally give better predictions. It is due to the diversification or independent nature as compared to each other. For example, the predictions of a random forest, a KNN, and a ANN may be combined to create a stronger final prediction. The key to creating a powerful ensemble is model diversity. In general, we assume equal weight for all models and taking the average of predictions.

3.5.2 Local models

As it will be detailed in the next chapter, after a clustering analysis of our data, it was found that the most reasonable number of groups was three. Notwithstanding, as it will be explained, the separation of the three clusters was not very good.

The intuition behind it is that training a model with more similar elements would give a more specialized model for those kind of data. As it was shown in [53] the results of the experiments showed that the meta-case retrieval with multiple case libraries approach improved the standard case retrieval with only one library.

In the following figure 3.8 we illustrate this main idea where we have one Local Model (LM) for each of the cluster of our dataset.

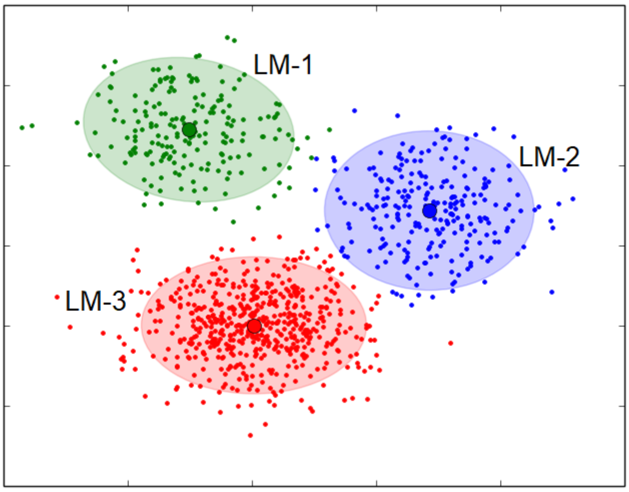


FIGURE 3.8: Methodology intuition.

In this figure 3.9 we can see the local modeling methodology proposed.

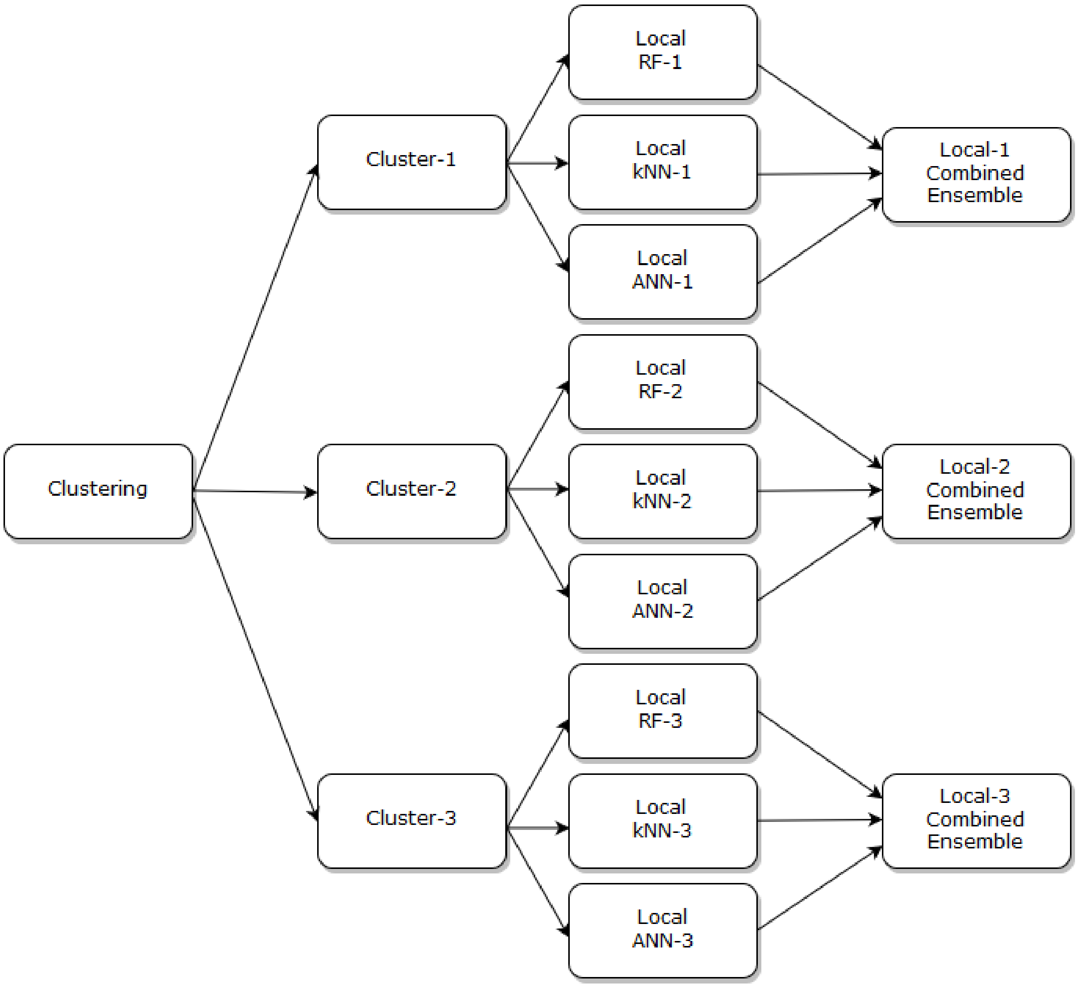


FIGURE 3.9: Local modeling methodology proposed.

After different tests to select the best k for separating the data into different clusters we decided to work with the k=3 specifically because this division would make sense in terms to the relation between temperature and wind speed, dividing the data into the ones with low temperatures, the ones with high temperature and the ones medium temperatures. These also related with the seasons. For example dates from Winter and Fall were group together and Summer and Spring also grouped together. And finally, the third would have a little of both for the medium temperature days.

3.5.2.1 Clustering model

We selected to work with the K-means algorithm for the creation of the clusters for the building of our local models. We selected the algorithm based on its advantages:

- practically work well even when some assumptions of balanced cluster are broken such as size within the dataset, joint distribution of features within each cluster is spherical, clusters have similar density.
- simple and easy to implement,
- easy to interpret the clustering results,
- fast and efficient in terms of computational cost, typically $O(K * n * d)$.

3.6 Evaluation

In this step of the methodology we proceed to the interpretation of the results, evaluation and validation.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models. There are two common methods of evaluating models in data science: Hold-Out and Cross-Validation. To avoid overfitting, both methods use a test set (not seen by the model) to evaluate model performance.

During the training we use the cross-validation method to choose among all the models the best one, and finally we Hold-out 20% of the dataset to use as unseen observations to perform our final testing.

During the testing we use different evaluation measures to be able to compare the performance between the different models.

After building a number of different numerical prediction models, there is a wealth of criteria by which they can be evaluated and compared. The evaluation metrics used will be RMSE, MAE, R2 and EV which were described in previous chapters.

Root Mean Squared Error

RMSE is the most used formula to measure the error rate of a numerical prediction model. However, it can only be compared between models whose errors are measured in the same units.

Mean Absolute Error

The mean absolute error (MAE) has the same unit as the original data, and it can only be compared between models whose errors are measured in the same units. It is usually similar in magnitude to RMSE, but slightly smaller.

Coefficient of Determination

The coefficient of determination (R2) summarizes the explanatory power of the numerical prediction model and is computed from the sums-of-squares terms.

R2 describes the proportion of variance of the dependent variable explained by the numerical prediction model. If the numerical prediction model is "perfect", SSE is zero, and R2 is 1. If the numerical prediction model is a total failure, SSE is equal to SST, no variance is explained by regression, and R2 is zero.

Explained variance

Explained variation measures the proportion to which a numerical prediction model accounts for the variation (dispersion) of a given data set. The usual definition of the coefficient of determination above explained is based on the fundamental concept of explained variance.

Chapter 4

Methodology application

In this chapter we present the application of the methodology that we previously described in order to explore the benefits of several approaches like global models vs local models and single models vs combined models.

4.1 Data Understanding

In order to test the proposed combined approach, we have carried out a number of experiments, which use real data from one wind farm in Spain, whose location is shown in Figure 4.1. The same as used in a previous paper [54] and which was provided by one of the authors for further research using reanalysis data feasibility in wind speed prediction.

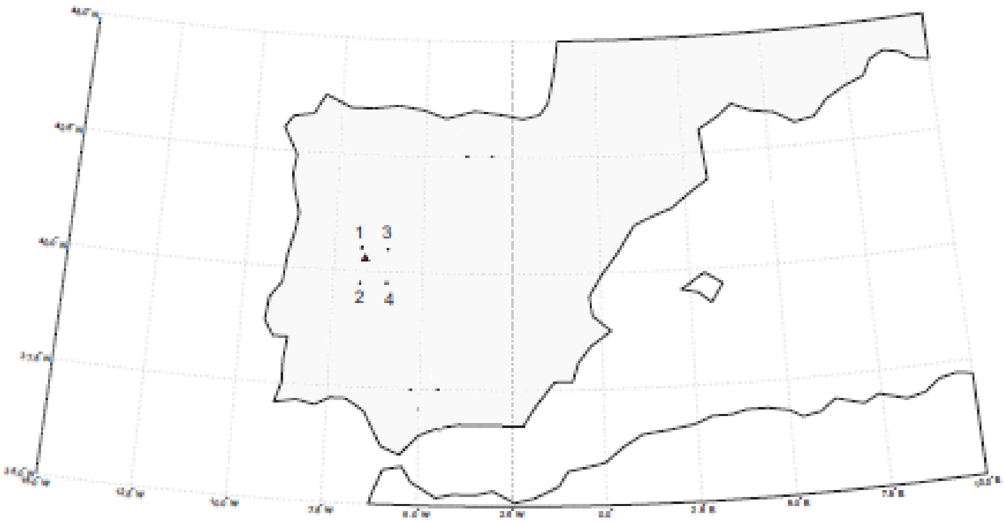


FIGURE 4.1: The Wind farm considered for the experiments. The four closest nodes from the Era-Interim reanalysis database (where predictive variables are calculated) are also displayed in the picture (Extracted from [55]).

The peculiarity of the predictive variables considered for this problem of wind power prediction in 4.1 is that these predictors are reanalysis predictive variables as explained in chapter 2.

That means that these variables come from a reanalysis project, which consists of combining past observation with a modern meteorological forecast model. In this

TABLE 4.1: Predictive variables considered at each node from the ERA-Interim reanalysis (Extracted from [55]).

Variable Name	**ERA-Interim variable**
skt	surface temperature
sp	surface pressure
u10	zonal wind component (u) at 10m
v10	meridional wind component (v) at 10m
temp1	Temperature at 500hPa
up1	zonal wind component (u) at 500hPa
vp1	meridional wind component (v) 500hPa
wp1	vertical wind component (!) at 500hPa
temp2	Temperature at 850hPa
up2	zonal wind component (u) at 850hPa
vp2	meridional wind component (v) at 850hPa
wp2	vertical wind component (!) at 850hPa

way, the information registered in the past is improved, producing regular gridded datasets of many atmospheric and oceanic variables, with a temporal resolution of few hours. The reanalysis projects usually extends over several decades and cover the entire planet. There are several current reanalysis projects at global scale, but maybe the most important one is the ERA-Interim reanalysis project, produced by the European Centre for Medium-Range Weather Forecasts (ECMWF), and which is used in this work. The main characteristics of the model are that the spatial resolution of the data set is approximately 15 km, on 60 vertical pressure levels from the surface up to 0.1 hPa. Moreover, Era-Interim provides 6-hourly atmospheric fields on model levels, pressure levels, potential temperature and potential vorticity. In order to tackle the wind power prediction in this work, we consider wind, pressure and temperature variables from ERA-Interim at four points around one specific wind farm under study.

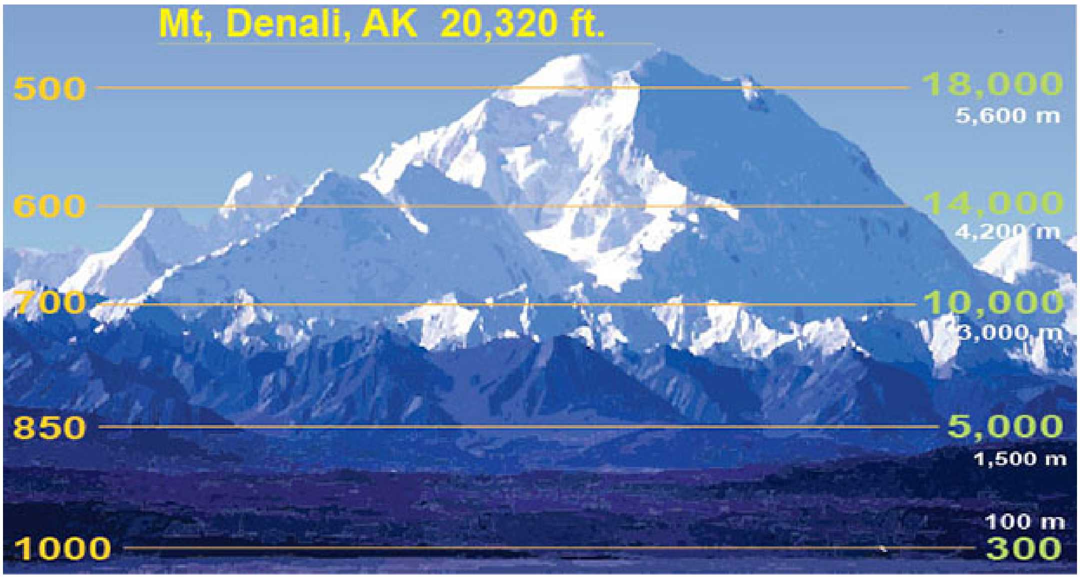


FIGURE 4.2: Atmospheric Pressure levels

The predictive variables presented in the table 4.1 are taken at different pressure levels (surface, 850 hPa, 500 hPa) as we can see in the figure 4.2. As a result, we have 12 predictive variables per four point around the wind farm considered, at time *t*. This give a total of 48 predictive variables [55].

Some of the predictive variables also include wind components in order to understand what wind components are they are shown in figure 4.3

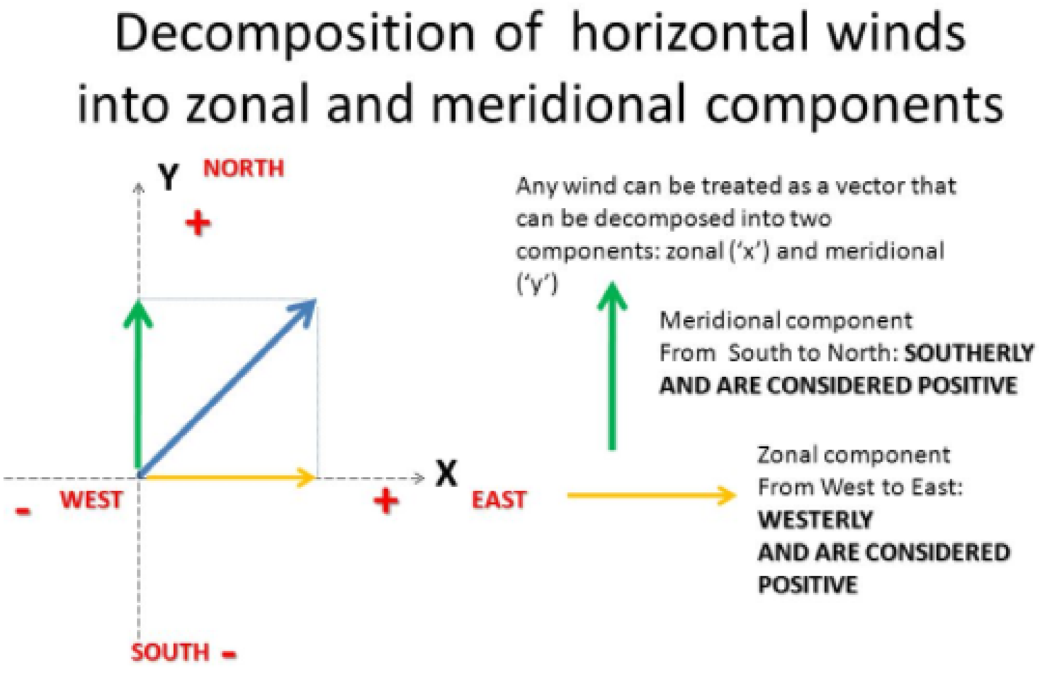

Decomposition of horizontal winds
into zonal and meridional components
Y
NORTH
+
X
EAST
+
WEST
-
SOUTH -
Any wind can be treated as a vector that can be decomposed into two components: zonal ('x') and meridional ('y')
Meridional component
From South to North: SOUTHERLY
AND ARE CONSIDERED POSITIVE
Zonal component
From West to East:
WESTERLY
AND ARE CONSIDERED
POSITIVE

FIGURE 4.3: Wind Components

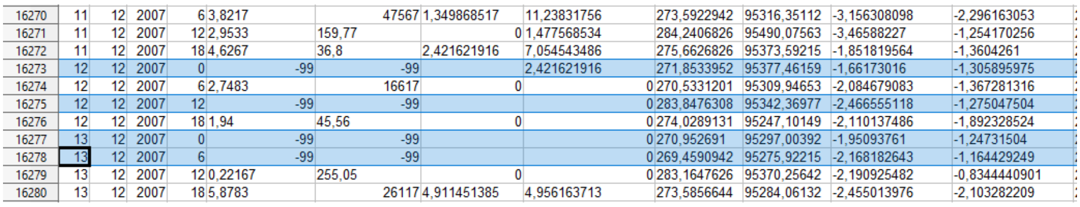
Summarizing, the initial dataset matrix has 22.815 rows and 55 variables in the original data matrix with observations ranging from 23/11/1995 to 17/02/2013. The variables are the 48 variables coming from the reanalysis, and the hour, date, month, year, wind power, wind direction and wind speed. The wind speed variables is the variable of interest, which we try to predict.

4.2 Data Preparation

The data available for this wind farm ranges from 23/11/1995 to 17/02/2013. Note that we only kept data every 6 hours (00h, 06h, 12h and 18h), to match the predictive variables from the ERA-Interim to the objective variables.

4.2.1 Data cleaning

During the data analysis we discovered that some values in the original data matrix for Wind speed, direction and power were missing 4.4 by having the first two -99 value and power value empty.



16270	11	12	2007	6	3,8217	47567	1,349868517	11,23831756	273,5922942	95316,35112	-3,156308098	-2,296163053
16271	11	12	2007	12	2,9533	159,77	0	1,477568534	284,2406826	95490,07563	-3,46588227	-1,254170256
16272	11	12	2007	18	4,6267	36,8	2,421621916	7,054543486	275,6626826	95373,59215	-1,851819564	-1,3604261
16273	12	12	2007	0	-99	-99		2,421621916	271,8533952	95377,46159	-1,66173016	-1,305895975
16274	12	12	2007	6	2,7483	16617	0	0	270,5331201	95309,94653	-2,084679083	-1,367281316
16275	12	12	2007	12	-99	-99		0	283,8476308	95342,36977	-2,466555118	-1,275047504
16276	12	12	2007	18	1,94	45,56	0	0	274,0289131	95247,10149	-2,110137486	-1,892328524
16277	13	12	2007	0	-99	-99		0	270,952691	95297,00392	-1,95093761	-1,24731504
16278	13	12	2007	6	-99	-99		0	269,4590942	95275,92215	-2,168182643	-1,164429249
16279	13	12	2007	12	0,22167	255,05	0	0	283,1647626	95370,25642	-2,190925482	-0,8344440901
16280	13	12	2007	18	5,8783	26117	4,911451385	4,956163713	273,5856644	95284,06132	-2,455013976	-2,103282209

FIGURE 4.4: Missing values in the data.

To complete the dataset we proceed to replace this missing values with the mean estimated by the 2 previous and 2 past observation for the same time range, this in order to not discard these available information about the pressure temperature and wind components.

However only the wind speed variable was considered in the study, the other two variables such as the direction and power were removed from the dataset.

$$ws(t) = (ws(t-1) + ws(t-2) + ws(t+1) + ws(t+2))/4 \tag{4.1}$$

where t is the same time range (i.e.: 00 , 06, 12, 18).

As part of the cleaning process we decided to remove some of the variables that were not used during the experimentation, as such we decided to discard after some testing the hour, date, month and year (see Appendix tables A) of the observations and also the direction and power of the wind. Thus finally, a set of 49 variables were used.

4.2.2 Data Transformation

Normalization of data is essential as the variables used for the study are of different units. Therefore, the data are scaled within the range 0–1 using the Scikit-learn library for python language, specifically the StandardScarler class.

```
class sklearn.preprocessing.StandardScaler(copy=True,
with_mean=True, with_std=True)
```

The class standardize features by removing the mean and scaling to unit variance.

Centering and scaling happen independently on each feature by computing the relevant statistics on the samples in the training set. Mean and standard deviation are then stored to be used on later data using the transform method.

4.2.3 Data Selection

In this step we applied the Unsupervised Entropy-based algorithm to calculate each of the 48 variables weights. With this process we expected to learn whether there are any irrelevant variables among the 48 previously described.

After running the algorithm we obtained the resulting weights in table 4.2. In it we can see that each cell contains the group of the same variable for the 4 points surrounding the wind farm. With this results we can see each variable seems to be equally relevant with the ones from the other points, for example the pressure in surface for the 4 points obtains almost the same weight, going from 9.5 for the 1st point to 10 to the 4th point. The same happens for the predictive variable temperature at 500 hpa for the 4 points, going from 8.6 for the 1st point to 8.7 for the last point. It happens the same for each of the 12 predictive variables.

With this experiment we can hypothesize that running the predictive models over the 48 predictive variables could give the same results than running the same model with only a subset of the dataset using just the 12 predictive variables for one of the 4 points.

TABLE 4.2: Predictive variables weights calculated by UEB-1 algorithm

Name	Weight	Name	Weight	Name	Weight	Name	Weight
skt-4	10.00	up1-4	5.12	sp-4	4.45	vp2-1	3.19
skt-3	9.91	up1-3	4.99	sp-2	4.44	vp2-3	3.12
skt-2	9.58	up1-2	4.86	u10-1	4.33	vp2-2	3.10
skt-1	9.51	up1-1	4.75	u10-2	4.31	vp2-4	3.02
temp2-2	8.70	vp1-2	4.68	up2-2	3.91	wp2-2	1.69
temp2-4	8.66	vp1-1	4.65	up2-1	3.90	wp2-1	1.69
temp2-1	8.65	vp1-4	4.60	up2-3	3.81	wp2-3	1.61
temp2-3	8.63	vp1-3	4.57	up2-4	3.80	wp2-4	1.56
temp1-3	7.97	u10-3	4.56	v10-2	3.63	wp1-3	0.03
temp1-1	7.95	u10-4	4.55	v10-1	3.58	wp1-4	0.02
temp1-4	7.93	sp-3	4.48	v10-4	3.45	wp1-1	0.01
temp1-2	7.91	sp-1	4.46	v10-3	3.41	wp1-2	0.00

The further experimentation has confirmed this hypothesis as shown in tables 4.3 and 4.4 where we tested using Random Forest and kNN with their default parameters using 48 variables vs 12 from one point.

TABLE 4.3: RandomForest feature selection results

N of features	Score time	RMSE	MAE	R2	EV
48	0.048	2.73	2.06	0.676	0.679
12	0.027	2.76	2.12	0.63	0.63

TABLE 4.4: KNeighbors selection results

N of features	Score time	RMSE	MAE	R2	EV
48	11.28	2.93	2.22	0.62	0.63
12	3.32	2.93	2.23	0.59	0.59

4.3 Modeling

In this step of the work we proceed to train our models for comparing their performance.

For all experiments we will use the already preprocessed data, and we split the dataset into 80% for training and 20% for testing. During the training we also preform 10 cross-validation.

In the following figure 4.5 we show the methodology to be used. We will first proceed to train our global models and later the local models to finally compare the testing performance of both.

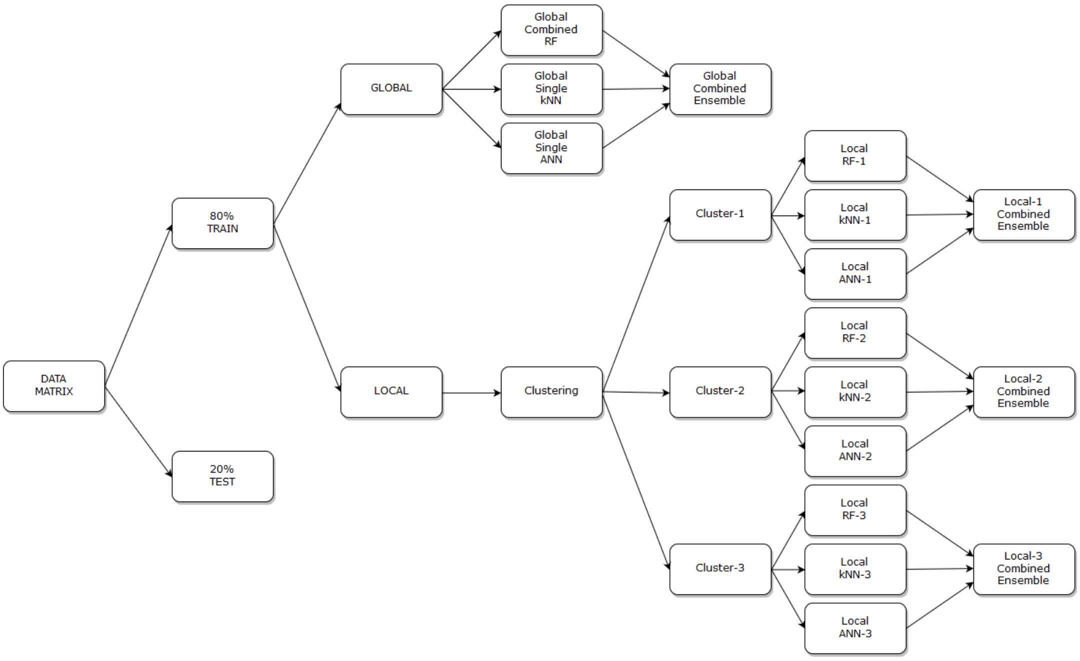


FIGURE 4.5: Training process flowchart.

In the following figure 4.6 we can see how the testing will be performed.

For the remaining 20% of unseen observations that we saved for testing, the corresponding global model was used (kNN, ANN, RF, ensemble), or the local model, where we proceed to assign the cluster class by calculating the distance to the previously calculated centroids and then we select the best model for that cluster either being the local, global or ensemble to predict the final output.

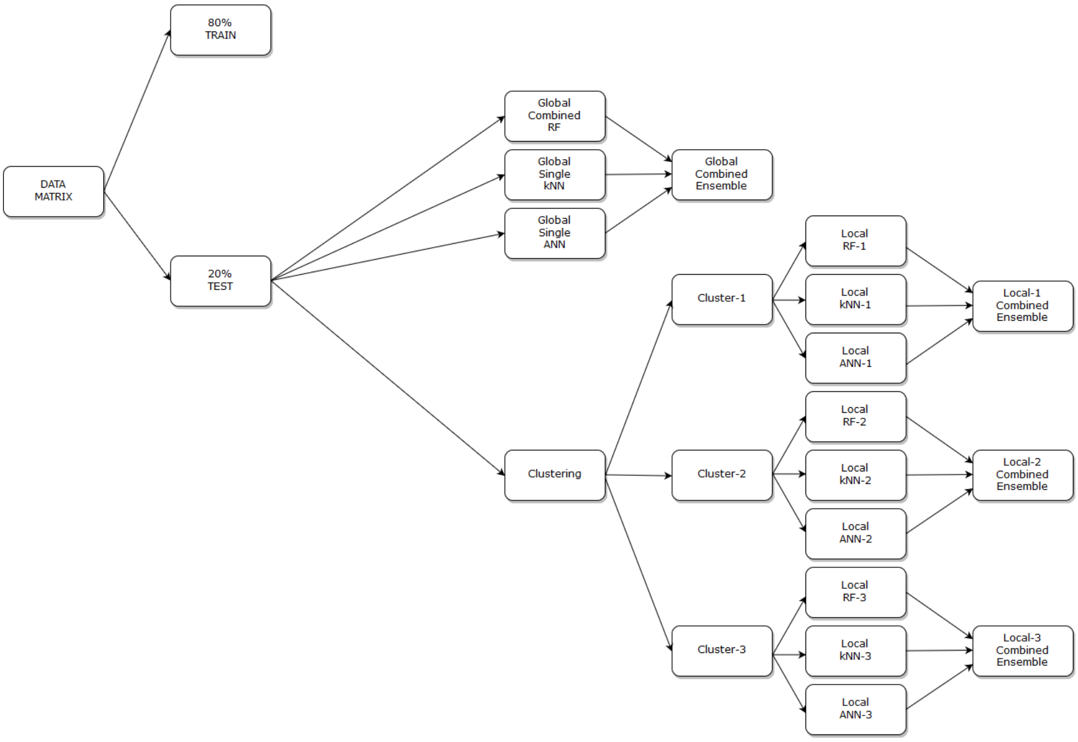


FIGURE 4.6: Testing process flowchart.

4.4 Global Models

In the following figure 4.7 we can see how the global models are going to be trained.

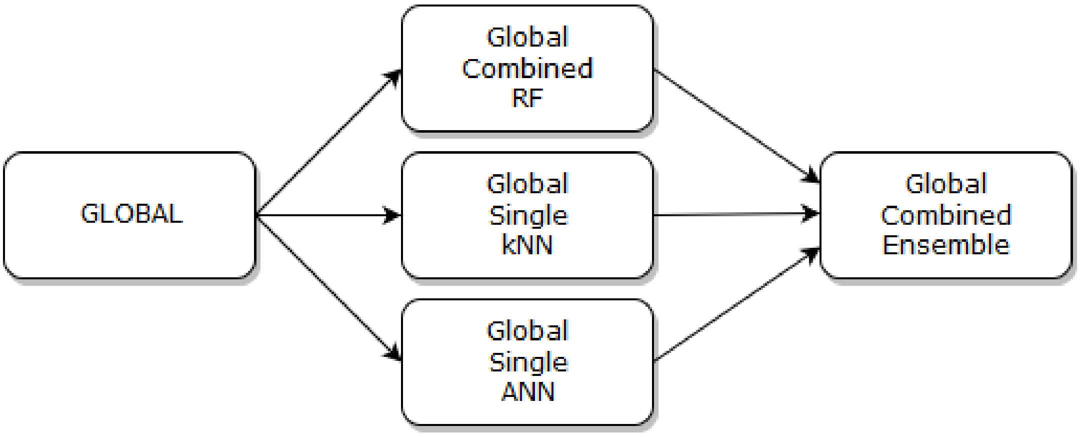


FIGURE 4.7: Global training process.

4.4.1 Single Models

4.4.1.1 K Nearest Neighbors Model

We use KNeighborsRegressor function from Scikit-learn which is a free software machine learning library for the Python programming language to verify the effectiveness of this method in this thesis.

The KNeighborsRegressor function of Scikit-learn target is predicted by local interpolation of the targets associated of the nearest neighbors in the training set.

```
class sklearn.neighbors.KNeighborsRegressor(n_neighbors=5,
weights='uniform', algorithm='auto', leaf_size=30, p=2,
metric='minkowski', metric_params=None, n_jobs=1, **kwargs)
```

Among them, *n_neighbors* represents the number of neighbors to take into account when performing the local interpolation.

We performed k=10 fold cross-validation run over 18.252 observations ranging from 23/11/1995 to 17/02/2013 which represents the 80% of the dataset. For each kfold 16.426 samples are randomly selected as training set and 1.816 as validation samples. We HOld-out 20% of the observations to later perform the testing phase on the unseen observations.

The main approach for the selection of parameters was trial and error and after some evaluation trial we came up with the best parameters presenting their particular results in the following table.

We present now the table 4.5 containing the cross-validation mean results of the training experiments performed in order to select the best kNN model based on the selection of the k.

TABLE 4.5: KNeighbors CV training results under different ks taking into account all 12 predictive variables for 1 point.

K	Score time	RMSE	MAE	R2	EV
3	2.86	3.08	2.35	0.54	0.55
5	3.32	2.93	2.23	0.59	0.59
8	3.27	2.86	2.18	0.61	0.61
10	2.70	2.84	2.16	0.61	0.62
25	3.45	2.83	2.16	0.62	0.62

In this table 4.5, MAE is short name for Mean Absolute Error and RMSE is the short name for Root Mean Square Error, R2 is short name for Determination Coefficient and EV short name for Explained Variance as explained in previous chapter. We can see from the results that for 8, 10 and 25 the improvement is minimal although the computation time tends to increase so for our best k we will choose k=10.

The model was then verified using a test sample of 20% of unseen observations. Figure 4.8 shows wind speed prediction of the test set of 20% unseen observations versus the actual wind speed of those observations.

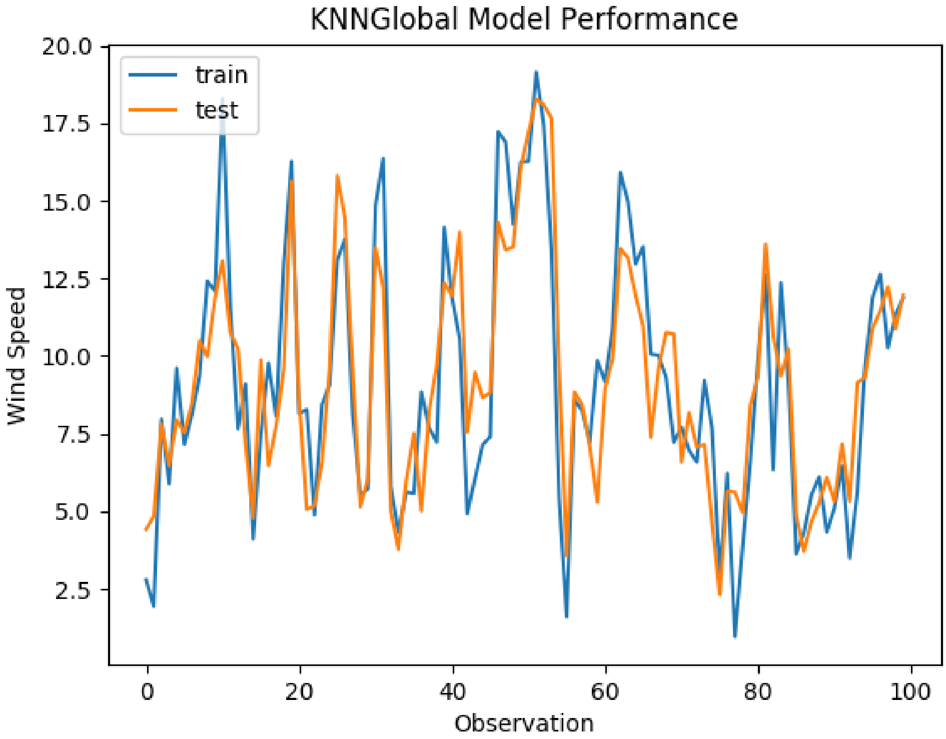


FIGURE 4.8: KNN Test Prediction

Figure 4.8 shows the kNN algorithm for wind speed prediction is feasible, and the table 4.5 also lists the mean absolute error and root mean square error under different number of selected k neighbors. We only show the results of applying the standard euclidean distance.

TABLE 4.6: KNN test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Time	RMSE	MAE	R2	EV
1.71	2.90	2.17	0.62	0.62

4.4.1.2 Artificial Neural Network Model

A feed forward neural network with backpropagation algorithm is implemented using Class KerasRegressor function from Scikit-learn which is a free software machine learning library for the Python programming language for the prediction model. Training is performed in batch mode of 500 with 200 epochs.

The database is constructed with the 18-year data from November 1995 to February 2013 as mentioned before. From this whole set of data 80% is used for training the neural network with k=10 fold cross-validation and 20% is used for testing the model.

One of the most vital tasks in constructing the ANNs is the choice of the number of *hidden layers* and the number of *neurons*. In this study, a number of tests are performed by varying the number of *hidden layers* and the number of *neurons* in the *hidden layer*. Finally we decided to only work with one *hidden layer* and further testing was conducted to choose the number of *neurons* of that *hidden layer* obtaining the following results shown in the table 4.7.

TABLE 4.7: ANN CV training results under different units taking into account all 12 predictive variables for 1 point.

Units	**RMSE**
1	4.27
2	3.08
3	3.09
4	2.80
5	2.77
6	2.76
7	2.75
8	2.76
9	2.72
10	2.72
11	2.72
12	2.70
13	2.71
14	2.71
15	2.72
16	2.72
17	2.69
18	2.68
19	2.70
20	2.69
21	2.67
22	2.68
23	2.66
24	2.65

The rectifier *ReLu* function is used in the *hidden layer* and *linear activation function* is used at the *output layer*. The optimum architecture is taken as *24 units* in the *hidden layer* by trial and error, when the root mean square error (RMSE) decreased gradually and became stable, and the training and testing error produced satisfactory results.

The accuracy of the trained network is assessed in two ways: Firstly, the root mean square error (RMSE) and mean absolute error (MAE) are determined and compared, as well as the determination coefficient R2 and the explained variance EV.

The performance index for training of ANN is given in terms of root mean square error (RMSE). The obtained values of RMSE, MAE, R2 and EV are given in Table 4.7

Secondly, the predicted wind speed output values are compared with the measured values. The results are presented in Figure 4.9 which shows the relative accuracy of the predicted wind speed output for the optimal model which is one hidden layer with *24 units*. The overall percentage error obtained from the tested results is shown in table 4.8.

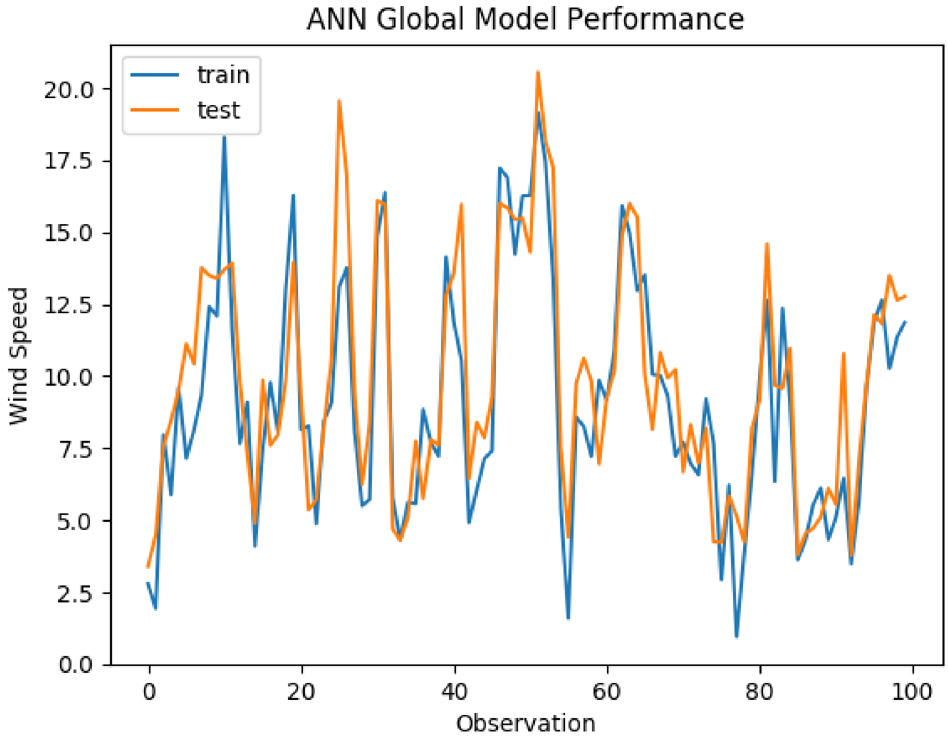


FIGURE 4.9: ANN Test Prediction

Figure 4.9 shows the training performance curve of neural network. The accuracy of the trained network is tested against available wind speed output data for 20% unseen observations.

TABLE 4.8: ANN test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Time	RMSE	MAE	R2	EV
0.02	2.67	2.04	0.68	0.69

The RMSE of the tested set obtained is 2.67. The RMSE and MAE are used together to diagnose the variations in the errors in a set of forecasts. The difference between RMSE and MAE is insignificant indicating the variance in the individual errors of the testing set is almost of the same magnitude.

Thus good, sufficient, and low values of error are obtained in testing the developed model for the prediction of wind speed output using neural networks. Figure 4.9 shows the magnitude of predicted wind speed.

4.4.2 Combined Models

4.4.2.1 Random Forest Model

We use *RandomForestRegressor* function from *Scikit-learn* which is a free software machine learning library for the Python programming language to verify the effectiveness of this method in this work.

The *RandomForestRegressor* function of *Scikit-learn* is used to establish the model of random forest regression algorithm as follows:

```
klearn.ensemble.RandomForestRegressor(n_estimators=10,
criterion='mse', max_depth=None, min_samples_split=2,
min_samples_leaf=1, min_weight_fraction_leaf=0.0,
max_features='auto', max_leaf_nodes=None,
min_impurity_decrease=0.0, min_impurity_split=None,
bootstrap=True, oob_score=False, n_jobs=1,
random_state=None, verbose=0, warm_start=False)
```

Among them, *n_estimators* represents the number of decision trees, and *max_features* represent the number of features used in each node of the tree for which we selected to work with the *sqrt* function. We decided to keep the other parameters as default since the criterion MSE is already suitable for our problem.

To run these experiments of parameters selection we performed k=10 fold cross-validation over 18.252 observations ranging from 23/11/1995 to 17/02/2013 that represents the 80% of the dataset. We Hold-Out 20% of the observation to later perform the testing phase on the unseen observations.

The main approach for the selection of parameters was trial and error and after some evaluation trial we came up with the best parameters.

We then present table 4.9 which contains the results for different number of trees for comparison purposes in order to select the best architecture for this global single model.

TABLE 4.9: RandomForest training CV results under different decision trees taking into account all 12 predictive variables for 1 point

N of trees	Score time	RMSE	MAE	R2	EV
10	0.027	2.76	2.12	0.63	0.63
50	0.13	2.63	2.01	0.66	0.66
100	0.28	2.62	2.01	0.66	0.67
200	0.56	2.61	2.00	0.66	0.67
500	1.44	2.60	1.99	0.66	0.67

By looking at the previous table we can see that the best performance is achieved by the model with 500 trees as number of estimators although the computation time is almost triple than using 200 trees as estimators, and since the difference is only 1 point we chose to work with 200 trees as configuration for our RF global model.

We then proceed to validate the model by performing training/validation with 80% of the dataset and then test with a held out of 20% of unseen observation.

We present a graphic which compares the original output versus the predicted ones for that specific model for Wind speed prediction.

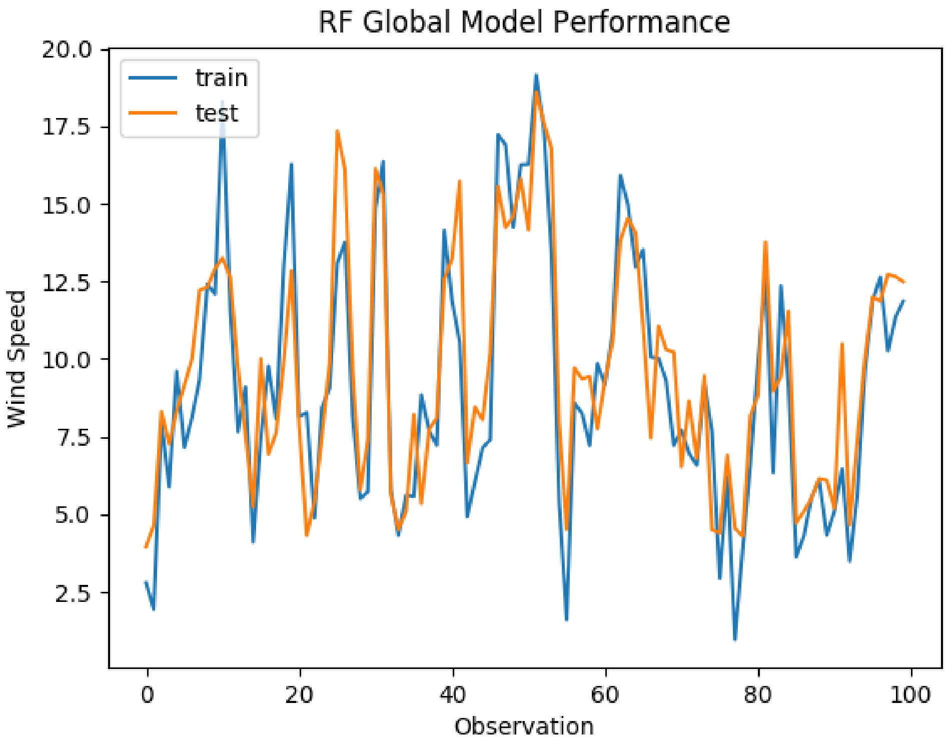


FIGURE 4.10: RandomForest Test Prediction

TABLE 4.10: RandomForest test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Time	RMSE	MAE	R2	EV
0.34	2.66	2.01	0.68	0.69

It can be seen from the figure that the predicted wind speed is close to the actual wind speed, so this method can be satisfactory applied to predict the Wind speed. As a whole, the random forest algorithm plays a very important role for this application. Figure 4.10 shows the random forest algorithm for wind power prediction is feasible, and the table 4.9 also lists the mean absolute error and root mean square error under different decision trees. This method has its certain superiority.

In table 4.9 MAE is short name for Mean Absolute Error and RMSE is the short name for Root Mean Square Error, R2 is short name for Determination Coefficient and EV is short name for Explained Variance, as presented in previous chapter.

4.4.2.2 Ensemble Model

To produce the results for the ensemble model we proceed to use the 3 previously trained global models and output for each of the 20% unseen observations the mean value of the 3 global models.

In the following table 4.11 and figure 4.11 we can check the performance of our combined ensemble model.

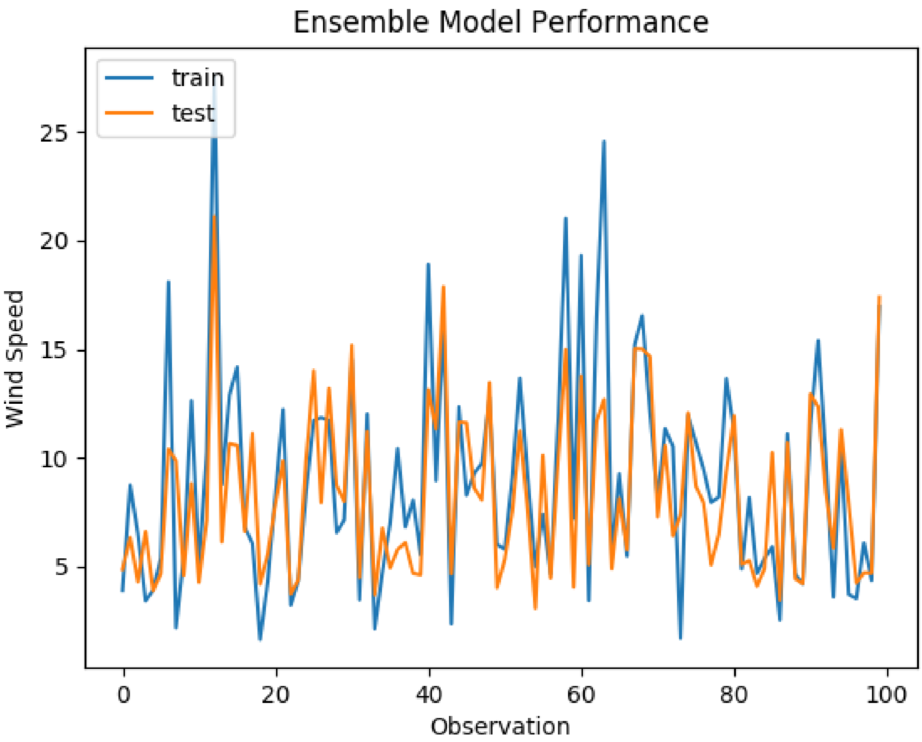


FIGURE 4.11: Ensemble Test Prediction

TABLE 4.11: Ensemble test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Time	RMSE	MAE	R2	EV
21.98	2.56	1.95	0.71	0.71

4.5 Local Models

In the following figure 4.12 we can see the proposed training for the local models.

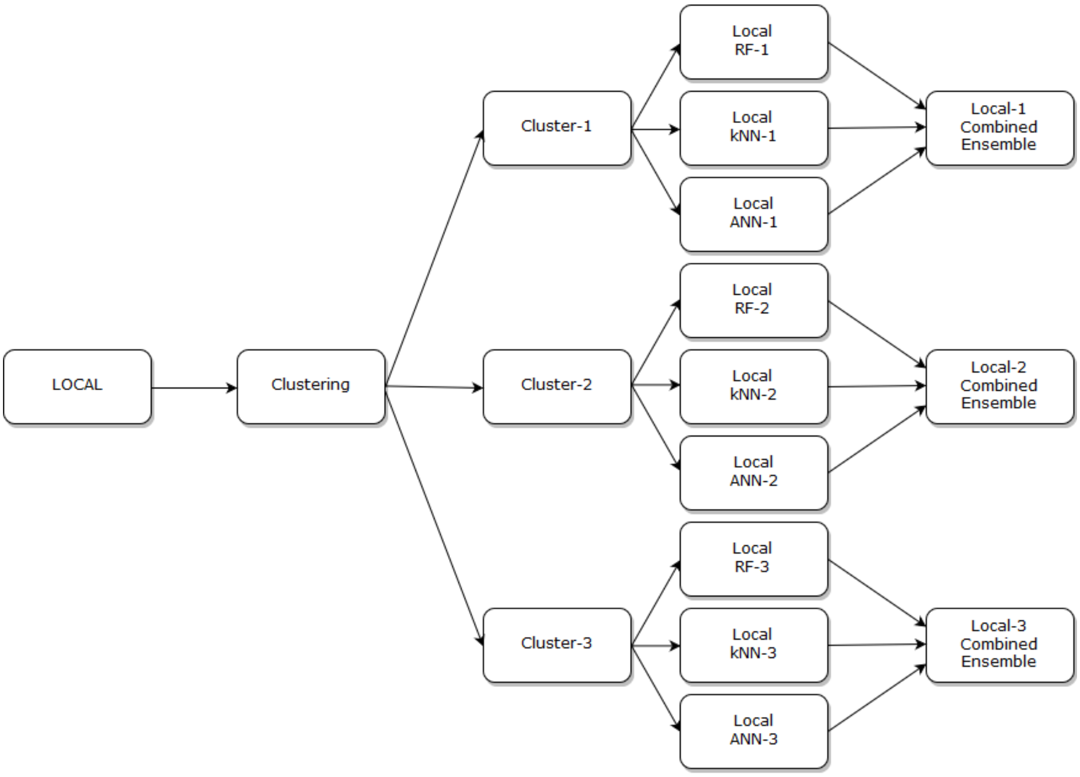


FIGURE 4.12: Local training process.

In the first step to create the local models we needed to obtain the *clusters*.

4.5.1 Clusters

In this section we explain how we obtained the *clusters* with our dataset in order train the *Local Models*, with the idea that if we use different clusters to train our local models these will become more specialized in observations belonging to each cluster.

The *KMeans* function of *Scikit-learn* is used to group the data into different *clusters*.

```
class sklearn.cluster.KMeans(n_clusters=8, init='k—means++',
n_init=10, max_iter=300, tol=0.0001,
precompute_distances='auto', verbose=0, random_state=None,
copy_x=True, n_jobs=1, algorithm='auto')
```

The main approach for the selection of the number of clusters was to test from 2 to 10 clusters and check the data to see if the interpretation of the partition made some sense comparing the value of the temperature and wind speed for each. Finally we concluded that for k=3 clusters the division of the data made the most sense grouping by low, medium and high temperature and having low speed, medium speed and high speed winds in each cluster. Notwithstanding, there was some overlapping among the elements of the three clusters, because they were not well separated. Probably this means that the clusters in this dataset were not clearly defined, and that there were not a radically different group observations from the others ones.

In the following figures 4.13 and 4.14 we can wee how the 3 clusters look like for the training and testing sets.

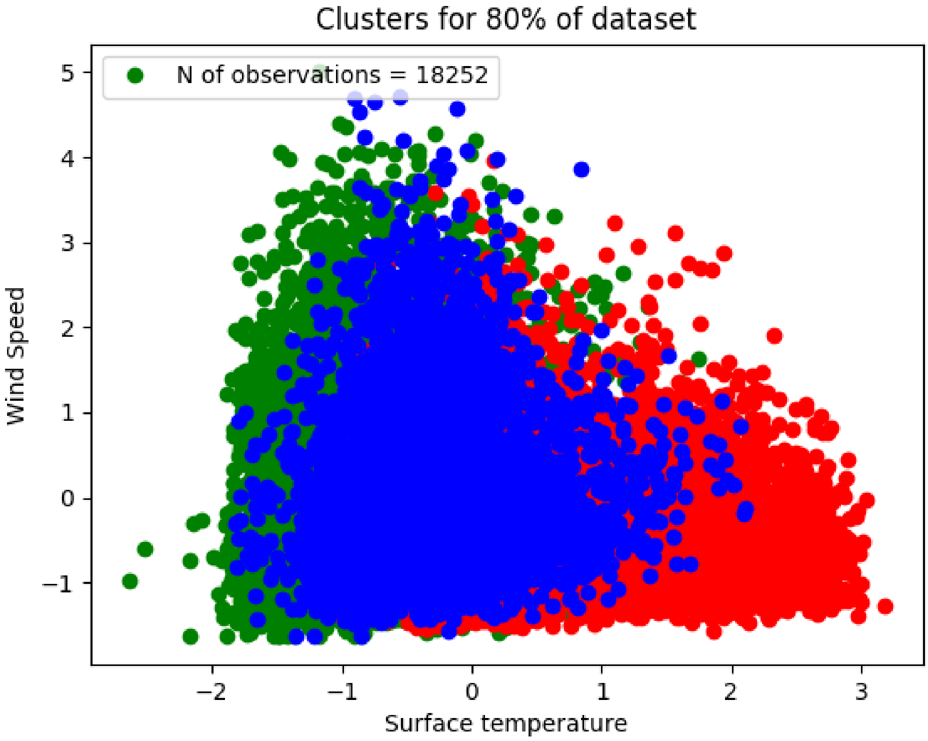


FIGURE 4.13: Train set clusters

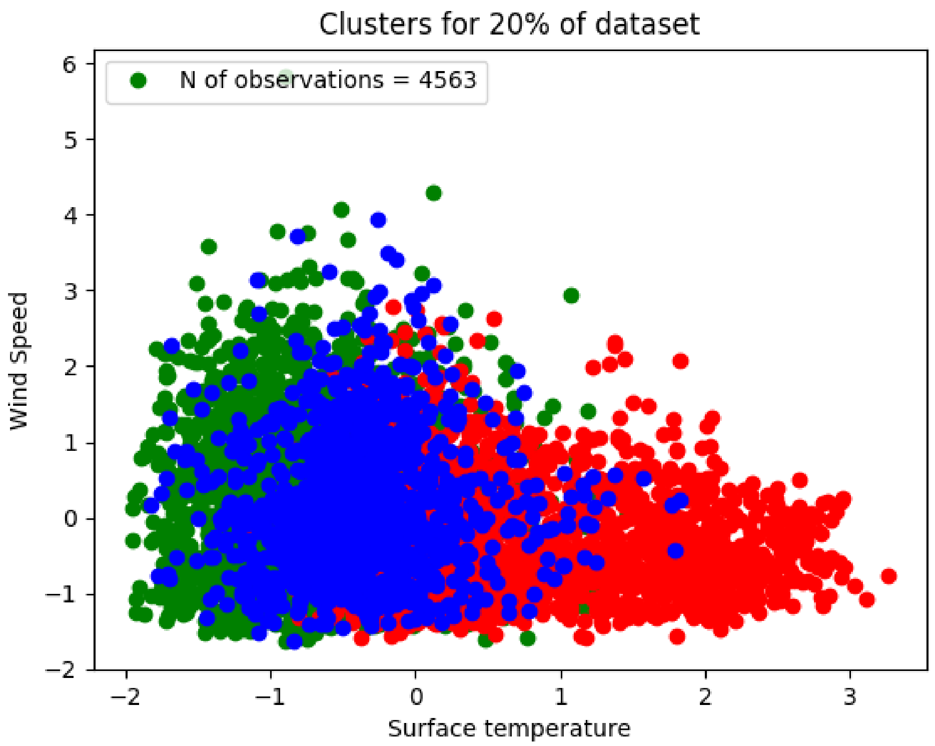


FIGURE 4.14: Test set clusters

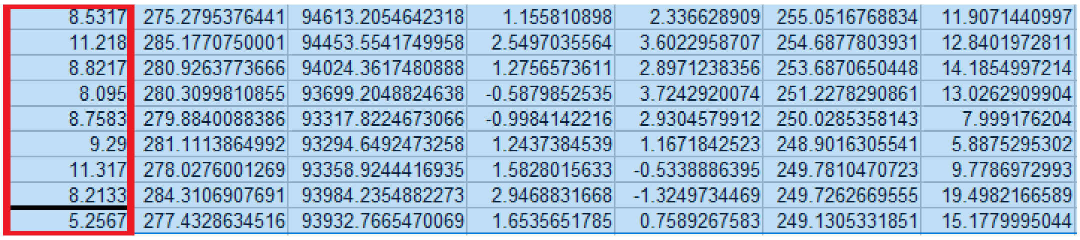
In order to maintain the same number of observations from each cluster in the testing set we proceed to first divide the whole data set into 3 clusters and then created the test set using 20% of each cluster.

In the following table 4.12 we can check the number of observations belonging to each cluster.

TABLE 4.12: Cluster partitions

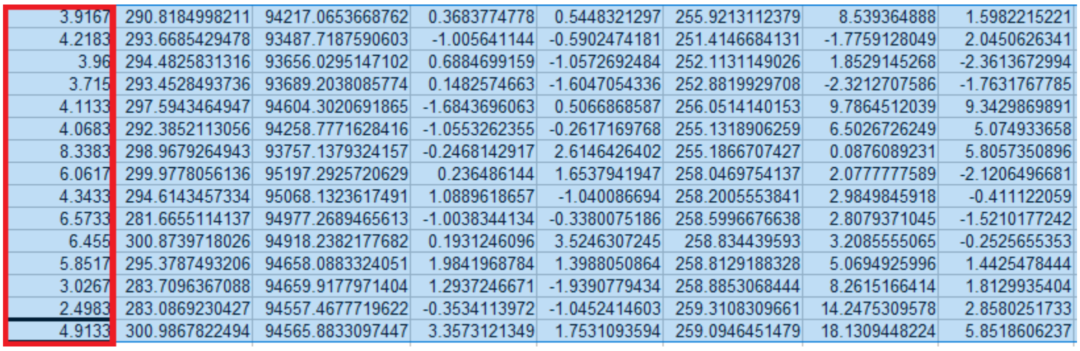
Partition	Total	Cluster-1	Cluster-2	Cluster-3
Total	22.815	5.844	8.562	8.409
Train	18.252	4.675	6.850	6.727
Test	4.563	1.169	1.712	1.682

By checking manually the data in each cluster we can see that the cluster-1 contains medium wind speed values ranging from 8 to 11 as shown in figure 4.15, then cluster-3 contains high wind speed values ranging from 9 to 17 as shown in figure 4.17 and finally cluster-2 contains low wind speed values ranging from 2 to 6 approx as shown in figure 4.16.



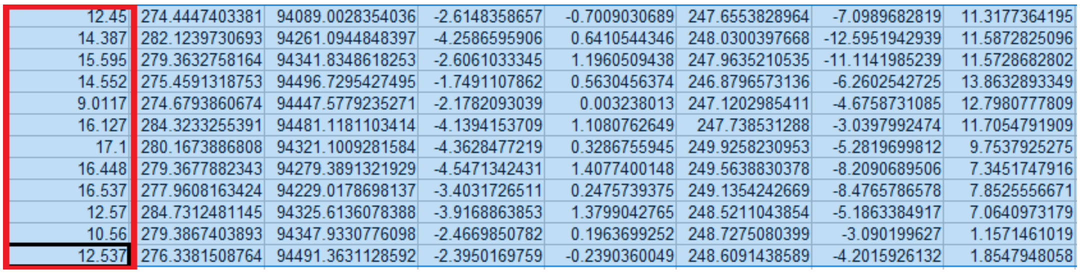
8.5317	275.2795376441	94613.2054642318	1.155810898	2.336628909	255.0516768834	11.9071440997
11.218	285.1770750001	94453.5541749958	2.5497035564	3.6022958707	254.6877803931	12.8401972811
8.8217	280.9263773666	94024.3617480888	1.2756573611	2.8971238356	253.6870650448	14.1854997214
8.095	280.3099810855	93699.2048824638	-0.5879852535	3.7242920074	251.2278290861	13.0262909904
8.7583	279.8840088386	93317.8224673066	-0.9984142216	2.9304579912	250.0285358143	7.999176204
9.29	281.1113864992	93294.6492473258	1.2437384539	1.1671842523	248.9016305541	5.8875295302
11.317	278.0276001269	93358.9244416935	1.5828015633	-0.5338886395	249.7810470723	9.7786972993
8.2133	284.3106907691	93984.2354882273	2.9468831668	-1.3249734469	249.7262669555	19.4982166589
5.2567	277.4328634516	93932.7665470069	1.6535651785	0.7589267583	249.1305331851	15.1779995044

FIGURE 4.15: Sample of Cluster-1 data



3.9167	290.8184998211	94217.0653668762	0.3683774778	0.5448321297	255.9213112379	8.539364888	1.5982215221
4.2183	293.6685429478	93487.7187590603	-1.005641144	-0.5902474181	251.4146684131	-1.7759128049	2.0450626341
3.96	294.4825831316	93656.0295147102	0.6884699159	-1.0572692484	252.1131149026	1.8529145268	-2.3613672994
3.715	293.4528493736	93689.2038085774	0.1482574663	-1.6047054336	252.8819929708	-2.3212707586	-1.7631767785
4.1133	297.5943464947	94604.3020691865	-1.6843696063	0.5066868587	256.0514140153	9.7864512039	9.3429869891
4.0683	292.3852113056	94258.7771628416	-1.0553262355	-0.2617169768	255.1318906259	6.5026726249	5.074933658
8.3383	298.9679264943	93757.1379324157	-0.2468142917	2.6146426402	255.1866707427	0.0876089231	5.8057350896
6.0617	299.9778056136	95197.2925720629	0.236486144	1.6537941947	258.0469754137	2.0777777589	-2.1206496681
4.3433	294.6143457334	95068.1323617491	1.0889618657	-1.040086694	258.2005553841	2.9849845918	-0.411122059
6.5733	281.6655114137	94977.2689465613	-1.0038344134	-0.3380075186	258.5996676638	2.8079371045	-1.5210177242
6.455	300.8739718026	94918.2382177682	0.1931246096	3.5246307245	258.834439593	3.2085555065	-0.2525655353
5.8517	295.3787493206	94658.0883324051	1.9841968784	1.3988050864	258.8129188328	5.0694925996	1.4425478444
3.0267	283.7096367088	94659.9177971404	1.2937246671	-1.9390779434	258.8853068444	8.2615166414	1.8129935404
2.4983	283.0869230427	94557.4677719622	-0.3534113972	-1.0452414603	259.3108309661	14.2475309578	2.8580251733
4.9133	300.9867822494	94565.8833097447	3.3573121349	1.7531093594	259.0946451479	18.1309448224	5.8518606237

FIGURE 4.16: Sample of Cluster-2 data



12.45	274.4447403381	94089.0028354036	-2.6148358657	-0.7009030689	247.6553828964	-7.0989682819	11.3177364195
14.387	282.1239730693	94261.0944848397	-4.2586595906	0.6410544346	248.0300397668	-12.5951942939	11.5872825096
15.595	279.3632758164	94341.8348618253	-2.6061033345	1.1960509438	247.9635210535	-11.1141985239	11.5728682802
14.552	275.4591318753	94496.7295427495	-1.7491107862	0.5630456374	246.8796573136	-6.2602542725	13.8632893349
9.0117	274.6793860674	94447.5779235271	-2.1782093039	0.003238013	247.1202985411	-4.6758731085	12.7980777809
16.127	284.3233255391	94481.1181103414	-4.1394153709	1.1080762649	247.738531288	-3.0397992474	11.7054791909
17.1	280.1673886808	94321.1009281584	-4.3628477219	0.3286755945	249.9258230953	-5.2819699812	9.7537925275
16.448	279.3677882343	94279.3891321929	-4.5471342431	1.4077400148	249.5638830378	-8.2090689506	7.3451747916
16.537	277.9608163424	94229.0178698137	-3.4031726511	0.2475739375	249.1354242669	-8.4765786578	7.8525556671
12.57	284.7312481145	94325.6136078388	-3.9168863853	1.3799042765	248.5211043854	-5.1863384917	7.0640973179
10.56	279.3867403893	94347.9330776098	-2.4669850782	0.1963699252	248.7275080399	-3.090199627	1.1571461019
12.537	276.3381508764	94491.3631128592	-2.3950169759	-0.2390360049	248.6091438589	-4.2015926132	1.8547948058

FIGURE 4.17: Sample of Cluster-3 data

Now that we have our data separated in 3 clusters we can then proceed to train our local models.

4.5.2 KNN Local models

In this section we show the results of a *K nearest neighbor* model for numerical prediction using for training and testing each cluster separately.

We maintain the same methodology for the creation of the *KNN global model* in creating the local model as well. We will use k=10 as the *n of neighbors*. We will also use 80% of each cluster as training set and the remaining 20% of the data as unseen observations for the final testing.

4.5.3 ANN Local models

In this section we show results of a Artificial neural network model for regression using for training and testing each cluster separately.

TABLE 4.13: Local KNN model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.17	2.73	2.08	0.62	0.62
Cluster-2	0.35	3.50	2.64	0.59	0.59
Cluster-3	0.32	2.38	1.81	0.56	0.56
Total-1-2-3	0.89	2.93	2.19	0.62	0.62

We maintain the same methodology for the creation of the ANN global model in creating the local model as well. We will use one hidden layer and 24 units as the architecture. We will also use 80% of each cluster as training set and the remaining 20% of the data as unseen observations for the final testing.

TABLE 4.14: Local ANN model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.04	2.92	2.29	0.56	0.60
Cluster-2	0.01	3.39	2.58	0.61	0.62
Cluster-3	0.00	2.85	2.27	0.36	0.48
Total-1-2-3	0.07	3.08	2.39	0.39	0.57

4.5.4 Random Forest Local models

In this section we show results of a *Random Forest* model for numerical prediction using for training and testing each cluster separately.

We maintain the same methodology for the creation of the *Random Forest Global model* in creating the *Local model* as well. We will use 200 as the n of trees estimators and the *sqrt* function as the max feature function. We will also use 80% of each cluster as training set and the remaining 20% of the data as unseen observations for the final testing.

TABLE 4.15: Local RF model test results for each cluster of 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.06	2.50	1.94	0.68	0.70
Cluster-2	0.11	3.22	2.41	0.65	0.66
Cluster-3	0.12	2.16	1.69	0.64	0.64
Total-1-2-3	0.3	2.68	1.42	0.68	0.69

4.5.5 Ensemble Local models

In this section we show results of an *Ensemble* model for numerical prediction using for training and testing each cluster separately.

We maintain the same methodology for the creation of the *Ensemble Global model* in creating the *Local model* as well. We will use the three previously trained local models using 80% of each cluster as training set and the remaining 20% of the data as unseen observations for the final testing.

TABLE 4.16: Local Ensemble model test results for each cluster of 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	**Time**	**RMSE**	**MAE**	**R2**	**EV**
Cluster-1	1.18	2.45	1.89	0.69	0.69
Cluster-2	1.36	3.19	2.41	0.66	0.66
Cluster-3	1.33	2.18	1.70	0.63	0.63
Total-1-2-3	3.87	2.66	2.01	0.68	0.68

Chapter 5

Discussion

In this chapter we will proceed to discuss the results obtained during the application of the defined methodology explained in chapter 3.

First we will discuss the different performance between the Global models by comparing the Single vs Combined proposed approaches. On a second step we will compare the proposed Local models approach to the Global models previously defined. Finally we compare all defined models to select the best one in terms of computational time and minimal error obtained.

5.1 Comparison of Single vs Combined Models

In this section we compare the defined models for the global Model methodology proposed, for which we selected two single numerical prediction models, which are the kNN and ANN ones, and two combined numerical prediction models which are the Random Forest and the ensemble model.

5.1.1 kNN

In order to be able to select the best k we based our experiments in trial and error and finally concluded that the best k parameter was 10.

In the first set of experiments we proceed to test a global KNN model, trained with the complete set of 80% of the data saved for training. In the following table 5.1 we can see that using the global model and testing it on the undivided 20% of unseen observations will return a RMSE of 3.61 not as good as the other tested predictive models.

TABLE 5.1: Global KNN model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
20%	2.37	3.61	2.77	0.41	0.42

5.1.2 ANN

We pursuit different approaches in order to select the best number of hidden layers and units and finally concluded to keep a very simple 1 hidden layer network and number of units using the N*2 approach which consist in multiplying the *N* number of variables inputs by 2, with this approach we got the best performance.

In the first set of experiments we proceed to test a global ANN model, trained with the complete set of 80% of the data saved for training. In the following table 5.2 we can see that using the global model and testing it on the undivided 20% of unseen observations will return a RMSE of 2.69 not as good as the other tested predictive models but better than the kNN model. We have to keep in mind that ANN really need large datasets to train and learn well the data representation. Maybe for this case this is a limitation.

TABLE 5.2: Global ANN model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
20%	0.02	2.69	2.04	0.68	0.68

5.1.3 Random Forest

In order to select the best parameters we based our testing in trial and error and finally concluded that the best performance was achieved by using 200 tree estimators.

In the following table 5.3 we can see that using the global model and testing it on the undivided 20% of unseen observations will return a RMSE of 2.65, which is a good value for this combined model.

TABLE 5.3: Global RF model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
20%	0.38	2.65	2.00	0.69	0.69

5.1.4 Ensemble

To produce the results for the ensemble model we proceed to use the 3 previously trained global models and output for each of the 20% unseen observations the mean value of the 3 global models.

In the following table 5.4 we can the see the model superiority achieving the best results for the selected metrics, but on the downside we can see the large computation time it take to output the predictive results.

TABLE 5.4: Ensemble test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Time	RMSE	MAE	R2	EV
21.98	2.56	1.95	0.71	0.71

5.1.5 Global comparison

Finally we can compare all the global models by side in shown table 5.5.

TABLE 5.5: Single vs Combined Models for 20% unseen data taking into account all 12 predictive variables for 1 point

Approach	Model	Time	RMSE	MAE	R2	EV
Single	kNN	2.37	3.61	2.77	0.41	0.42
	ANN	**0.02**	2.69	2.04	0.68	0.68
Combined	RF	0.38	2.65	2.00	0.69	0.69
	Ensemble	21.98	**2.56**	1.95	0.71	0.71

We can clearly see at table 5.5 that the Ensemble model has its superiority in terms of minimal RMSE predictive error since it gets 2.56 comparing to the next best one which is the Random Forest, which is also a combined method which obtains 2.65, but it has its down size when comparing the computation time. We can see that ensemble model takes much longer than the other methods to obtain the final output.

5.2 Comparison of Global vs Local Models

Based on the experiments results we can discuss the results separately for each of the models since they seem to be really dependent on the model.

We explained in the previous chapters that the first set of experiments consisted in creating global models, which we train with 80% of the data set and then tested with 20% of unseen observations.

Later we obtained the train data and test data into 3 clusters using the K-means algorithm and produced for each of the selected algorithms specialized models from which we performed tests using each cluster.

In following sections we will proceed to compare the results for each of the selected globals vs locals models.

5.2.1 k Nearest Neighbors

During the local modeling phase we tested the same kNN global model but on each of the clusters and tested with the 20% unseen data, for which we can see the results in following table 5.6.In there we can see that the cluster with more RMSE error is the second cluster which is the one with low-medium temperatures, which is also the same that gave the highest error for the RF global and local model, but accordingly to what is expected we get the same error running the global model over the complete 20% and running the global model over the 20% but separately in each cluster. The total RMSE error is 3.61.

Finally we proceed to train 3 different local KNN models each with one cluster derived from the 80% of data used for training of the global models. We used the same parameters of the global model for the definition of each of the local models. This means all use k=10 neighbors for the regression. In the following table 5.7 we can see the performance on each cluster has improved regarding when using the global model. In cluster-1 for instance we can see that the RMSE error goes from 2.77 to 2.73 when using the local model, for cluster-2 the RMSE goes from 4.22 to

TABLE 5.6: Global KNN model test results for each cluster of the 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.57	3.77	3.01	0.27	0.29
Cluster-2	0.75	4.22	3.23	0.40	0.44
Cluster-3	0.78	2.76	2.15	0.40	0.45
Total-1-2-3	2.10	3.61	2.77	0.41	0.42

3.50 and for cluster-3 the RMSE goes from 2.76 to 2.38 giving a total RMSE for the complete 20% of 2.93 when using Local models vs the already mentioned 3.61 giving when using the Global model.

So we can state that for the kNN using the more specialized local models really improves the performance of the prediction.

This result only applies to the K nearest neighbor model.

TABLE 5.7: Local KNN model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.17	2.73	2.08	0.62	0.62
Cluster-2	0.35	3.50	2.64	0.59	0.59
Cluster-3	0.32	2.38	1.81	0.56	0.56
Total-1-2-3	0.89	2.93	2.19	0.62	0.62

5.2.2 ANN

During the local modeling phase we tested the same ANN global model but on each of the clusters created with the 20% unseen data, for which we can see the results in following table 5.8.In the table we can see that the cluster with more RMSE error is the second cluster which is the one with low-medium temperatures, which is also the same that gave the highest error for the RF and KNN global and local models. Accordantly to what is expected we get the same error running the global model over the complete 20% and running the global model over the 20% but separately in each cluster, the total RMSE error is 2.69.

TABLE 5.8: Global ANN model test results for each cluster of the 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.00	2.57	2.00	0.66	0.68
Cluster-2	0.00	3.20	2.43	0.66	0.66
Cluster-3	0.00	2.18	1.68	0.63	0.63
Total-1-2-3	0.00	2.69	2.04	0.68	0.68

Finally we proceed to train 3 different Local ANN models each with one cluster derived from the 80% of data used for training of the global models. We used the same parameters of the global model for the definition of each of the local models. This means the same 1 layer with 24 units architecture for the prediction. In the following table 5.9 we can see the performance on each cluster has not improved the ones when using the global model. In cluster-1 for instance we can see that the RMSE error goes from 2.57 to 2.79 when using the local model, for cluster-2 the RMSE goes from 3.20 to 3.22 and for cluster-3 the RMSE goes from 2.18 to 2.36 giving a total RMSE for the complete 20% of 2.81 when using Local models vs the already mentioned 2.69 giving when using the Global model.

So we can say that for the ANN using the more specialized a.k.a local models does not improve the performance of the prediction and we can argue that in the case of using the cluster approach we are using much less data for the training which is a key when training neural networks.

TABLE 5.9: Local ANN model test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.04	2.79	2.16	0.60	0.61
Cluster-2	0.01	3.22	2.45	0.65	0.66
Cluster-3	0.00	2.36	1.86	0.57	0.59
Total-1-2-3	0.05	2.81	2.15	0.65	0.66

5.2.3 Random Forest

During the local modeling phase we tested the same RF global model but on each of the clusters obtained and tested with the 20% unseen data, for which we can see the results in following table 5.10. In this table 5.10 we can see that the cluster with more RMSE error is the second cluster which is the one with low-medium temperatures, but accordingly to what is expected we get the same error running the global model over the complete 20% and running the global model over the 20% but separately in each cluster. The total RMSE error is 2.65.

TABLE 5.10: Global RF model test results for each cluster of the 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.09	2.48	1.93	0.68	0.70
Cluster-2	0.13	3.18	2.40	0.66	0.67
Cluster-3	0.14	2.14	1.67	0.64	0.65
Total-1-2-3	0.36	2.65	2.00	0.69	0.69

Finally we proceed to train 3 different Local RF models each with one cluster derived from the 80% of data used for training of the global models. We used the same parameters of the global model for the definition of each of the local models. In the following table 5.11 we can see the performance on each cluster has not improved the ones when using the global model. In cluster-1 for instance we can see that the

RMSE error goes from 2.48 to 2.50 using the local model, for cluster-2 the RMSE goes from 3.18 to 3.22 and for cluster-3 the RMSE goes from 2.14 to 2.16 obtained a total RMSE for the complete 20% of 2.68 when using local models vs the already mentioned 2.65 giving when using the global model.

TABLE 5.11: Local RF model test results for each cluster of 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	0.06	2.50	1.94	0.68	0.70
Cluster-2	0.11	3.22	2.41	0.65	0.66
Cluster-3	0.12	2.16	1.69	0.64	0.64
Total-1-2-3	0.3	2.68	1.42	0.68	0.69

5.2.4 Ensemble

During the local modeling phase we tested an ensemble model for which we used the three previously defined local models (RF, kNN and ANN).We can see the results in following table 5.13. In this table we can see that the cluster with more RMSE error is the second cluster which is the one with low-medium temperatures, but contrary to what is expected we get the more error running the local ensemble model over the complete 20% than running the global ensemble model over the 20%. The total RMSE error for the global ensemble model is 2.56 as can be seen in table 5.12 and on the other hand the local ensemble model RMSE is 2.66 .

TABLE 5.12: Ensemble test results for 20% unseen data taking into account all 12 predictive variables for 1 point

Time	RMSE	MAE	R2	EV
21.98	2.56	1.95	0.71	0.71

TABLE 5.13: Local Ensemble model test results for each cluster of 20% unseen data taking into account all 12 predictive variables for 1 point

Partition	Time	RMSE	MAE	R2	EV
Cluster-1	1.18	2.45	1.89	0.69	0.69
Cluster-2	1.36	3.19	2.41	0.66	0.66
Cluster-3	1.33	2.18	1.70	0.63	0.63
Total-1-2-3	3.87	2.66	2.01	0.68	0.68

5.3 Comparison among all Models

In this final comparison we want to compare all trained numerical predictive models previously defined to finally be able to conclude which one achieved the best performance in terms of minimal predictive error obtained and minimal computational time used.

TABLE 5.14: Comparison Among all Models for 20% unseen data taking into account all 12 predictive variables for 1 point

Group	Approach	Model	Time	RMSE	MAE	R2	EV
Global	Single	kNN	2.37	3.61	2.77	0.41	0.42
		ANN	**0.02**	2.69	2.04	0.68	0.68
	Combined	RF	0.38	2.65	2.00	0.69	0.69
		Ensemble	21.98	**2.56**	1.95	0.71	0.71
Local		kNN	0.05	2.81	2.15	0.65	0.66
		ANN	0.89	2.93	2.19	0.62	0.62
		RF	0.3	2.68	1.42	0.68	0.69
		Ensemble	3.87	2.66	2.01	0.68	0.68

As we can see in table 5.14 we finally put all results together. And besides the improvement we obtained with the respect to the kNN global model when applied corresponding local kNN models, still the best predictive performance in terms of minimal predictive error was achieved by the Combined ensemble model, but in terms of computational time the best model was the global ANN model.

The combined models global RF model, ensemble of global models, and the RF local models have shown the best RMSE values (2.65, 2.56, 2.68). In addition, taking a look to the MAE values, again they show the lowest values, and specifically the Local RF model has the lowest MAE value. Given its low computational time cost is one of the best candidates to be used for the wind speed prediction.

Analyzing the local modeling approach, it can be seen as we have outlined that the three clusters were not clearly well defined, because the cluster-2 was formed by days which probably, due to its chronological time (season) should belong to the cluster with high temperatures, but its current temperatures were more similar to another weather season. This cluster-2 has been more difficult to predict for the local models. On the contrary, usually the other clusters (cluster-1 and cluster-3) have been predicted with a lower error by local models. Thus, it can be concluded, that the approach of using a local model for a well identified and defined group of observations is a good one, when effectively this group of observations is well defined.

Concretely, the kNN local models improve the error rates of the global kNN model. Probably, it is due to the local nature of the kNN method. Regarding the RF, probably it does not improve as a local models, because the local exploration of the observations it is inherent to the nature of the several decision trees included in the random forest combination model. Regarding the ANN local models probably does not improve the error rates of the global ANN model, because the ANN method requires a lot of data to be fully effective, and then, the local models, just for one part of the observations is constraining the predictive power of the ANN approach.

Furthermore, it is worth to outline that a global model could have a RMSE value lower than a combination of several local models. However, as the RMSE is an average value, probably is more effective to use the best local model known for each cluster, because the actual error could be lower in each cluster, and for the clusters where the local models are not so good (like the cluster-2 in our dataset) to use the global model.

5.4 Benchmark comparison

As we stated in previous chapter 2 in section 2.1.4 there exists a relation between wind speed prediction and generated wind power, so we believe we can compare our results with the ones obtain in [55] since their results were obtained using the same reanalysis data from the same wind farm in Spain.

We can see in following table 5.15 were EML (Extreme Machine Learning) and GPR (Gaussian Process) methods were used for wind power prediction on reanalysis data using the same 48 features as input, that they obtained 7.87 as RMSE error using the EML approach and 5.86 as RMSE using GPR approach.

TABLE 5.15: Comparative best results of the wind power estimation by the ELM and GPR. (Extracted from [55]).

Model	**RMSE**	**MAE**	**R2**
48 features-ELM	7.87	6.49	0.41
48 features-GPR	5.86	4.41	0.67

Comparing the results obtained in [55] as we showed in 5.15 to our results discussed in previous section in table 5.14 we can see that all our models RMSE are below half of their best error achieved except for the kNN global model which was the highest RMSE error obtained of 3.6 and all them were obtained using only a subset of the dataset as explained before 12 predictive variables from 1 point instead of all 4 points surrounding the wind farm.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this study, we showed that wind speed can be predicted from available reanalysis atmospheric data with error margin up to 2.6% using only 12 predictive variables of one point instead of all 48 mentioned inputs with a global model.

We explored several predictive methods and grouped them for further comparison in two main groups: Global and Local models.

In this work we have applied a data science methodology to derive several machine learning methods, and to explore its suitability for wind speed prediction. We have shown that using data selection techniques, like feature weighting, we can reduce the number of explicative variables from 48 to 12, reducing the complexity of the models, at the same time that nearly maintaining the predictive accuracy.

The local modeling approach has been explored in this application, comparing its performance against some global models. Moreover, several single models have been compared against some combined models, like a Random Forest approach or an ensemble of other models, with very good results.

In addition, it has been shown that the use of data-driven approaches to the short-term wind speed prediction is suitable, and they provide with good predictive models. In the literature, all models applied are usually physical models or purely statistical models. Thus, this work is pioneering the use of machine learning models for the wind speed prediction.

In the literature, we have found no models predicting the wind speed, but the wind power. Both variables are positively correlated because as much wind speed you have, more wind power is generated. Thus, comparing the RMSE error rates obtained in our work for wins speed prediction, and assuming that the wind power prediction would maintain the predictive rates, they outperform the RMSE rates found in the literature, which are around RMSE values of 5-6.

The developed models offers a reliable indication of the wind speed to the specific selected wind farm by using the input variables such as temperature, pressure and wind components.

The predicted wind speed output for the specific selected wind farm using the ensemble combined model shows the best performance among all tested predictive models. This model would be helpful for energy planners and the wind farm owners for future planning and execution.

6.2 Future work

For future work, we are planning a further study to verify the usefulness of reanalysis data as input for actual numerical predictions.

It is possible to propose several ideas of ways in which the models could be improved:

- Other model combinations, in this case we selected Random Forest and Ensemble as combined approaches, where the Ensemble was a Combination between RF, kNN and ANN, but other models could also be combined,
- In the feature selection we could also include Month, Hour and other seasonal information which we discarded during our preprocessing cleaning phase,
- In the cluster approach we could also create clusters for different seasons,
- We could add some genetic algorithm for the selection of the best parameter of each single model and then try combining these models.

Appendix A

Complementary Experiment results

A.1 Wind Speed Prediction using all 48F+M+H

In this experiments we were trying to identify the relevance that the variables of Month and hour had over the results, after several experiments since we noted that not improve was a result of using that information as input we decided to remove them form the dataset during the data preprocessing cleaning process.

A.1.1 RandomForest Results

To run this tests we used default parameters. We can see the results in table A.1.

TABLE A.1: RandomForest results using all 48F+M+H

	48 F	**48 F + M**	**48 F + H**	**48 F +M+H**
Fit time:	12.354	12.485	12.439	12.921
Train CV RMSE =	1.146	1.142	1.138	1.136
Train CV MAE =	0.815	0.812	0.807	0.806
Train CV R2 =	0.944	0.944	0.944	0.945
Train CV EA =	0.944	0.944	0.944	0.945
score time:	0.044	0.041	0.038	0.040
Test CV RMSE =	2.744	2.743	2.738	2.737
Test CV MAE =	2.077	2.075	2.068	2.069
Test CV R2 =	0.674	0.674	0.675	0.675
Test CV EA =	0.677	0.677	0.678	0.678

A.1.2 kNN Results

To run this tests we used default parameters. We can see the results in table A.2.

A.1.3 ANN Results

For the architecture of the ANN we used a simple one layer neural network with N*2 hidden units. We can see the results in table A.3.

A.2 Wind Speed Prediction with feature selection

In this experiments we were trying to identify the relevance that the variables based on the calculated weights had on the final wind speed prediction, after several experiments since we noted that no improve was a result of removing some variables

TABLE A.2: kNN results using all 48F+M+H

	48 F	48 F + M	48 F + H	48 F +M+H
Fit time:	0.165	0.147	0.177	0.156
Train CV RMSE =	3.000	3.003	3.012	3.014
Train CV MAE =	2.284	2.285	2.292	2.294
Train CV R2 =	0.614	0.613	0.611	0.610
Train CV EA =	0.621	0.620	0.618	0.618
score time:	0.919	0.944	1.040	1.050
Test CV RMSE =	3.778	3.783	3.788	3.790
Test CV MAE =	2.881	2.884	2.886	2.883
Test CV R2 =	0.385	0.384	0.382	0.382
Test CV EA =	0.399	0.398	0.397	0.397

TABLE A.3: ANN results using all 48F+M+H

	48 F	48 F + M	48 F + H	48 F +M+H
Fit time:	5.669	7.051	3.489	4.795
Train CV RMSE =	2.726	2.713	2.708	2.712
Train CV MAE =	2.067	2.055	2.048	2.052
Train CV R2 =	0.681	0.684	0.686	0.685
Train CV EA =	0.682	0.685	0.686	0.685
score time:	0.177	0.177	0.094	0.102
Test CV RMSE =	2.771	2.757	2.750	2.757
Test CV MAE =	2.101	2.089	2.079	2.085
Test CV R2 =	0.668	0.671	0.673	0.671
Test CV EA =	0.671	0.674	0.675	0.674

from the input we decided to keep the 4 variables of a single point and remove the other 3 point information form the dataset during the data preprocessing cleaning process.

In order to select the best 12, 22 and 28 best features out of the total 48 we selected from this table A.4 the first 12 which weight values are > 7.5, then the first 22 which weight values are above > 4.5 and finally the first 28 which weight values are above > 4.0, and combined them also with adding Hour and Month or removing hour and month.

A.2.1 RandomForest Results

To run this tests we used default parameters. We can see the results in table A.5.

A.2.2 kNN Results

To run this tests we used default parameters. We can see the results in table A.6.

TABLE A.4: Predictive variables weights calculated by UEB-1 algorithm

Name	Weight	Name	Weight	Name	Weight	Name	Weight
skt-4	10.00	up1-4	5.12	sp-4	4.45	vp2-1	3.19
skt-3	9.91	up1-3	4.99	sp-2	4.44	vp2-3	3.12
skt-2	9.58	up1-2	4.86	u10-1	4.33	vp2-2	3.10
skt-1	9.51	up1-1	4.75	u10-2	4.31	vp2-4	3.02
temp2-2	8.70	vp1-2	4.68	up2-2	3.91	wp2-2	1.69
temp2-4	8.66	vp1-1	4.65	up2-1	3.90	wp2-1	1.69
temp2-1	8.65	vp1-4	4.60	up2-3	3.81	wp2-3	1.61
temp2-3	8.63	vp1-3	4.57	up2-4	3.80	wp2-4	1.56
temp1-3	7.97	u10-3	4.56	v10-2	3.63	wp1-3	0.03
temp1-1	7.95	u10-4	4.55	v10-1	3.58	wp1-4	0.02
temp1-4	7.93	sp-3	4.48	v10-4	3.45	wp1-1	0.01
temp1-2	7.91	sp-1	4.46	v10-3	3.41	wp1-2	0.00

TABLE A.5: RandomForest results with feature selection

	12F	22F	28F	12F+M+H	22F+M+H	28F+M+H
Fit time:	4.205	7.172	9.794	4.927	9.306	12.557
Train CV RMSE =	1.962	1.715	1.681	1.940	1.682	1.654
Train CV MAE =	1.422	1.230	1.204	1.406	1.205	1.182
Train CV R2 =	0.835	0.874	0.879	0.839	0.879	0.883
Train CV EA =	0.835	0.874	0.879	0.839	0.879	0.883
score time:	0.046	0.048	0.054	0.045	0.070	0.070
Test CV RMSE =	4.680	4.131	4.045	4.675	4.058	3.981
Test CV MAE =	3.652	3.174	3.117	3.647	3.109	3.056
Test CV R2 =	0.057	0.265	0.295	0.060	0.291	0.318
Test CV EA =	0.060	0.268	0.298	0.063	0.293	0.321

A.2.3 ANN Results

For the architecture of the ANN we used a simple one layer neural network with N*2 hidden units. We can see the results in table A.7.

TABLE A.6: kNN results with feature selection

	12F	**22F**	**28F**	**12F+M+H**	**22F+M+H**	**28F+M+H**
Fit time:	0.066	0.071	0.109	0.067	0.116	0.197
Train CV RMSE =	3.972	3.428	3.329	3.928	3.404	3.337
Train CV MAE =	3.080	2.622	2.545	3.046	2.597	2.549
Train CV R2 =	0.323	0.496	0.525	0.338	0.503	0.522
Train CV EA =	0.323	0.496	0.530	0.339	0.503	0.528
score time:	0.260	0.884	0.642	0.441	1.951	1.067
Test CV RMSE =	4.888	4.302	4.181	4.846	4.278	4.178
Test CV MAE =	3.804	3.300	3.201	3.775	3.275	3.193
Test CV R2 =	-0.029	0.204	0.247	-0.011	0.212	0.249
Test CV EA =	-0.025	0.206	0.258	-0.008	0.214	0.260

TABLE A.7: ANN results with feature selection

	12F	**22F**	**28F**	**12F+M+H**	**22F+M+H**	**28F+M+H**
Fit time:	16.582	2.629	3.044	2.986	3.765	4.159
Train CV RMSE =	4.632	4.093	4.051	4.589	4.063	4.014
Train CV MAE =	3.640	3.185	3.155	3.606	3.160	3.124
Train CV R2 =	0.080	0.281	0.296	0.097	0.292	0.309
Train CV EA =	0.081	0.282	0.297	0.097	0.293	0.310
score time:	0.094	0.189	0.324	0.440	0.579	0.657
Test CV RMSE =	4.637	4.104	4.071	4.598	4.087	4.042
Test CV MAE =	3.646	3.194	3.174	3.613	3.180	3.148
Test CV R2 =	0.075	0.276	0.287	0.090	0.282	0.297
Test CV EA =	0.078	0.279	0.291	0.094	0.284	0.301

Bibliography

[1] S. S. Depuru-V. K. Devabhaktuni J. W. Lee E. Niko-laidis B. Andersen A. Afjeh Y. Kumar J. Ringenberg. "Wind energy: Trends and enabling technologies". In: *Renewable and Sustainable Energy Reviews* 53 (2016), pp. 209–224.

[2] Georgilakis PS. "A Technical challenges associated with the integration of wind power into power systems". In: *Renewable and Sustainable Energy Reviews* 12(3) (2008), pp. 852–63.

[3] Smith JC. "Utility wind integration and operating impact state of the art". In: *IEEE Trans Power Syst* 22(3) (2007), pp. 900–8.

[4] Soder L. "Experience from wind integration in some high penetration areas." In: *IEEE Trans Energy Convers* 22(1) (2007), pp. 4–12.

[5] Zuyi L. Jianhui W Shahidehpour M. "Security-constrained unit commitment with volatile wind power generation." In: *IEEE Trans Power Syst* 23 (2008), 1319–27.

[6] Pelgrum E Kling WL Brand AJ Ummels BC Gibescu M. "Impacts of wind power on thermal generation unit commitment and dispatch". In: *IEEE Trans Energy Convers* 22 (2007), 44–51.

[7] M. Uzunoglu YA. Tascikaraoglu. "A review of combined approaches for prediction of short-term wind speed and powe". In: *Renewable and Sustainable Energy Reviews* 34 (2014), pp. 243–254.

[8] Salama MMA. El-Fouly THM El-Saadany EF. "One day ahead prediction of wind speed and direction". In: *IEEE Trans Energy Convers* 23 (2008), pp. 191–201.

[9] Marvuglia A McKeogh EJ. Foley AM Leahy PG. "Current methods and advances in forecasting of wind power generation." In: *Renewable and Sustainable Energy Reviews* 37 (2012), pp. 1–8.

[10] Zheng S. Shi J Guo J. "Evaluation of hybrid forecasting approaches for wind speed and power generation time series". In: *Renewable and Sustainable Energy Reviews* 16 (2012), pp. 3471–80.

[11] Chen Z. laabjerg F. "Power electronics for modern wind turbines." In: *Synthesis Lectures on Power Electronics* (2006).

[12] Wikipedia contributors. *Wind power forecasting*. URL: https://en.wikipedia.org/w/index.php?title=Wind_power_forecasting&oldid=790098242.

[13] J. Zack. "Overview of wind energy generation forecasting." In: *TrueWind Solutions* (2003).

[14] Malik O Mandal P. Soman SS Zareipour H. "A review of wind power and wind speed forecasting methods with different time horizons." In: *Proceedings of the 2010 north american power symposium (NAPS)*. 2010, pp. 1–8.

[15] M. Zivkovic L. Lazic G. Pejanovic. "Wind forecasts for wind power generation using the Eta model." In: *Renew Energy* 35 (2010), pp. 1236–1243.

[16] Robert P.Broadwater Jaesung Jung. "Current status and future advances for wind speed and power forecasting." In: *Renewable and Sustainable Energy Reviews* 31 (2014), pp. 762–777.

[17] Siebert N Giebel G Barthelmie R. Kariniotakis G Pinson P. "The state of the art in short-term prediction of wind power – from an offshore perspective." In: *Sea Tech Week* (2004).

[18] U. Focken M. Lange. "Physical approach to short-term wind power prediction." In: *Springer* New York, NY (2006).

[19] L. Landberg. "Short-term prediction of local wind conditions." In: *J Wind Eng Ind Aerodyn* 89 (2001), pp. 235–245.

[20] Waldl H-P C. Focken U Lange M. "SPreviento – a wind power prediction system with an innovative upscaling algorithm." In: *Proceedings of the 2001 european wind energy conference and exhibition* (2001).

[21] Villanueva J Sanisidro MJ Loureiro Y Cantero E et al. Marti I Cabezon D. "LocalPred and RegioPred. Advanced tools for wind energy prediction in complex terrain." In: *Proceedings of the 2003 european wind energy conference and exhibition* (2003).

[22] C. McGowin. "California wind energy forecasting system development and testing. Phase 1: Initial testing." In: *Electric Power Research Institute (EPRI)* (2003).

[23] L. Landberg. "Short-term prediction of the power production from wind farms." In: *Wind Eng Ind Aerodyn* 80 (2003).

[24] J. Hardenberg L.A. Smith M.S. Roulston D.T. Kaplan. "Using medium-range weather forecasts to improve the value of wind energy production." In: *Renew Energy* 28 (2003), pp. 585–602.

[25] T. Evgeniou M. Hibon. "To combine or not to combine: selecting among forecasts and their combinations." In: *Int J Forecast.* 21 (2005), pp. 15–24.

[26] P. Mandal M.E. Kaye J. Meng L. Chang T. Senjyu M. Hibon T. EvgeniouA.U. Haque. "TA new strategy for predicting short-term wind speed using soft computing models." In: *Renew Sustain Energy* 16 (2012), pp. 4563–4573.

[27] X. Huang X. Wang P. Guo. "A review of wind power forecasting models." In: *Energy Procedia* 12 (2011), pp. 770–778.

[28] Fasullo John Shea Dennis Walsh John Dee Dick and National Center for Atmospheric Research Staff (Eds). *The Climate Data Guide: Atmospheric Reanalysis: Overview and Comparison Tables.* URL: https://climatedataguide.ucar.edu/climate-data/atmospheric-reanalysis-overview-comparison-tables.

[29] Dee DP Poli P Brugge R Fielding K Fuentes M Kållberg PW Kobayashi-S Uppala S Simmons A Berrisford P. "The ERA-Interim archive Version 2.0". In: (2011).

[30] Miquel Sànchez-Marrè Héctor Núñez. "Instance-Based Learning Techniques of Unsupervised Feature Weighting do not perform so badly". In: *Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004* (2004).

[31] H. Liu. M. Dash. "Proc. of SIGMOD Data Mining and Knowledge Discovery Workshop (DMKD)." In: (1999).

[32] M. P. Sebastian K. A. Abdul Nazeer. "Improving the Accuracy and Efficiency of the k-means Clustering Algorithm." In: *Proceedings of the World Congress on Engineering (WCE)* 1 (2009).

[33] P. A. Vela M. E. Celebi H. A. Kingrav. "A Comparative Study of Efficient Initialization Methods for the K-Means Clustering Algorithm." In: *Expert Systems with Applications* 40(1) (2013), pp. 200–210.

[34] Hardle. *Applied Nonparametric Regression*. Cambridge University Press, 1990.

[35] E. Mangalovaa E. Agafonovb. "Wind power forecasting using the k-nearest neighbors algorithm." In: *International Journal of Forecasting* 30 (2014), pp. 402–406.

[36] N. S. Altman. "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46(3) (1992), 175–185.

[37] Wikipedia contributors. *K-nearest neighbors algorithm*. URL: https://en.wikipedia.org/w/index.php?title=K-nearest_neighbors_algorithm&oldid=810910752.

[38] Hong. "GEFCom2014 probabilistic electric load forecasting: an integrated solution with forecast combination and residual simulation". In: *International Journal of Forecasting* 32 (2016), pp. 1012–1016.

[39] A. Seidel P. Seidel and O. Herbarth. "Multilayer perceptron tumour diagnosis based on chromatography analysis of urinary nucleosides." In: *Neural Netw* 20(5) (2007), 646–651.

[40] E. Alpaydin. *Introduction to Machine Learning, 2nd ed. Cambridge, MA, USA.* 2010.

[41] Junping Zhang Qi Wang Lisa Lamont Yujie Lin Uwe Kruger and Lana El Chaar. "Seasonal Analysis and Prediction of Wind Energy Using Random Forests and ARX Model Structures." In: *IEEE Transactions on Control Systems Technology* 23 (2015).

[42] Mabel E.Fernandez M. Carolin. "Analysis of wind power generation and prediction using ANN: A case study." In: *Renewable Energy* 33 (2008), pp. 986–992.

[43] Andreas Zell. *Simulation Neuronaler Netze [Simulation of Neural Networks] (in German) (1st ed.).* Addison-Wesley. ISBN 3-89319-554-8, 1994.

[44] Wikipedia contributors. *Artificial neural network*. URL: https://en.wikipedia.org/w/index.php?title=Artificial_neural_network&oldid=814418634.

[45] Xiaohui Li Zehong Zhou and Huaren Wu. "Wind Power Prediction based on Random Forests." In: *School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing, China* ().

[46] Fang Kuangnan. "Random forest combination forecast theory and its application in financial." In: *Xiamen university press* 5 (2012).

[47] ZHANG Ke. XU Ge. "Based on random forest model of real estate price evaluation." In: *Statistics and Decision* 17 (2014), pp. 22–25.

[48] B. Sevilla M. Sànchez-Marrè K. Gibert. "Evolving GESCONDA to an Intelligent Decision Support Tool." In: *5th International Congress on Environmental Modelling and Software (iEMSs'2010). iEMSs 2010 Proceedings* 3 (2010), pp. 2015–2024.

[49] Casella George Lehmann E. L. *Theory of Point Estimation (2nd ed.).* New York: Springer. ISBN 0-387-98502-6. MR 1639875., 1998.

[50] Torrie J. H. Steel R.G.D. *Principles and Procedures of Statistics with Special Reference to the Biological Sciences.* McGraw Hill, 1960.

[51] *Coefficient of Determination definition.* URL: Statrekwebsite.

[52] Smith H. raper N. R. *Applied Regression Analysis.* Wiley-Interscience. ISBN 0-471-17082-8., 1960.

[53] Joaquim Comas Ulises Cortés Manel Poch Miquel Sànchez-Marrè Ignasi R.-Roda. "Using Meta-cases to Improve Accuracy in Hierarchical Case Retrieval". In: *Computacion y sistemas* 4 (2000).

[54] S. Jiménez-Fernández J. Acevedo-Rodríguez L. Prieto L. Cornejo-Bueno L. Cuadra and S. Salcedo-Sanz. "Wind power ramp events prediction with hybrid machine learning regression techniques and reanalysis data." In: *Energies* 10(11) (2017), pp. 1–27.

[55] Adrian Aybar-Ruiz Luis Prieto Laura Cornejo-Bueno Carlos Camacho-Gomez and Sancho Salcedo-Sanz. "Feature selection with a Grouping Genetic Algorithm and Extreme Learning Machine Approach for Wind Power Prediction." In: *Advances in Artificial Intelligence. CAEPIA 2016. Lecture Notes in Computer Science* 9868 (2016).