



POLITECNICO DI TORINO

College of Computer Engineering, Cinema and Mechatronics

Master's Degree Thesis

**Wind Turbine Condition Monitoring  
Through Artificial Neural Networks  
Using SCADA Data**

**Supervisors**

prof. Bartolomeo MONTRUCCHIO  
dr. Antonio Costantino MARCEDDU  
dr. Edoardo GIUSTO

**Candidate**

Pier Paolo POLITI

DECEMBER 2022

*Ai miei genitori Antonio e  
Lucia*

# Summary

Wind power has been an important part of the energy market for the last decade. Anyway, the main problems associated with wind farms are operation and maintenance costs, which represent up to one-third of the total cost of energy production. The purpose of this thesis is to analyze the behavior of the turbines located on a wind farm in southern Italy to detect in real-time any malfunctions or alarms generated by the control system. Several Feedforward Neural Networks (FNN) and Recurrent Neural Networks (RNN) models have been created starting from data coming from Supervisory Control And Data Acquisition systems (SCADA). This allowed both to reproduce the characteristic behavior of the critical components of wind turbines and to develop a Statistical Process Control (SPC), which is useful for evaluating their anomalous behavior. This thesis is part of a collaboration between the Politecnico di Torino and the Turin-based company Sirius. While most similar works have relied on utilizing data from a single turbine to predict future failures on it, this approach attempts to use data from multiple turbines, of the same model and wind farm, to predict an alarm or abnormal behavior on any turbine located in the farm, even if newly installed.

A preliminary analysis on available wind farm information was performed to investigate both the number of available sensors and potential malfunctions, that may fall within the cases of interest.

The work can be divided into two phases:

- **Models Elaboration**, in which different models representing the state of health of wind turbines were developed using data from extended periods when turbines were not subjected to prolonged downtime.
- **Control**, where the developed models served as a reference. The ability of the system to detect the presence of anomalies was validated using data from new turbines representing a general steady-state operation. Only after this final phase, the models were considered ready to be used on real-time data.

**Model Elaboration phase:** The first step concerned data acquisition. Extraction and formatting of sensory data, averaged over 10 minutes, were then performed. The second step regarded data preprocessing. Through various data filtering and cleaning operations, it has been possible to isolate the data in which the turbines produce electricity, excluding both failures and operating limitations imposed by Terna. In addition, to represent the health status of the turbine, all outliers, responsible for the abnormal behaviors, were eliminated. The third step concerned model processing. Different Multilayer Perceptrons and Long Short-Term Memory (LSTMs) models were created and trained to represent the normal behavior of the wind turbines, which was used as a reference to identify unusual behaviors. The models can be divided into 4 categories based on the output variable:

- Wind Turbine Models - output: Active Power
- Gearbox Oil Models - output: Gearbox Oil Temperature
- Gearbox Bearing Models - output: Gearbox Bearing Temperature
- Generator Models - output: Generator Temperature

The model with the best performance were selected for each category. The fourth step regarded the validation. The output variable generated by each of the best models was compared with the actual value to evaluate the possible applicability of the system for condition monitoring.

**Control phase:** Real-time behavior was reproduced to achieve the monitoring and prevention goal. Data from additional turbines, of the same model and in the same wind farm, were acquired and preprocessed but without the elimination of the outliers, so the abnormal behaviors can be still represented. This data was processed by the selected models to calculate the ideal value of the output variable. Subsequently, an analysis of the deviation between the real value and the estimated one was carried out. The deviation is compared with selected control charts and their limits, focusing on the samples where the actual value differs significantly from the ideal. The purpose is to look for abnormal behavior preceding an alarm.

This solution tries to predict different types of output variables. With the aid of some known alarms, it is possible to demonstrate that the trained systems could anticipate alarms signaled by the control system, in particular the prediction of a general state of emergency of the turbines and alarms on the gearbox. It was also able to highlight other anomalous behaviors which are not reported by the control system.

# Contents

<b>List of Figures</b>	VI
<b>List of Tables</b>	X
<b>1 Introduction</b>	1
1.1 Overview	1
1.2 Scope of Thesis	1
1.3 Programming Environment	2
<b>2 General Notions</b>	3
2.1 Notions of wind energy and mechanics of wind turbines	3
2.1.1 Wind	3
2.1.2 Wind Turbine	3
2.1.3 Wind Energy	6
2.1.4 Power Curve	7
2.2 Notions of Statistic	8
2.2.1 Standard Deviation	8
2.2.2 Normal Distribution	9
2.2.3 Empirical Rule	9
2.2.4 Control Chart	10
2.2.5 Data Smoothing	11
2.2.6 Correlation Matrix	12
2.3 Time Series Forecasting Theory	13
2.3.1 Rolling Window	13
2.4 Machine Learning	14
2.4.1 General Introduction	14
2.4.2 Regression and Classification	15
2.4.3 Artificial Neural Network	15
2.4.4 Types of ANNs and Terminology	16
2.4.5 Recurrent Neural Network	17
2.4.6 Overfitting and Underfitting	19
2.4.7 Metrics	19
2.4.8 Hyperparameters	20
2.4.9 Partition of Database	21
2.4.10 Normalization	21

<b>3</b>	<b>Methodology</b>	22
3.1	Introduction to the Methodology	22
3.2	Models Elaboration	23
3.2.1	Data Acquisition and Preprocessing	23
3.2.2	Model Processing	24
3.2.3	Validation	24
3.3	Control phase	25
3.3.1	Preprocessing	25
3.3.2	Prediction of Output Variable	25
3.3.3	Testing	25
<b>4</b>	<b>Solution - Model Elaboration phase</b>	26
4.1	Gamesa G90	26
4.2	Data Acquisition	27
4.2.1	Data Formatting	29
4.3	Data Preprocessing	30
4.3.1	Data Cleaning	31
4.3.2	Downtime Filtering	31
4.3.3	Outliers Filtering	34
4.4	Wind Direction Analysis	38
4.5	Models Processing	40
4.5.1	Features Selection	41
4.5.2	Datasets	42
4.5.3	Artificial Neural Networks Architectures	44
4.5.4	Training and Test	47
4.6	Validation	61
4.6.1	Delta/Output Chart	61
4.6.2	Individual Chart	61
4.6.3	Results Validation	61
<b>5</b>	<b>Solution - Control phase</b>	70
5.1	Data Acquisition	70
5.2	Data Preprocessing	71
5.2.1	Data Cleaning	71
5.2.2	Downtime Filtering	71
5.3	Prediction of Variable Output	71
5.3.1	Wind Turbine Model	72
5.3.2	Gearbox Oil Model	73
5.3.3	Gearbox Bearing Model	74
5.3.4	Generator Model	74

5.4	Testing	75
5.4.1	Power Curve Outliers	75
5.4.2	Alternative Control Charts	76
5.5	Results	78
5.5.1	Wind Turbine Model	78
5.5.2	Gearbox Oil Model	84
5.5.3	Gearbox Bearing Model	87
5.5.4	Generator Model	89
<b>6</b>	<b>Conclusions</b>	<b>94</b>
6.1	Discussion	94
6.2	Real-time Prediction	94
6.3	Future Works	94
	<b>Bibliography</b>	<b>I</b>

# List of Figures

2.1	Horizontal axis WTG [17]. . . . .	4
2.2	Components of a WTG [17]. . . . .	4
2.3	Real nacelle of a WTG [20]. . . . .	6
2.4	Flow pattern around a wind turbine. . . . .	6
2.5	From left to right: blade pitch angle [19] and yaw [19] controls. . . . .	7
2.6	Ideal power curve [21]. . . . .	8
2.7	Example of STDs of two samples with the same mean [22]. . . . .	9
2.8	The Empirical Rule: for a normal distribution 99.7% of values are inside 3 STD of the mean [23]. . . . .	10
2.9	Example of individual chart with UCL and LCL limits [24]. . . . .	11
2.10	Example of correlation matrix [26]. . . . .	12
2.11	Difference between time series analysis and time series forecasting [29]. . . . .	13
2.12	Seasonal, trend and noise components of the time series [29]. . . . .	14
2.13	Rolling window method [29]. . . . .	14
2.14	The implementation of an artificial neuron [32]. . . . .	15
2.15	Example of an Artificial Neural Network [29]. . . . .	16
2.16	Multilayer Perceptron configuration [2]. . . . .	16
2.17	Example of Recurrent Neural Network [29]. . . . .	17
2.18	Unfolding operation performed on a RNN cell [35]. . . . .	18
2.19	LSTM Cell [37]. . . . .	19
2.20	Splitting operation of a dataset [44]. . . . .	21
3.1	This figure describes the methodology proposed in the thesis. In light blue are evidenced the operations performed for each proposed model [46] [47] [48] [49]. . .	23
3.2	Methodology of the system on real-time monitoring. . . . .	25
4.1	WTG 6: Power curve before filtering. . . . .	31
4.2	WTG 6: Power curve without limitation periods. . . . .	32
4.3	WTG 6: Power curve after filtering based on T-Status. . . . .	33
4.4	WTG 6: Power curve without negative or null power in output. . . . .	34
4.5	WTG 6: Power curve without unrealistic low power. . . . .	34
4.6	WTG 6: Result of Data smoothing filter on the Power Curve. In orange are represented the samples eliminated. . . . .	35

4.7	WTG 6: Power curve after data smoothing operation. . . . .	35
4.8	WTG 6: Correlation matrix of the dataset before filtering. . . . .	36
4.9	WTG 6: Correlation matrix of the dataset after filtering operation. . . . .	37
4.10	WTG 6, from left to right: 3D power curve before and after filtering. . . . .	38
4.11	WTG 2 and WTG 7: 3D power curve after filtering. . . . .	39
4.12	Compass rose [28]. . . . .	39
4.13	Comparison of the power curves of the WTG 6 and WTG 7 considering only the wind direction ENE and NNE. . . . .	40
4.14	The 2 different approach (L1 and L2) for creating the rolling window. . . . .	43
4.15	Phases of creation of training, validation, and test data sets. . . . .	43
4.16	Active power output variable for WTG 6 and division of timestamp values in train- ing (green), validation (blue), and test dataset (orange). . . . .	44
4.17	Active power output variable for WTG 6 and division of timestamps value in the training set (green), validation set (blue) and test set (orange) for timeseries fore- casting approach. . . . .	44
4.18	ANN Model of MLP Architecture with an input layer with 5 neurons and 1 hidden layer of 50 neurons. . . . .	46
4.19	ANN Model of MLP Architecture with an input layer with 5 neurons and 2 hidden layers of 50 and 18 neurons. . . . .	46
4.20	ANN Model of LSTM Architecture with input shape [5, 13]. . . . .	47
4.21	ANN Model of 4 Stacked LSTM Architecture with input shape [5, 13]. . . . .	48
4.22	Loss of the best network for the wind turbine model. . . . .	49
4.23	RMSE metric of the best ANN of the wind turbine model. . . . .	49
4.24	$R^2$ metric of the best ANN of the wind turbine model. . . . .	50
4.25	Real and predicted value of active power for the WTG 6 test set. . . . .	50
4.26	Power curves real and predicted of the test set of the WTG 6 turbine. . . . .	51
4.27	Power curves 3D (with Wind Direction) real and predicted of the test set of the WTG 6 turbine. . . . .	51
4.28	Loss of the best network for the gearbox oil model. . . . .	52
4.29	RMSE metric of the best ANN of the gearbox oil model. . . . .	53
4.30	$R^2$ metric of the best ANN of the gearbox oil model. . . . .	53
4.31	Real and predicted value of gearbox oil temperature for the WTG 6 test set. . . . .	54
4.32	Loss of the best network for the gearbox bearing model. . . . .	55
4.33	RMSE metric of the best ANN of the gearbox bearing model. . . . .	56
4.34	$R^2$ metric of the best ANN of the gearbox bearing model. . . . .	56
4.35	Real and predicted value of gearbox bearing temperature for WTG 6 test set. . . . .	57
4.36	Loss of the best network for the generator model. . . . .	58
4.37	RMSE metric of the best ANN of the generator model. . . . .	59
4.38	$R^2$ metric of the best ANN of the generator model. . . . .	59
4.39	Real and predicted value of generator temperature for WTG 6 test set. . . . .	60
4.40	Wind turbine model - STD of the delta values of the 3 test set and limit LCL and UCL. . . . .	62

4.41	Wind turbine model - Delta values compared with output active power. . . . .	62
4.42	Wind turbine model - Shewhart control chart on the WTG 6 test set. . . . .	63
4.43	Gearbox oil model - STD of the delta values of the 3 test set and limit LCL and UCL. . . . .	64
4.44	Gearbox oil model - Delta values compared with gearbox oil temperature. . . . .	65
4.45	Gearbox oil model - Shewhart control chart on the WTG 6 test set. . . . .	65
4.46	Gearbox bearing model - STD of the delta values of the 3 test set and limit LCL and UCL. . . . .	66
4.47	Gearbox bearing model - Delta values compared with gearbox temperature value. . . . .	66
4.48	Gearbox bearing model - Shewhart control chart on the WTG 6 test set. . . . .	67
4.49	Generator model - STD of the delta values of the 3 test set and limit LCL and UCL. . . . .	68
4.50	Generator model - Delta values compared with generator temperature value. . . . .	68
4.51	Generator model - Shewhart control chart on the WTG 6 test set. . . . .	69
5.1	Real and predicted values of active Power of WTG 1. . . . .	72
5.2	3D power curves with the actual value and predicted value for WTG1. . . . .	73
5.3	Real and predicted values of gearbox oil temperature of WTG 1. . . . .	74
5.4	Real and predicted values of gearbox bearing temperature of WTG 1. . . . .	74
5.5	Real and predicted values of generator temperature of WTG 1. . . . .	75
5.6	Power curves outliers for WTG1, with real (blue) and predicted (orange) values for good predictions, and real (black) and predicted (red) values for outliers. . . . .	76
5.7	Principle of creating the control chart "Sum of $\Delta_{th}$ " for positive outliers: top figure shows the $\Delta$ values, while the bottom one shows the possess of sum of $\Delta_{th}$ values. In the last three samples, the value of $1.5 \times UCL$ is exceeded and alarms are generated. . . . .	77
5.8	Principle of creating the control chart "Sum of events" for positive outliers: top figure shows the $\Delta$ values, while the bottom one shows the possess of the sum of numbers of outliers. In the last three samples, the value of 5 consecutive outliers is exceeded and alarms are generated. . . . .	78
5.9	Control chart of $\Delta$ active power for WTG 1. . . . .	79
5.10	Control chart average of $\Delta$ active power for WTG 1. . . . .	79
5.11	Sum of $\Delta_{th}$ control chart for negative $\Delta$ values calculated for active power of WTG 1. . . . .	80
5.12	Control chart of $\Delta$ active power for WTG 5. . . . .	81
5.13	Control chart average of $\Delta$ active power for WTG 5. . . . .	82
5.14	Power curves outliers Sum of $\Delta_{th}$ control chart for negative $\Delta$ values calculated for active power of WTG 5. . . . .	82
5.15	Power curves outliers for WTG5, with real (blue) and predicted (orange) values for good predictions, and real (black) and predicted (red) values for outliers. . . . .	83
5.16	Delta/Output chart of $\Delta$ active power for each output active power of WTG 5. . . . .	83
5.17	Control chart of $\Delta$ gearbox oil temperature for WTG 5. . . . .	85
5.18	Sum of $\Delta_{th}$ control chart for positive $\Delta$ values calculated for gearbox oil temperature of WTG 5. . . . .	85
5.19	Sum of events control chart for positive $\Delta$ values calculated for gearbox oil temperature of WTG 5. . . . .	86

5.20	Delta/Output chart of $\Delta$ gearbox oil temperature for each output gearbox oil temperature of WTG 5. . . . .	86
5.21	Control chart of $\Delta$ gearbox bearing temperature for WTG 4. . . . .	87
5.22	Sum of $\Delta_{th}$ control chart for positive $\Delta$ values calculated for gearbox bearing temperature for WTG 4. . . . .	88
5.23	Sum of events control chart for positive $\Delta$ values calculated for gearbox bearing temperature of WTG 4. . . . .	88
5.24	Delta/Output chart of $\Delta$ gearbox bearing temperature for each output gearbox bearing temperature of WTG 4. . . . .	89
5.25	Control chart of $\Delta$ generator temperature for WTG 3. . . . .	90
5.26	Sum of $\Delta_{th}$ control chart for positive $\Delta$ values calculated for generator temperature for WTG 3. . . . .	90
5.27	Sum of $\Delta_{th}$ control chart for negative $\Delta$ values calculated for generator temperature for WTG 3 . . . . .	91
5.28	Control chart of $\Delta$ generator temperature for WTG 5. . . . .	92
5.29	Sum of $\Delta_{th}$ control chart for positive $\Delta$ values calculated for generator temperature for WTG 5. . . . .	93
5.30	Delta/Output chart of $\Delta$ generator temperature for each output generator temperature of WTG 5. . . . .	93

# List of Tables

4.1	WTGs used in model elaboration phase (for ANNs training) and relative data periods used. . . . .	28
4.2	WTG 2: Original data set. . . . .	29
4.3	WTG 2: Data set after the cleaning operations. . . . .	30
4.4	WTG 6: Number of sample and null values for each filtering operation. . . . .	31
4.5	Restriction periods. . . . .	32
4.6	Status Filtering: Operation of filtering based on status code value. All last 4h samples before an alarm, fault, or error are eliminated. The row with the red background represents the samples affected by an alarm or error, instead, the row with font red represents the samples eliminated. . . . .	33
4.7	Inputs and outputs used for the different models, divided for categories. . . . .	41
4.8	Number of samples in training, validation, and test sets for each turbine, for both the two approaches of prediction. . . . .	45
4.9	Metrics evaluation on the 3 different test sets for the wind turbine model. . . . .	50
4.10	Metrics evaluation on the 3 different test sets for the gearbox oil model. . . . .	53
4.11	Metrics evaluation on the 3 different test sets for the gearbox bearing model. . . . .	56
4.12	Metrics evaluation on the 3 different test sets for the generator model. . . . .	59
5.1	WTGs used in Control phase and relative data periods used. . . . .	70

# Chapter 1

## Introduction

In this chapter a brief overview of the thesis will be provided, followed by some notions on the involved technologies.

### 1.1 Overview

The growing world energy demand and international agreements, aimed at reduction of the impact of climate change, have led in recent years to the rapid development of renewable energy sources. As evidenced by the substantial increase in installed capacity among renewable, a role of fundamental importance is covered by wind energy.

Horizontal axis wind turbines are the devices most used for the production of electricity from large-scale wind sources and were subject to a sudden spread over the last few years. The development of technological innovation and the increase in the average power per single turbine made it of crucial importance to the optimization of O&M strategies to anticipate possible failures and avoid prolonged maintenance times. Today a high availability of power is required and long downtime of the wind turbine can lead to unacceptable production losses.

Today a high availability of power is required and long downtime of the wind turbine can lead to unacceptable production losses. SCADA is the acronym for Supervisory Control and Data Acquisition. This type of system is the most widely used for monitoring and preventing failures. They are now installed on every wind turbine and provide a large amount of data with high frequency, often associated with a saving of the average value every 10 minutes

### 1.2 Scope of Thesis

The scope of this thesis is to develop several Artificial Neural Networks (ANN) models to simulate the behavior of wind turbines and some of the main components to detect real-time malfunctions.

To verify its effectiveness and applicability, the models are tested on real wind turbines located in Southern Italy. This thesis is part of a collaboration between the Politecnico di Torino and the Turin-based company Sirius [1], which is active in the field of remote electrical control systems since its establishment in the year 2000 within the technology spinoff of Politecnico di Torino. It has based its strength on research and development of cutting-edge solutions in the electrotechnical field. Today, it collaborates with major manufacturers in the electrotechnical sector by providing engineering support and products with the highest technological content. With the help of their VireoXPower [15] software, it was possible to extract not only a considerable amount of data from the various sensors mounted on the turbines but also the list of alarms and limitations that affected the wind farm. The thesis work was divided into several phases:

- Data acquisition. A preliminary analysis of the information available for a wind farm was carried out to analyze both the number of sensors available and the possible malfunctions that may fall within the cases of interest. An extraction and formatting of sensory data averaged over 10 minutes were then performed.
- Preprocessing. Through various data filtering and cleaning operations, it was possible to isolate the data in which the turbines did not show failures and had no operating limitations or anomalous behavior.
- Model processing. Different Multilayer Perceptron [2] and Long Short-Term Memory (LSTMs) [36] models were created and trained to represent the normal behavior of the wind turbines which was used as a reference to identify unusual behaviors. The models with the best performance were selected for each output variable.
- Analysis of the results. The output variable generated by each of the best models was compared with the actual value to evaluate the possible applicability of the system for condition monitoring. Real-time behavior was reproduced to achieve the monitoring and prevention goal. Unfiltered data from additional turbines of the same model and in the same wind farm were used to calculate the ideal behavior of the output of interest. Subsequently, an analysis of the deviation between the real value and the estimated one was carried out. The deviation is compared with the selected control charts and their limits, focusing on where the actual behavior differed significantly from the ideal.

This solution tries to predict different types of output variables. To simulate the power production, only information on environmental factors was initially used, and then information on the internal sensors was used to estimate the temperature of the gearbox component and the generator. With the aid of some known alarms, it is possible to demonstrate that the trained systems could detect failures and anticipate alarms signaled by the control system, in particular the prediction of a general state of emergency of the turbines and faults on the gearbox.

### 1.3 Programming Environment

This thesis was developed using Python [3] as the programming language, specifically version 3.8, and Spyder [6] as the Integrated Development Environment (IDE). For writing the code necessary to train and test the ANNs, Sirius has made available a workstation capable of offering considerable computing power thanks to its NVIDIA RTX 3060 Graphics Processing Unit (GPU) [5]. Remote connection to the server was accomplished through the use of an internal Sirius VPN using the Fortinet [4] tool. A suitable programming environment was then installed on the server, an operation made easier by using the Anaconda distribution [10], which allows a more easy installation of all the necessary packages. The code needed to train ANNs has been written with the Keras [7] library, a high-level neural networks API written in Python and able to run on Tensorflow [11], a framework for machine learning. The following libraries were also used:

- Matplotlib [8], a 2D plotting library for Python.
- Seaborn [14], a visualization library based on Matplotlib.
- NumPy [9], a BSD-licensed Python package used for scientific computing.
- Pandas [13], an open-source BSD-licensed Python library that provides high-performance data structures and data analysis.
- Scikit-learn [12], an open-source, BSD-licensed Python library for data analysis.

## Chapter 2

# General Notions

### 2.1 Notions of wind energy and mechanics of wind turbines

Surface heating at different points on the planet produces different effects, which cause atmospheric movements. The movement of air caused by solar energy is known as wind.

Wind turbines, also known as Wind Turbine Generators (WTGs) are devices that convert the kinetic energy of the wind into electricity.

#### 2.1.1 Wind

The inhomogeneity of solar irradiance on the Earth's surface causes variations in atmospheric pressure across the planet, so high-pressure conditions occur in warm areas and low-pressure conditions occur in cooler areas. These large-scale differences cause a compensatory movement of air which is generally called wind.

Considered on a large scale, it is possible to witness a circulation of air masses at different latitudes that are cyclically affected by the seasons; on a smaller scale, however, there is a difference in warming between land and water masses, resulting in daily land and sea breezes. The profile and irregularities of the land or water surface have a significant impact on the wind and its local characteristics. The wind blows with greater speed over wide and flat surfaces such as the sea, and this is the main element of interest for coastal or marine wind farms. The wind strengthens on the tops of plateaus or in valleys oriented parallel to the dominant wind direction, while it slows down on irregular surfaces, such as cities or forests.

When managing energy produced by the wind it is critical to consider the wide variations in velocity between different locations: regarding the installation of wind turbines, sites only a few kilometers away from each other may be subject to different conditions and have different interests.

#### 2.1.2 Wind Turbine

As specified above, a wind turbine is a machine that converts a moving mass of air into mechanical energy, which is then transformed into electricity. The first conversion is carried out by the rotor blades, while the second is done by an electric generator. The energy captured by the wind blades is transferred to a generator, which converts the rotational energy into electrical energy and transfers it to a transformer usually located at the foot of the wind tower. The transformer is responsible for converting the electricity to high voltage and then feeding it into the electrical transmission grid. A full scheme of a horizontal axis WTG is shown in Figure 2.1.

A wind turbine consists of the following parts, which are also schematized in Figure 2.2:

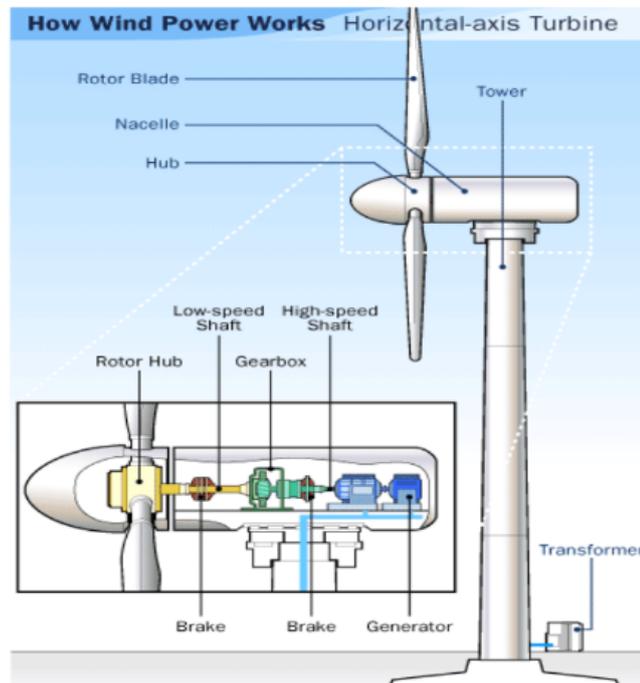


Figure 2.1. Horizontal axis WTG [17].

- Tower: It is the supporting structure of the wind turbine. It is made of a steel tube or cement, both of which can absorb the vibrations generated by the rotating motion of the blades. The height of the tower is determined by the location of the wind farm and the local wind characteristics. On land, the nacelle is placed at a height of 1 or 1.2 times the rotor diameter. In areas with low winds, the nacelle is elevated to expose it to stronger winds. At sea, it can be placed lower, typically at a height of 0.8 times the rotor diameter. The towers have a conical shape, with the base with a larger diameter than the top. It is made up of several bolted-together sections. Tubular towers have the advantage of protecting the instrumentation inside them and are much safer and easier to maintain than truss ones.

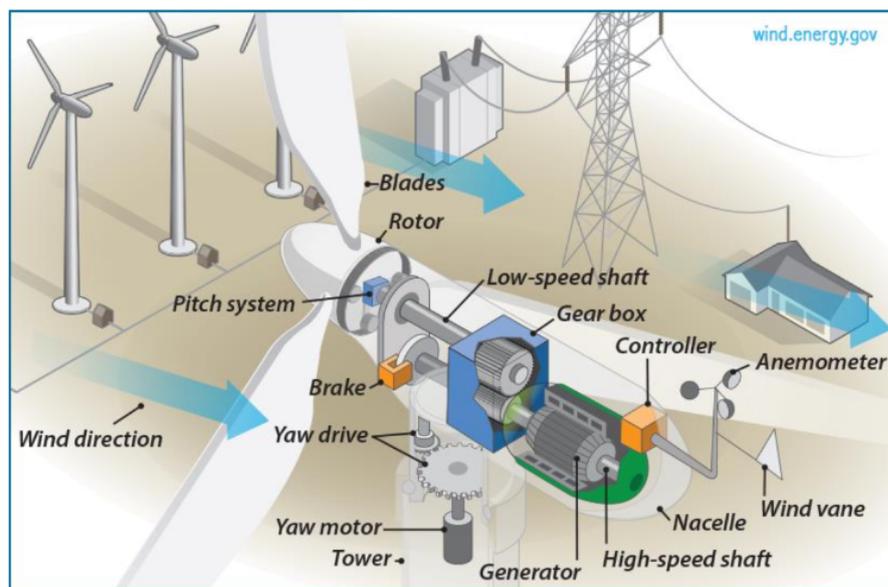


Figure 2.2. Components of a WTG [17].

The turbine is accessed via a staircase inside the tower. Because the wind can blow from any direction, the towers are conical for symmetry. They must also be capable to withstand the elements for years. The foundations allow the towers to be anchored to the ground.

- **Blades:** they are the elements that interact with the wind, and their shape is designed to achieve good aerodynamic efficiency. They are made of composite material and shaped like the wing of an airplane, they are light and strong. When they are hit by the wind, they rise and begin to rotate, causing the rotor of the rotor. The more powerful they are, the larger they are. In addition, the blade can rotate up to 90 degrees around its axis.
- **Rotor:** It consists of a set of rotating blades, the hub, the shaft, and the pitch control mechanism, which is a system whose function is to adjust the position and inclination of the blades with respect to the wind and to stop the rotation of the rotor if the wind is too strong or too weak. The rotor shaft rotates at a relatively slow speed of 10 - 40 revolutions per minute. The dimensions and the weight of the electric generator are roughly proportional to its rotational speed. It is critical to design generators with a high rotation speed (e.g., 1,000 or 1,500 rpm per minute) and to use an intermediate gearbox to convert the slow rotation of the shaft into the high rotation speed of the generator.
- **Nacelle:** It is placed on top of the tower and hosts all the components of the WTG except for the rotor and the blades. A picture of a real nacelle with its inside visible is depicted in Figure 2.3. A speed sensor on the back of the turbine constantly measures the wind direction. When the wind direction changes, the sensor sends a signal to the yaw control drives, which cause the entire nacelle, and thus the turbine rotor and blades, to rotate in the direction of the wind.
- **Gearbox:** It is used to increase the rotor speed up to the values required by conventional generators. The multiplier ratio in some turbines can exceed 1:100. The effect is accomplished in three steps. The first step is usually a multiplier planetary (input and output have the same center of rotation), while the others are parallel multipliers. The gearbox is lubricated and the oil is continuously filtered and cooled. As part of maintenance preventive maintenance, which is standard practice, operators normally check both the temperature of the multiplier and its vibrations.
- **Generator:** It is the unit of transformation of energy mechanical energy into electrical power. Asynchronous generators are essentially three-phase motors with induction. The generator transfers electrical energy to the grid. With a 50 Hz grid, the synchronous speed is 1,500 rpm. Active power represents the actual power that the generator transmits to the grid. Also, the generator consumes a certain amount of reactive power. This last value can be monitored and is often managed at the wind farm level to adjust the voltage on the grid

There are numerous wind turbine design variations, but the most commonly used technology today involves a three-bladed rotor. The operation of wind turbines leads one to believe that a turbine with many blades is more powerful than one with three. This conclusion is partially correct. More blades allow for greater power extraction, but they also increase the mutual disturbance that the blades cause each other due to the turbulent wake they leave behind. As a result of the reduction in thrust on the blades, a rotor with many blades is more powerful but less efficient than a rotor with few blades. A wind turbine with three blades placed at 120 degrees from each other represents the right compromise between power and efficiency. A turbine with four propellers means more weight and therefore more cost. Same for the turbine with two blades, which while achieving similar performance would require an increase in structural dimensions and therefore higher cost.

Designing a wind turbine involves several structural choices that are not directly related to power but take into account factors such as costs, maintenance operations, landscape impact, noise pollution, and many others. Maintenance is still a critical component in the proper operation of these devices. Unfortunately, wind turbines continue to be afflicted by some accidents that can occur if regular monitoring and maintenance are not performed. These may be mainly due to internal component failures or damage to the turbine blade lubrication system due to impacts [18].



Figure 2.3. Real nacelle of a WTG [20].

### 2.1.3 Wind Energy

The power associated with the flow of a moving air mass is equal to:

$$P = \frac{1}{2}\rho S_r V_0^3$$

where  $\rho$  is the air density,  $S_r$  is the section through which the air mass flows and  $V_0$  is the wind speed. The rotor faces the wind so the  $S_r$  represents the area of the rotor. Since a certain amount of energy is subtracted from the wind kinetic energy, the speed downwind of the rotor results is lower than the upwind one. As a result, the diameter of the flow tube at the back of the rotor is greater than that at the front. This behavior is depicted in Figure 2.4.

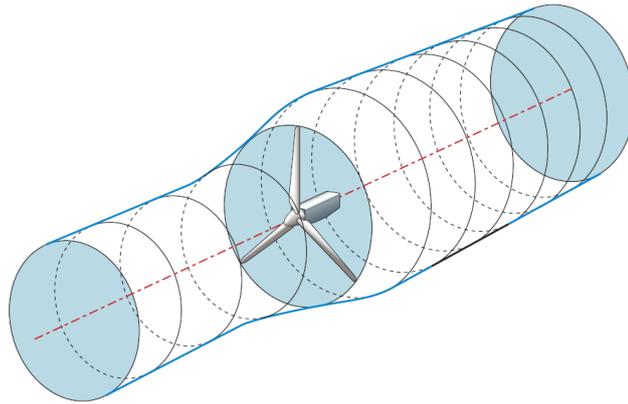


Figure 2.4. Flow pattern around a wind turbine.

Thanks to A. Betz, it was discovered that it is impossible to entirely convert the kinetic energy of a mass of air into mechanical energy. He also discovered that there is an upper limit to the amount of kinetic energy that can be converted. Due to this reason, Betz introduced a parameter, called the power coefficient  $C_p$ , which can be calculated as a function of the ratio of the speed of the wind wake behind the rotor to the speed ahead. The power produced by a wind turbine is thus:

$$P_{wtg} = \frac{1}{2}C_p\rho S_r V_0^3$$

The optimal value of this parameter is 0.593. As a result, convertible energy accounts for roughly one-third of wind energy. Therefore, it is not possible to design a turbine with a higher value of the power coefficient. But, this condition occurs for a rotor under ideal conditions when the outgoing air velocity is one-third of the incoming air velocity and the rotor is infinitely thin. Today's turbines have power coefficients of about 70-80% of the theoretical limit. On a WTG, there are two primary controls:

- Blade pitch angle control: a blade is similar to a wing. The surface area available to the incoming wind is critical for increasing aerodynamic forces on the rotor blades. The angle of attack  $\alpha$  is defined as the angle between the incident flow vector and the plane of the blade segment, while the pitch angle  $\beta$  represents the angle between the plane of the blade segment and the plane of the rotor. Blade pitch angle control can rotate the blades around their axis in such a way as to have, for each wind speed, an optimal blade angle of attack.
- Yaw control: by rotating the nacelle, the turbine is oriented, actively or passively, in the direction of the wind to maximize the efficiency of the energy conversion process.

Both types of controls are depicted in Figure 2.5.

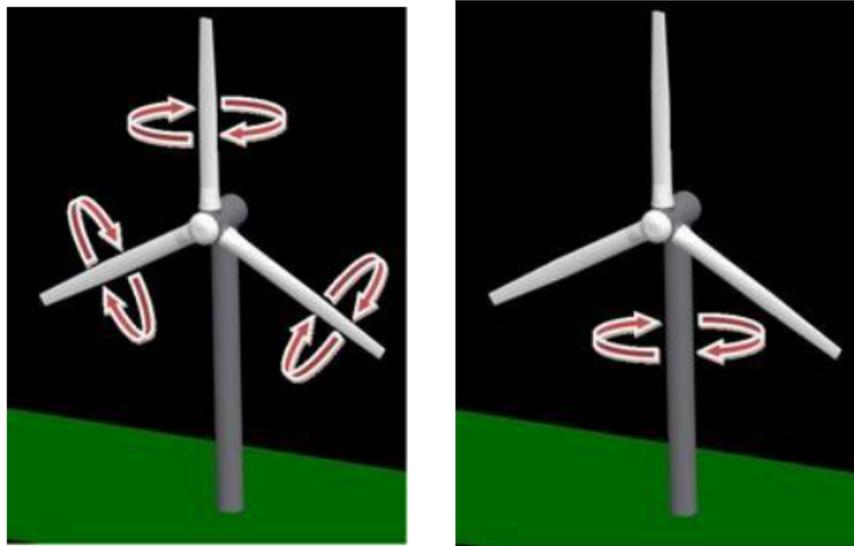


Figure 2.5. From left to right: blade pitch angle [19] and yaw [19] controls.

#### 2.1.4 Power Curve

As it is possible to see in Figure 2.6, the power curve of a wind turbine is a graph representing the relationship that links the electrical power produced by a wind turbine to wind speed. Although wind turbine manufacturers provide guaranteed power curves, the performance of a wind turbine can vary depending on weather conditions as well as the location in which it operates. As a result, a power curve must be created that represents the normal operation of each WTG installed at a given site. Wind turbine performance is then monitored and compared to that of normal operation.

##### Cut-in-speed

At very low wind speeds, the wind force on the turbine blades is insufficient to rotate them, so the active power produced by the turbine is zero. As the wind speed increases, the wind turbine begins to rotate and generate power.

The speed at which the turbine begins to rotate and generate power is known as cut-in speed and it is typically between 3 and 4 meters per second.

### Rated output speed

When the wind speed exceeds the cut-in speed, the amount of electric power output rapidly increases. However, when a certain speed is reached, the power output reaches the limit of the electric generator. Such limit is known as the rated output power and the wind speed at which it is reached is known as the rated output wind speed. The turbine design is intended to limit power to this maximum level at higher wind speeds.

### Cut-out-speed

When the speed exceeds the rated output wind speed, the forces on the turbine structure increase and the rotor is at risk of being damaged. Consequently, once a certain speed is exceeded, braking is used to stop the rotor. This limit is called the cut-out speed, and when it is exceeded the energy produced returns to zero.

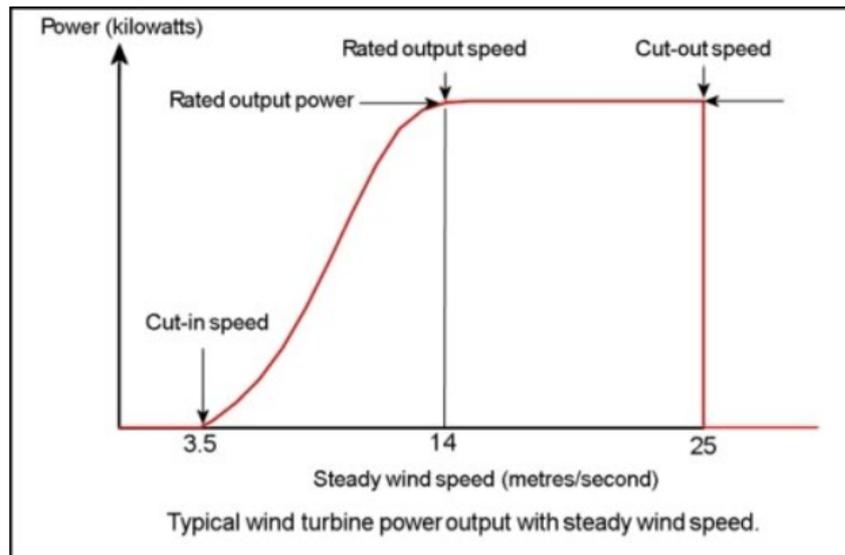


Figure 2.6. Ideal power curve [21].

## 2.2 Notions of Statistic

### 2.2.1 Standard Deviation

Standard deviation is a statistical dispersion index, representing an estimate of the variability of a population of data or a random variable. A low standard deviation indicates that the values are close to the mean, which represents the expected value of the set; on the contrary, a high standard deviation indicates that the values are distributed over a wider range. Standard deviation, abbreviated as SD or STD, is most generally represented in mathematical texts and equations mostly by lowercase Greek letter  $\sigma$ :

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

8

where:

- $N$  is the number of samples of the population.
- $x_i$  are the current samples.
- $\mu$  is the arithmetic mean.

The arithmetic mean  $\mu$  of two or more numbers, or simply average, is the sum of the numerical values considered divided by their number.

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

Figure 2.7 shows an example of two samples with the same mean (100) but different standard deviations. They are illustrated as a measure of dispersion around the mean. The red population has a standard deviation of 10 and the blue population has a standard deviation of 50 [22].

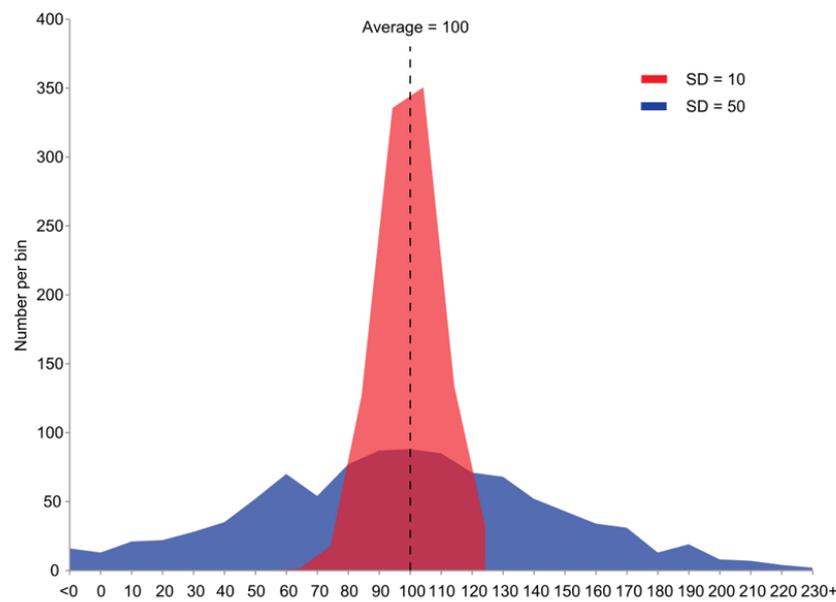


Figure 2.7. Example of STDs of two samples with the same mean [22].

## 2.2.2 Normal Distribution

The normal distribution, also known as the Gauss distribution, is a continuous probability distribution that is frequently used as a first approximation to describe real-valued random variables that have a single mean value.

The associated graph of the probability density function is symmetrical and has a bell shape, which is known as the bell curve or normal curve of error.

## 2.2.3 Empirical Rule

The empirical rule, also known as the 68-95-99.7 rule, is a shortened form used in statistics to recall the percentage of values in a normal distribution that falls within an estimated range:

- Around 68% of scores is within 1 standard deviation of the mean.
- Around 95% of scores is within 2 standard deviations of the mean.
- Around 99.7% of scores is within 3 standard deviations of the mean.

The empirical rule (or  $3\sigma$  rule), which is shown in Figure 2.8, asserts a convention in which almost all values are considered to be within 3 standard deviations of the mean, and thus it is empirically considered 99.7 percent probability as an almost certainty.

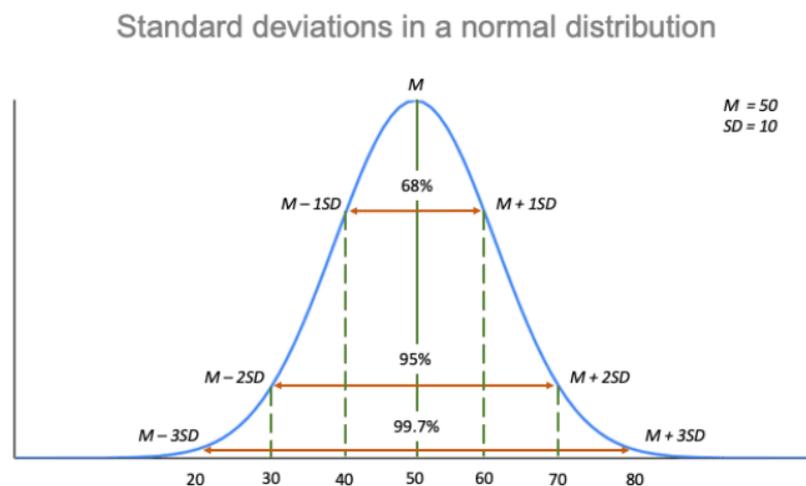


Figure 2.8. The Empirical Rule: for a normal distribution 99.7% of values are inside 3 STD of the mean [23].

## 2.2.4 Control Chart

Control charts, also known as Shewhart charts, are analysis tools used in project quality control and are typically used for the analysis of repetitive processes. They aid in the visualization of variations, the detection and correction of problems when they occur, the prediction of ranges of expected results, and the analysis of patterns of process variation caused by unique or common causes.

### Control Limits

Control limits are plotted in lines above and below the center line in quality control charts. The center line represents the mean. Control limits act as thresholds to identify abnormal signals (points over and above limits) and aid in determining whether a process is statistically controlled.

- Upper Control Limit (UCL): represents the upper limit within which the data is acceptable.
- Lower Control Limit (LCL): represents the lower limit within which the data is acceptable.

Points outside of the UCL and LCL values indicate that the process is out of control and/or unstable.

Control charts show how a process behaves over time and when it is subject to a variation caused by an unusual cause, resulting in an out-of-control condition and unpredictability.

## Individual Chart

The individual chart is used to track the mean and variation of a process over time using individual samples. As it is possible to see in the example reported in Figure 2.9, the y-axis shows the mean and the control limits while the x-axis shows the sample units. Control rules are based on the Empirical rule, so the data must be normally distributed (or transformed) when using control charts, otherwise the chart may report an unexpectedly high rate of false alarms.

The calculation of control limits looks like this:

- UCL :  $\mu + 3\sigma$ .
- LCL :  $\mu - 3\sigma$ .

The value of  $3\sigma$  can still be modified, either to include more values within the control or to fit the chart to distributions that are not perfectly normal, in which case the empiric rule may no longer be valid.

The number of standard deviations from the mean of a value is known as the Z-score. The Z-score is a statistical measure that represents the relationship between a value and the mean of the values in a group. A Z-score equal to 0 indicates that the score is equal to the mean.

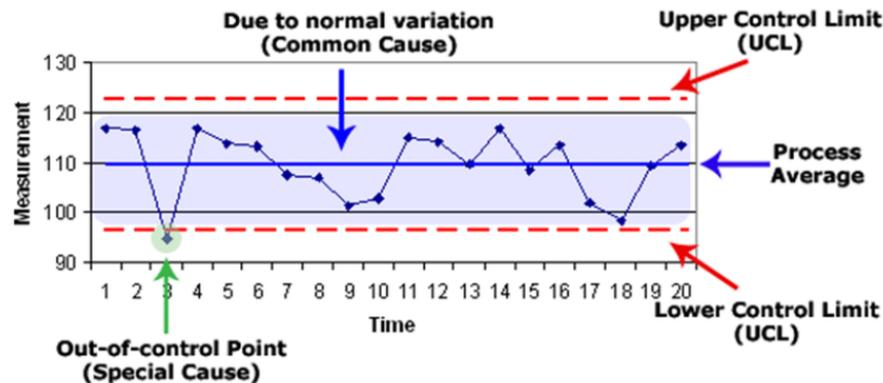


Figure 2.9. Example of individual chart with UCL and LCL limits [24].

### 2.2.5 Data Smoothing

Data smoothing is a statistical technique that removes or reduces outliers from data sets by using statistical noise removal algorithms to make patterns more visible.

This method utilizes simple filtering to better predict different patterns. It concentrates on creating a basic direction for data points, avoiding volatile data, and drawing a smoother curve between data points.

### Data Binning

To get a better view of a pattern in the data set, it may be useful to divide the original data into small intervals to be analyzed separately. This operation can be performed through Data binning, which is a method that involves sorting data and then distributing the sorted values into several intervals known as bins. They perform local smoothing by consulting the vicinity of the values.

There are 2 methods to divide data into bins:

- Equal Frequency Binning, where bins have an equal frequency.
- Equal Width Binning, where bins have equal width. The range of each bin is defined as:

$$[min + w], [min + 2w] \dots [min + nw]$$

where

$$w = \frac{max - min}{no\ of\ bins}$$

### 2.2.6 Correlation Matrix

A correlation matrix is a table that displays the correlation coefficients for various variables. It is a useful tool for identifying and visualizing data patterns. It is made up of rows and columns that represent the variables, and each cell in the table contains the correlation coefficient. Furthermore, it is frequently used in the analysis of multiple linear regression models. The correlation matrix in multiple linear regression determines the correlation coefficients between the independent variables in a model [27].

#### Correlation

The correlation coefficient is a number that indicates how strong the relationship between the variables is. It can have values ranging from -1 to 1. In particular:

- -1: Completely negative correlation. This implies that the variables tend to move in opposite directions (i.e., when one variable increases, the other decreases).
- 0: There is no correlation. The variables are unrelated to each other.
- 1: Completely positive correlation. This implies that the variables tend to move in the same direction (i.e., when one variable increases, the other also increases).

Figure 2.10 shows an example of a correlation matrix.

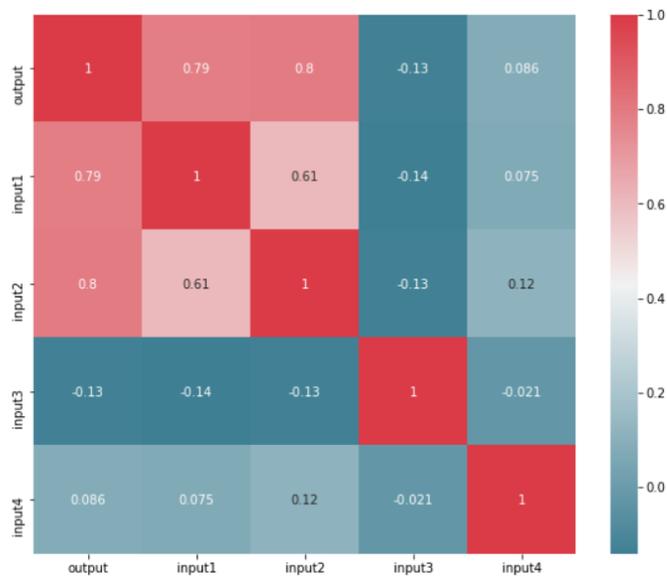


Figure 2.10. Example of correlation matrix [26].

## 2.3 Time Series Forecasting Theory

Time series data are observations of a single unit collected in an orderly way at many points in time. They can be collected continuously over time or discretely using regular intervals or simply at scheduled times. In terms of the number of parameters in the data, time series data can be univariate if only one observation parameter is present or multivariate if multiple parameters are present. Time series data is a collection of several sensor measurements for a single turbine over a long time.

The sequential nature of the data influences the selection of specific data pre-processing techniques and patterns capable of capturing patterns. Machine learning recurs to statistical techniques that allow a machine to learn patterns from data and make predictions about them based on them. While analysis allows making assumptions based on existing data, forecasting allows predicting future values based on historical data. This difference is highlighted in the example shown in Figure 2.11.

Sensor_ID	Time Stamp	Value
Sensor_1	01/01/2020	20
Sensor_1	01/02/2020	21
Sensor_2	01/01/2020	22
Sensor_2	01/02/2020	23
Sensor_1	01/05/2020	?
Sensor_1	01/06/2020	?

The table is annotated with two brackets: a left-side bracket labeled "Time series analysis domain" covers the first four rows, and a right-side bracket labeled "Time series forecasting" covers the last two rows.

Figure 2.11. Difference between time series analysis and time series forecasting [29].

Real-world data are frequently influenced by deterministic components that are difficult to detect at first glance via visual analysis. Trends, cyclicity, seasonality, and irregularities are examples of these components. Irregularities are components that cannot be explained by data. A trend defines a long-term increase or decrease in data that can also change direction. Seasonality and cyclicity are related but differ in the length of intervals. In fact, seasons have a relatively fixed duration, whereas cycles can repeat themselves over time. Figure 2.12 graphically depicts the differences between these components.

The deep learning models under consideration in this thesis are known to be capable of dealing with nonstationarity in data.

### 2.3.1 Rolling Window

The input and output sequence formats used for the problem under consideration in this thesis are created using the rolling window principle. A sequence of values is used as input while a single value is returned as output. The method used for data transformation is the sliding window or moving window. According to this principle, the data is divided into windows of a certain length, where each successive window sequence starts with the value one step later. For this experiment, the window length was set to 2 hours or 12 timestamps (equivalent to 10-minute averaged values). The timestamp represents a digital record of the time of an event. The sliding step is 1. The sliding window principle is shown in Figure 2.13.

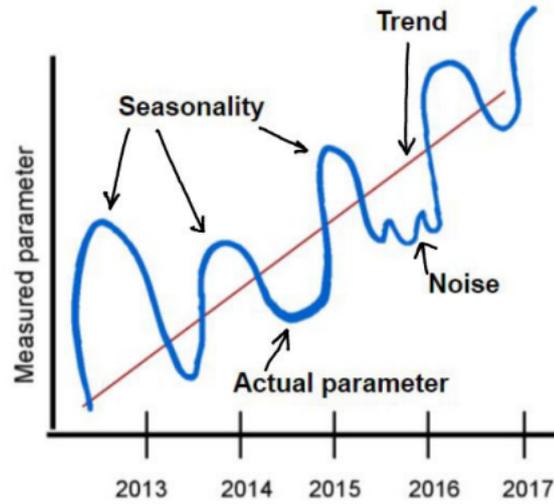


Figure 2.12. Seasonal, trend and noise components of the time series [29].

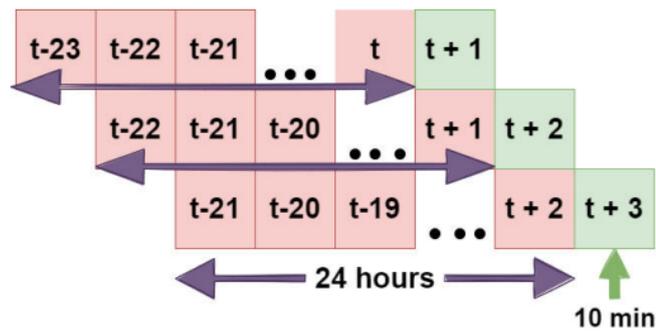


Figure 2.13. Rolling window method [29].

## 2.4 Machine Learning

### 2.4.1 General Introduction

In terms of processing and analysis, the amount of data generated by a wind turbine is beyond human reach, while it can be feasible with the help of computers. *Machine Learning (ML)* is a branch of *Artificial Intelligence (AI)* and *Computer Science (CS)* that focuses on the use of big data models to make predictions. Machine learning can be subdivided into three main categories which differ based on the learning approach:

- *Supervised Learning (SL)*, which builds a model from labeled training data, which is then used to try to make predictions about unavailable or future data. The term supervised means that in the set of samples the desired outputs are already known since they were previously labeled.
- *Unsupervised Learning (UL)*, which does not exploit labeled data but observes the structure of the data, identifies patterns, and extracts meaning-laden information.
- *Reinforcement Learning (RL)*, which focuses on creating a system that can maximize its performance based on interactions with the environment. Reward signals, also called reinforcers, are introduced to improve performance. This reinforcement is not given by labels

(tags) and is a measurement of the quality of actions taken by the system. By evaluating the metric values, it can be verified whether the model can generalize or not.

### 2.4.2 Regression and Classification

Based on the desired output, supervised algorithms can be divided into:

- *Classification algorithms*, which are capable of predicting the class to which new, never seen data belong, based on the knowledge acquired from labeled data. Labels are unordered discrete values that belong to a group of a class. The classification problem may be binary or multi-class.
- *Regression algorithms*, which are used to handle problems with continuous output. The input parameters are presented in the same format as the classification problems, while the labels are quantitative. In regression models, there are several predictor variables (descriptive) and a continuous target variable (the outcome). The model looks for a relationship between these variables in order to predict an outcome. The patterns discovered in the input data are saved as model parameters, and this knowledge is later applied to new data to predict the output. The type of output and the model are decisive in the choice of evaluation metrics.

### 2.4.3 Artificial Neural Network

An *Artificial Neural Network (ANN)* is a mathematical model that mimics the human brain through a set of algorithms [30]. The concept of ANN was first proposed by Warren McCulloch and Walter Pitts in 1943 and was then implemented by Frank Rosenblatt in 1958 as a *Perceptron*, which is the mathematical model of a neuron. Its structure is similar to that of a biological neuron [31].

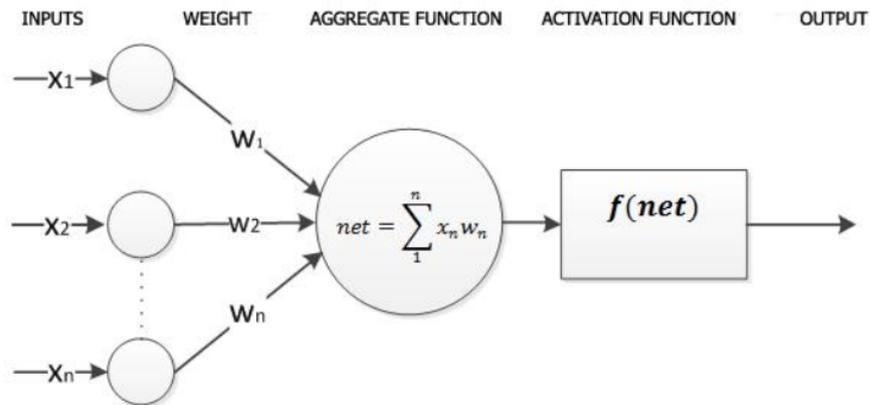


Figure 2.14. The implementation of an artificial neuron [32].

In Figure 2.14 the structure of an artificial neuron is shown alongside the description of its logic. Each of the  $n$  inputs, represented by  $x_1, x_2, x_n$ , is multiplied by its relative weight and the result is added with a value called bias to obtain the net input function  $\Sigma$ . The bias allows for the introduction of a shift into the activation function. The activation function determines the calculation of the output. The result proceeds to the activation function, which determines the calculation of the output and if a neuron should be activated or not. The scope is to decide whether or not the input of the neuron provided to the network is significant during the prediction process.

As represented in Figure 2.15, an Artificial Neural Network is composed of multiple neurons called nodes connected to each other through links called synapses. A simple ANN consists of

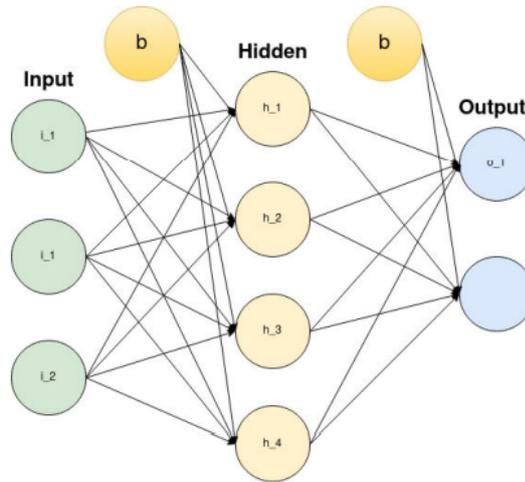


Figure 2.15. Example of an Artificial Neural Network [29].

at least three layers of simple neurons. The first layer, called the input layer, is followed by the hidden layer in the middle and an output layer. The neurons in the layers are interconnected and a weight is assigned to each connection.

During the training phase, the output values are compared with labels and the result is called the loss function or cost function. The loss function is used in the back-propagation process to adjust the output results, which means that the error in the output layer is carried back through the neural network. Instead, during the testing phase, a new set of data is sent to the neural network and the output error value represents the value of the metric. ANNs are part of supervised learning as they normally require labeled data to learn the task to do.

#### 2.4.4 Types of ANNs and Terminology

A *Multilayer Perceptron (MLP)* is a type of *Feedforward Neural Network (FNN)* in which each layer is fully connected [34]. An example of MLP is shown in Figure 2.16.

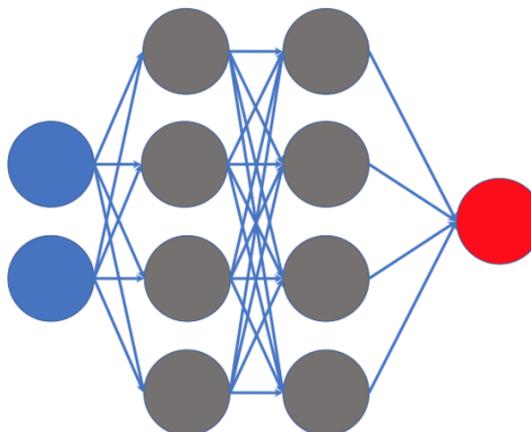


Figure 2.16. Multilayer Perceptron configuration [2].

*Recurrent Neural Networks (RNN)* represent another type of supervised ANN which will be better highlighted in the next subsection.

The term *Deep Learning (DL)* refers to a collection of neural networks. Deep learning algorithms are based on ANNs having at least four layers. [30]. The Artificial Neural Network represents a black box approach: it is therefore not possible to determine the number of layers or nodes per layer to be used to reach the objective. It is, therefore, necessary to make multiple attempts to arrive at an optimal solution [38].

### 2.4.5 Recurrent Neural Network

*Recurrent Neural Networks (RNNs)* are ANNs that, differently from FNNs which only admit connections between nodes in different layers, allow connections even within the same layer. This characteristic is depicted in Figure 2.17. In RNNs, nodes can also admit loops and/or can also be connected with neurons from a previous layer. This feature allows RNNs to "remember" the previous input while processing a new one. In an RNN network, the output of a neuron can influence both itself and other neurons at a later time, which in turn will again interfere with the behavior of the neuron by forming a loop. Time series require these mechanisms because the ordered sequence of input values is critical for predicting the output.

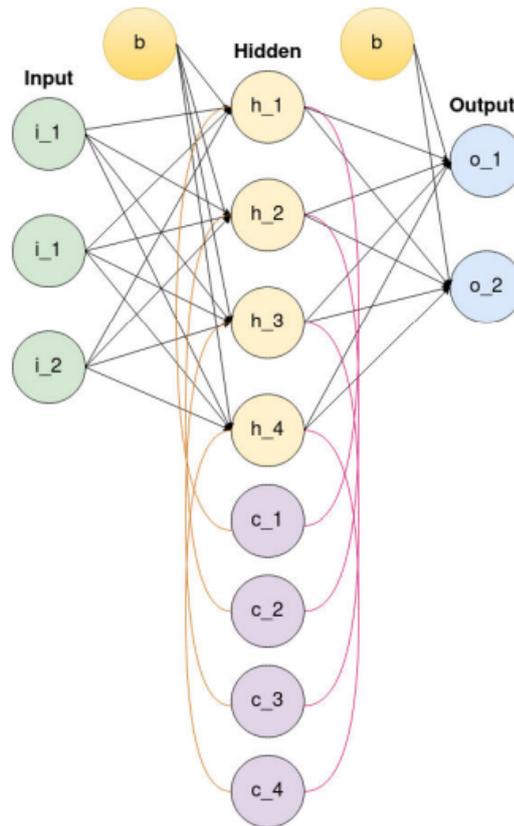


Figure 2.17. Example of Recurrent Neural Network [29].

An RNN network can be represented by a cell with a loop. Through a network unfolding operation, depicted in Figure 2.18, the RNN is transformed into a feed-forward one. An RNN cell is a recurrent network section that preserves an internal state  $h(t)$  for each time instant. A cell consists of a fixed number of neurons and can be considered a kind of layer. In this network at each instant the output will be  $h_t = f(h_{(t-1)}, X_t)$  where  $h_t$  depends on the input  $X_t$  and the previous state  $h_{(t-1)}$ .

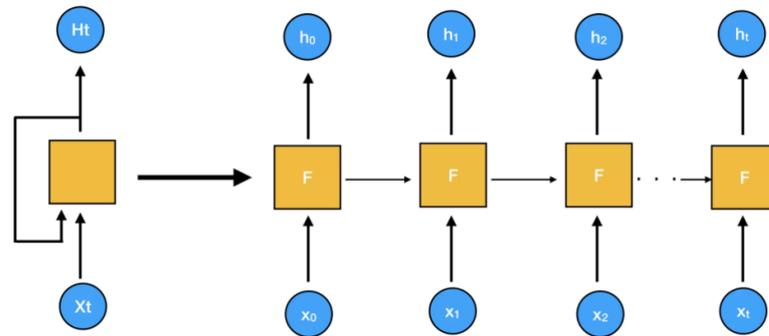


Figure 2.18. Unfolding operation performed on a RNN cell [35].

The input and output of RNN models can be single data or sequences in all 4 combinations. Particularly for regression problems, these versions are implemented:

- Many-to-one with a sequence as input and a single time as output.
- Many-to-many with both input and output sequences.

Time series with many inputs suffer from the problem of disappearing or exploding gradients, and this happens because updating the weights for an output requires numerous multiplications that cause the weights to tend to zero or infinity

### Long Short-Term Memory

*Long Short-Term Memory (LSTM)* neural networks are particular RNNs capable of learning dependencies in long time series data, and they solve the problem of the vanishing of gradient. Like all RNNs, LSTMs can be considered as a set of memory cells. The main feature of the LSTM cell is that it is able to control how much information to remember from the previous cell, how much information to retain from the current cell, and how much to inject into the next cell. These tasks are handled by the gates of the cell. (Figure: 2.19). Each LSTM hidden layer has as many hidden cells as the number of time steps. Moreover, each hidden cell is composed of multiple hidden units. The unit represents the number of neurons per cell. [50]

### LSTM Model For Time Series Forecasting

Long Short-Term Memory networks can be used to forecast time series and are divided as follows:

- A *Univariate time series* is based on a single dependent variable. Basic to the univariate forecasting approach is that the value of a time series at time  $T$  is related exclusively to the values of the same variable in previous time steps.
- A *Multivariate time series* is composed of several time-dependent variables, each of which is dependent not only on its previous values but also on other variables.

LSTM for Time Series Forecasting:

- Univariate LSTM models: single time series variable, prediction of the next value in the sequence.
- Multivariate LSTM models: time series data where there is more than one observation (variable) for each time-stamp and prediction of the next value in the sequence.[45]

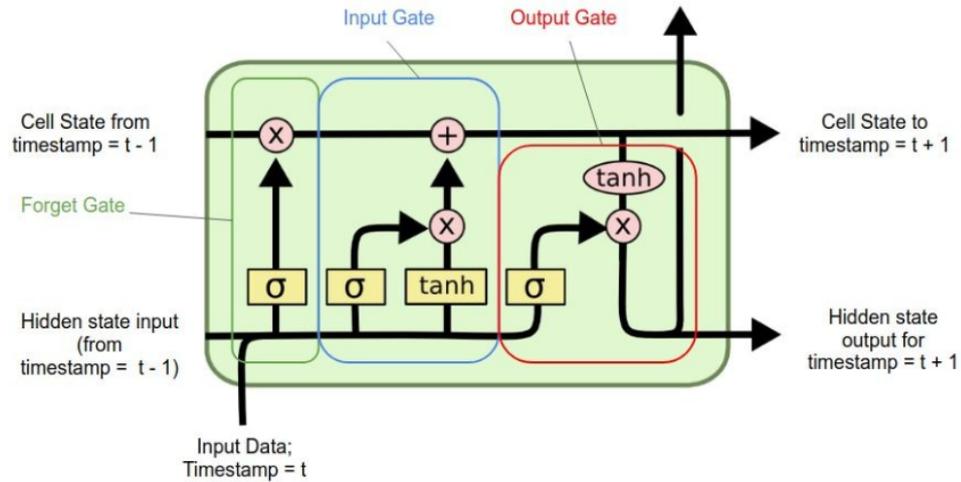


Figure 2.19. LSTM Cell [37].

- Multiple input time series: multiple parallel input time series observation and one output time series that depends on the multiple input time series inputs.
- Multiple parallel time series: multiple parallel time series and a next value must be forecast for each.
- Univariate Multi-Step LSTM Models: one variable in time series, prediction of multiple values in a sequence of the series.
- Multivariate Multi-Step LSTM models: multiple observation time series, multi-step value prediction (i.e., multiple consecutive values) in the prediction sequence.
  - Multiple inputs and multiple outputs.
  - Multiple inputs and multiple outputs in parallel.

## 2.4.6 Overfitting and Underfitting

Generalization refers to the ability to apply concepts learned from a machine learning model to other examples not seen by the model during learning [39]. A good model should generalize well: this means that it should work the same way with both used data already seen and new data from the same problem domain. A model is called:

- overfitted when it learns too many features from the input data, thus limiting its ability to generalize to new data. This means that random fluctuations or noise in the training data are learned as concepts by the model.
- underfitted when it fails to learn sufficient features from the input data and is unable to generalize to new data.

An optimal model will be neither overfitted nor underfitted.

## 2.4.7 Metrics

Metrics are functions used to monitor and measure the performance of a model. The purpose of the metrics is to provide a numerical estimate of the model's performance in the prediction task. The most commonly used evaluation metrics for regression problems are Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Error (MAE),  $R^2$ .

- Mean absolute error (MAE) represents the average of the absolute difference between the original and predicted values in a data set.
- Mean Squared Error (MSE) represents the average of the squared difference between the actual and predicted values in a data set.
- Root Mean Squared Error (RMSE) is the square root of the Mean Squared error. The advantage over MSE is that the error is comparable with the output measure. It measures the standard deviation of the error, so it is the most widely used.
- $R^2$  is an indicator that, starting from the regression line, summarizes in a single numerical value how much the analyzed quantity deviates from that line on average, so it is an indicator of model goodness-of-fit. The best performance is achieved when this value is equal to 1.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

where:

$\hat{y}$  - predicted value of y

$\bar{y}$  - mean value of y

The lower the values of MAE, MSE, and RMSE, the better a model fits the data. The higher the  $R^2$  value, the better the model fits a data set and indicates how well the predictor variables can explain the variation in the response variable

RMSE and MSE, compared with MAE, are more sensitive to observations that are further from the mean because they are based on the square of the error. This implies that RMSE is most useful when large errors are particularly undesirable.

Performance metrics are measured for each epoch and then can be represented as learning curves. The optimal goal is for the training and validation loss to be as close as possible to each other, while both are in the proximity of 0.

## 2.4.8 Hyperparameters

In machine learning, a hyperparameter is a parameter that is used to control the learning process. They are variables that must be set before training and that influence the network training process. The most relevant hyperparameters are the following [40]:

- Learning rate: it controls how much we are adjusting the network weights with respect to the loss gradient. It represents the speed of network updates; low values slow down the learning process but allow learning an optimal set of weights, while high values speed it up but increase the risk, for the network, of not calculating the weights well and thus overfit.
- Sample: is a single element of a data set. It can be, for example, a single observation at a given time-stamp in a time series.
- Batch: is a set of N samples, processed independently and in parallel. The batch size is a hyperparameter that defines the number of samples to be analyzed together during training before updating the internal model parameters.

- Epoch: is a hyperparameter that defines the number of times the learning algorithm will work on the entire training dataset. And used to divide training into multiple stages to perform periodic evaluations of the model [41].

With Keras and similar tools some callback functions can be called at the end of each epoch; such as the learning rate changer (ReduceLROnPlateau in Keras), and the early training stopper (EarlyStopping in Keras): these two callbacks are called when the outputs of the metrics have stopped improving [42].

EarlyStopping stops the training of the model if there is no improvement in the value of the loss function or a metric on the validation set for a certain epoch (patience) [52].

ReduceLROnPlateau reduces the learning rate parameter of the optimization function with a certain value (factor) if there is no improvement on the value of the loss function or a metric for the validation set after a certain epoch (patience), until a limit is reached (min lr, abbreviation of minimum learning rate).

### 2.4.9 Partition of Database

For training a neural network, it is advisable to divide the database into three parts [43]:

- Training dataset, which is used to train the network.
- Validation dataset, which is used to evaluate the network during each epoch of training.
- Test dataset, which is used to test the network in order to verify its ability to generalize toward new data.

This subdivision is illustrated in Figure 2.20. The commonly used percentages for the split between train, validation, and test are 60%-20%-20% or 80%-10%-10%. The last configuration was used in this thesis.

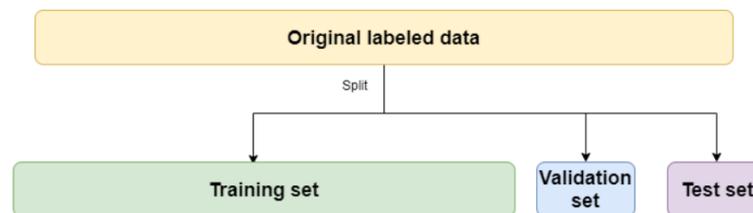


Figure 2.20. Splitting operation of a dataset [44].

### 2.4.10 Normalization

Some machine learning algorithms perform better when the numerical input variables are scaled to a standard range. Normalization is a set of operations used to get the numerical characteristics of a distribution on a similar scale. The two most popular techniques for scaling numerical data are the following:

- Scaling to a range, which scales each input variable separately from each other in a certain range, usually set to 0-1. It is defined as follows:

$$X_{sc} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

- Standardization (z-score standardization), which scales each input variable separately by subtracting its mean and dividing by its standard deviation. This causes the distribution to have a mean of 0 and a standard deviation of 1. It is defined as follows:

$$z = \frac{x_i - \mu}{\sigma}$$

# Chapter 3

## Methodology

This chapter will discuss the individual steps performed necessary to create the final model.

### 3.1 Introduction to the Methodology

The work done in this thesis aims to propose a comprehensive methodology to design an effective approach for real-time **alarm prediction and fault detection** based on the use of Artificial Neural Networks and Statistical Process Control (SPC). While most similar works have relied on utilizing data from a single turbine to predict future failures on it, this approach tries to use data from multiple turbines of the same model and wind farm to try to predict a downtime on any turbine in the farm, even when newly installed.

The development of a good model requires large databases covering several years of operation of a wind turbine and information on the maintenance work performed. These premises were unfortunately not applicable in this case study, so the biggest challenge was to find a different approach. The main disadvantage was the absence of a comprehensive maintenance list (which presented only information on routine maintenance and replacement of minor components) so the alarm system (Downtime Manager) was relied upon to validate the ANN models. In detail, the Downtime Manager causes the turbine to shut down when an alarm is generated and consequently the turbine does not produce. These alarms are due to outliers on data derived from SCADA measures or shutdowns caused by wind farm operators. After the emergency is over, the turbine returns to regular production. The solution proposed in this thesis is not intended to replace the Downtime Manager, but to try to anticipate the alarms generated by it through the generation of other alarms, or to try to find anomalous behaviors that are not reported. A series of alarms close in time can be interpreted as a possible fault of the turbine or some of its components in the near future.

The methodology describes all of the steps required to put in place this control system. So the first task performed was to analyze the data of all wind farms managed by Sirius and at the same time available on the VireoXPower software to search for a wind farm with enough samples and with numerous observations (number of information) for each sample. After several analyses, a wind farm was selected because it had numerous turbines on which numerous sensors were installed. Despite this, the monitoring periods had a maximum length of one year of work so it was decided to merge data from multiple turbines.

As depicted in Figure 3.1, the work can be divided into two phases:

- **Models Elaboration**, in which, in order to develop different models representative of the state of health of the wind turbines, an analysis of the data was made to identify extended periods in which they were free from prolonged downtime.

- **Control**, where the developed models served as a reference. With data from new turbines representing a general steady-state operation, the ability of the system to identify the presence of anomalies was validated. Only after this final phase, the models were considered ready to be used on real-time data.

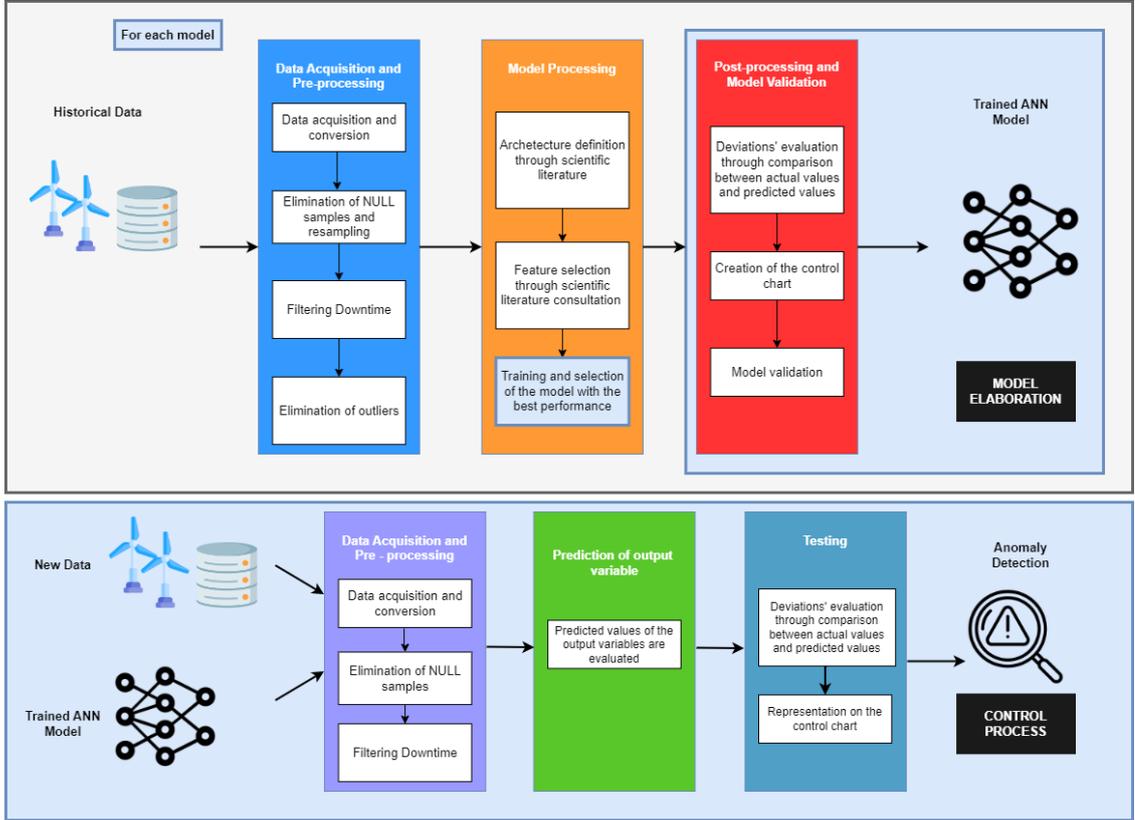


Figure 3.1. This figure describes the methodology proposed in the thesis. In light blue are evidenced the operations performed for each proposed model [46] [47] [48] [49].

## 3.2 Models Elaboration

### 3.2.1 Data Acquisition and Preprocessing

The first step concerned the selection of turbines to be used for model realization, choosing from 10 possible turbines from the wind farm. The turbines with the fewest problems and a few prolonged downtimes were chosen. The acquired data were converted to have a similar and useful format to be processed.

Subsequently, data cleaning operation were taken into account. The presence of a NULL value on each sample implies the elimination of the entire sample. Since limited data available are available, it was chosen to perform a resampling operation on the absent samples. This operation is essential particularly to train a network based on time series because the absence of a timestamp involves the exclusion of many time series from the training of the neural network. Since the various supervised learning models, during their training phase, need to learn from wind turbine data to **predict possible downtime**, those that actually occurred have been removed, including short ones. The limitation periods of the wind farm have also been removed from the available data, as during these periods the wind turbines do not produce at maximum efficiency. Finally, to represent the health status of the turbine, the data are filtered to eliminate all possible remaining outliers, even through the analysis of the power curve of each wind turbine.

During model training, the presence of outliers is a particularly dangerous condition, since it can compromise the accuracy of the various models.

### 3.2.2 Model Processing

The next step was a study of model generation, focusing on architecture and feature selection. ANNs are a powerful modeling tool because they are capable of modeling the complex nonlinear relationships between the characteristic parameters of a wind turbine. An elaborate study of the literature was performed to analyze the various approaches and performances, which allowed getting an idea about the type of ANN and the approximate reference number for layers, nodes, and cells.

Therefore, it was decided to build multiple models to predict different output variables. The models can be divided into 4 categories :

- **Wind Turbine Models - output: Active Power.** A first model category was built to predict the active power produced by the turbine, having as input only environmental data not derived from measurements on internal components sensors. This allowed trying to detect generic emergencies not better identified by the Downtime Manager.
- **Gearbox Oil Models - output: Gearbox Oil Temperature.** A second model category aimed to predict gearbox oil temperatures in order to look for general malfunctions of this component. The gearbox is a component that is often affected by faults.
- **Gearbox Bearing Models - output: Gearbox Bearing Temperature.** A third model category tried to predict the temperature of the gearbox bearing.
- **Generator Models - output: Generator Temperature.** A final model category tried to predict the temperature of the generator.

Regarding feature selection, different types of variable inputs were provided for each model. The choice of the type of network to be used fell on some types of FNN and some multivariate LSTM, based on a multiple-input time series. Different neural network architectures were taken into account to train and evaluate all models.

To carry out the training of each individual model, the available dataset was separated into three parts:

1. Training dataset: to effectively train the model
2. Validation dataset: needed to evaluate the efficiency of the model and solve the overfitting problem
3. Test dataset: new data for the model are used to evaluate its actual performance.

After performance analysis through metrics, the best network, alongside its feature input and output, was selected for each model.

### 3.2.3 Validation

Initially, the best model is selected for each category. Each trained model needs to be validated through the control charts. The model generates the value of the output variable, which characterizes the state of health of the system. The output variable is then compared with the real value measured by the system at the same time. The difference between the two values is evaluated using the control charts to identify possible anomalies present in the system. The preliminary analysis of the values of the test sets through the control charts allows for checking that the variance of the deviations calculate for all the samples is small. This informs that there are no anomalies in these sets, but also allows the definition of the Upper Control Limit and Lower Control Limit values to be used in the Shewhart charts for the control phase.

### 3.3 Control phase

#### 3.3.1 Preprocessing

At this stage, filtering operations involved selecting data from when the turbine is in production and not in an idle state and eliminating those samples with NULL values. Samples related to downtime or limited power state are also removed since only data on which the turbine is producing should be provided to the model.

#### 3.3.2 Prediction of Output Variable

In this phase, the previously selected best models are used to predict the values of the related output variable, on preprocessed data.

#### 3.3.3 Testing

As before, the difference between the real and predicted value of the output variable is calculated for each timestamp and evaluated with different control charts. It is important to find a correlation between these values above the threshold and the info reported in the Downtime Manager, so various solutions have been proposed to improve Shewhart control charts performance, also analyzing the frequency of errors over time.

As summarized in Figure 3.2, the validated system can then be used to find errors in real-time, using all of the models that have been demonstrated to perform well in their respective objectives.

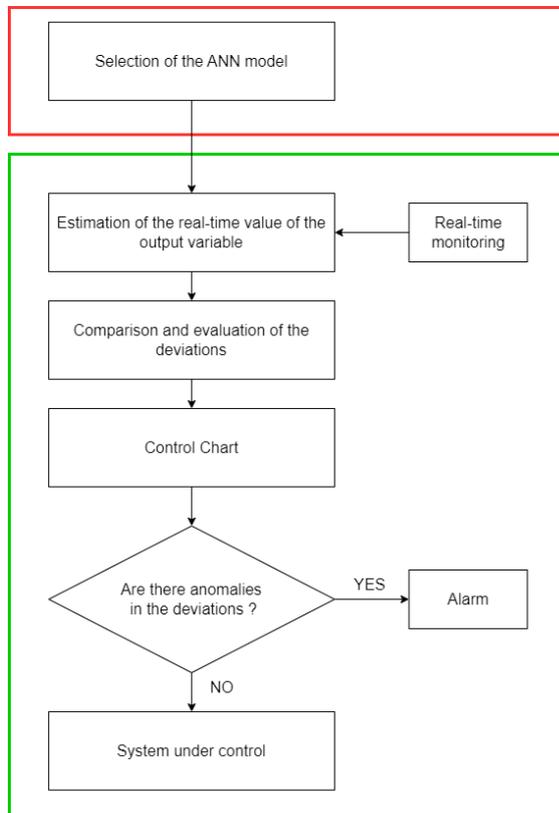


Figure 3.2. Methodology of the system on real-time monitoring.

## Chapter 4

# Solution - Model Elaboration phase

This chapter will discuss the Model Elaboration phase, showing the different solutions developed to address the problem of developing models that represent a turbine's healthy status, with an emphasis on the technology and approaches used.

### 4.1 Gamesa G90

The proposed solution was applied to a wind farm in southern Italy consisting of 10 G90 wind turbines of the Spanish manufacturer Gamesa Corporación Tecnológica (Grupo Auxiliar Metalurgico, SA) [16]. This manufacturer has been in business since 1994. However, since 2017, the Grupo Auxiliar METalurgico, SA, a division of Gamesa Corporación Tecnológica, has remained inactive and was taken over by Siemens Gamesa Renewable Energy. The main technical specifications of the G90 wind turbine are as follows:

- Power:
  - Rated power: 2.00 MW.
  - Cut-in wind speed: 3 m/s.
  - Cut-out wind speed: 25 m/s.
  - Survival wind speed: 49 m/s
- Rotor:
  - Blade number: 3.
  - Diameter: 90 m.
  - Area: 6,362  $m^2$ .
- Gearbox:
  - Type: spur/planetary.
  - Gear stages: 3.
  - Transmission Ratio: 0.11.
  - Brand: Echesa (Gamesa Group).
- Generator:
  - Type: Doubly-fed Asynchronous.
  - Brand: Cantarey.

- Voltage: 690 V.
- Main frequency: 50 Hz.
- Speed max: 1900 U/min.
- Weight:
  - Single Blade: 5.8 t.
  - Hub: 18.6 t.
  - Rotor: 36 t.
  - Nacelle: 70 t.
  - Tower: 255 t.
  - Total: 361 t.

The tower made by Gamesa is a large circular steel tube characterized by multiple coating as corrosion protection.

## 4.2 Data Acquisition

At the beginning of the thesis, an analysis of the data available was made by using the VireoX-Power software developed by Sirius. It allows to store and display the different measurements made available through the sensors installed on the various wind turbines in each field and on the period of storage of this information over time. For each wind farm, 10-minute averaged data is available. Normally this operation is performed to reduce the amount of space required to store the huge quantity of information that continuously arrives from all wind farms. The type of alarms affecting the different turbines was then taken into account in an effort to foresee potential failures. Following an analysis of the available wind farms, the choice fell to one of them. For confidentiality reasons, will not be explicitly mentioned. Although a lot of data has been made available, a period of about one year is available. This amount of data could then be insufficient for the training of neural networks if seasonal phenomena are wanted to be taken into account. Given the large number of turbines of the same model in the same location, it was decided to train the neural networks using data from 3 different wind turbines.

After identifying the data of interest, the dataset for each wind turbine was downloaded. Only the periods in which they operated at full capacity were selected, thus excluding long periods of inactivity due to emergencies.

As previously specified, in the training phase, the data coming from these 3 turbines were concatenated together to train different neural networks and evaluate the actual efficiency of the networks in the process of predicting the different output data of interest. The choice to concatenate the data from more turbines was conditioned by not having long periods available, so the data from a single turbine was insufficient to decently train the neural networks. Furthermore, this choice makes it possible to prevent neural networks from learning to perfectly recognize the functioning of a single turbine. Turbines of the same model may have slight differences: for this reason, taking data from more turbines allows learning features common to all turbines of the same model. The neural network obtained as a result of training can then be applied to other turbines for monitoring and fault detection. From the "Downtime Manager", section available in VireoXPower, it was observed that these turbines were not affected by downtime due to gearbox or generator malfunctions. In contrast, however, they had long periods of automatic shutdowns due to generic emergency states or shutdowns produced by the operators. It was therefore decided to exclude also data from these periods. In Table 4.1 are reported the names of the 3 turbines and the periods considered for data extraction:

These periods are the ones remaining after the data cleanup operation where the following emergencies occurred:

- WTG-6:

WTG_06	04/09/2021	01/12/2021
	20/12/2021	01/05/2022
WTG_07	04/09/2021	15/12/2021
	01/02/2022	01/03/2022
WTG_02	01/08/2021	30/10/2021
	01/12/2021	15/01/2022

Table 4.1. WTGs used in model elaboration phase (for ANNs training) and relative data periods used.

- Wind Turbine Emergency: 01/09/2021 - 01/09/2021, 08/12/2021 - 17/12/2021.
- WTG-7:
  - Wind Turbine Emergency: 01/09/2021 - 01/09/2021, 22/03/2022 - 22/03/2022
  - Button Emergency: 22/12/2021 - 27/01/2021.
- WTG-2:
  - Wind Turbine Emergency: 11/11/2021 - 11/11/2021, 25/11/2021 - 25/11/2021, 23/11/2021 - 24/11/2021, 07/03/2022 - 07/03/2022.
  - Button Emergency: 19/11/2021 - 19/11/2021.

The following types of data were extracted for each turbine, and their value represents the average over the 10-minute interval:

- Active Power.
- Ambient Temperature.
- Blade Pitch Angle.
- Gearbox Bearing Temperature.
- Gearbox Oil Temperature.
- Generator RPM.
- Generator Temperature.
- Wind Direction.
- Wind Speed.
- Rotor RPM.
- STD Wind.
- T-Status.

T-Status is a code that represents the state of the turbine over a given period. This information is not synchronized over the 10 minutes, so in the database is inserted an additional time information whenever there is a change of state. For confidentiality reasons, it is not possible to present the status codes, but they can be grouped into 4 main categories:

- WTG Produce.
- WTG Downtime (No Wind, Maintenance).
- WTG in Alarm or Fault.
- Error of communication.

### 4.2.1 Data Formatting

After extracting the data for all individual turbines, data formatting was performed. The columns of the dataframe and data values were reformatted, the columns were renamed, the time intervals to be excluded were removed, and rows related to state change were removed, associating, instead, the information of the current T-Status with the individual samples. Table 4.2 and 4.3 reports an example of data cleaning for WTG 2: the former reports the original dataset, while the latter reports the dataset after the cleaning operations.

	Date	SSESM.SM.WPP.Linea 02.SM02.DMAV.Active Power [kW]	Flag	SSESM.SM.WPP.Linea 02.SM02.DMAV.Ambient Temp [C]	Flag.1	SSESM.SM.WPP.Linea 02.SM02.DMAV.Blade Pitch Angle [degs]	Flag.2	SSESM.SM.WPP.Linea 02.SM02.DMAV.Wind Speed [m/s]	Flag.11	SSESM.SM.WPP.Linea 02.SM02.DMIMAX.Rotor RPM [rpm]	Flag.12	SSESM.SM.WPP.Linea 02.SM02.DMSD.Wind Speed [m/s]	Flag.13	SSESM.SM.WPP.Linea 02.SM02.ST.WTG Status [ ]	Flag.14	Unnamed: 31
0	2021/07/31 11:59:09.536	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-
1	2021/08/01 00:10:00	745.70245361328		27.275207519531		1.700409412384		8.2578544616699	16.098749160767		0.66969513893127		-	-	-	-
2	2021/08/01 00:20:00	903.30462646484		27.382696151733		1.6987500190735		8.658989906311	16.597499847412		0.37264826893807		-	-	-	-
3	2021/08/01 00:30:00	972.51745605469		27.379817962646		NV		8.7866563796997	16.698749542236		0.42674195766449		-	-	-	-
4	2021/08/01 00:40:00	926.98883056641		27.40470123291		NV		8.6459197998047	16.597499847412		0.31432789564133		-	-	-	-
5	2021/08/01 00:50:00	860.17517089844		27.476119995117		NV		8.4340257644653	16.297500610352		0.4170914888382		-	-	-	-
6	2021/08/01 01:00:00	816.79821777344		27.480195999146		NV		8.3149394989014	16.098749160767		0.3295322060585		-	-	-	-
7	2021/08/01 01:10:00	656.43115234375		27.532447814941		1.6990875005722		7.8419189453125	15.397500038147		0.52493494749069		-	-	-	-
8	2021/08/01 01:20:00	478.52139282227		27.882682800293		NV		7.1160516738892	13.698749542236		0.41140720248222		-	-	-	-
9	2021/08/01 01:30:00	540.34246826172		28.227643966675		1.8438007831573		7.3224692344666	14.497500419617		0.42620924115181		-	-	-	-
10	2021/08/01 01:40:00	695.32458496094		27.919244766235		1.6987500190735		7.930157661438	15.798749923706		0.32141709327698		-	-	-	-
11	2021/08/01 01:50:00	754.49967792969		27.608959197998		1.7005780935288		8.1712265014648	NV		0.55347567796707		-	-	-	-
12	2021/08/01 02:00:00	821.30413818359		27.080183029175		1.7032499313355		8.2312746047974	16.5		0.69215446710587		-	-	-	-
13	2021/08/01 02:10:00	924.40435791016		26.727304458618		1.7051999568939		8.758339881897	16.698749542236		0.81726515293121		-	-	-	-
14	2021/08/01 02:20:00	NV		NV		NV		8.980073928833	NV		0.85306996107101		-	-	-	-
15	2021/08/01 02:30:00	916.27697753906		26.400299072266		1.7145555019379		8.6384010314941	16.698749542236		0.92044937610626		-	-	-	-
16	2021/08/01 02:40:00	983.37316894531		26.147018432617		1.6999125480652		9.0023603439331	NV		0.72267061471939		-	-	-	-
17	2021/08/01 02:50:00	1125.939453125		25.787708282471		1.7013750076294		9.4748811721802	NV		0.70497590303421		-	-	-	-

Table 4.2. WTG 2: Original data set.

Date	WTG_02_Active_Power	WTG_02_Ambient_Temp	WTG_02_Blade_Pitch_Angle	WTG_02_Gearbox_Bearing_Temp	WTG_02_Gearbox_Oil_Temp	WTG_02_Generator_RPM	WTG_02_Generator_Temp	WTG_02_Wind_Dir	WTG_02_Wind_Speed	WTG_02_Rotor_RPM	WTG_02_STD_Wind	WTG_02_WTG_Status
2021-08-01 00:10:00	745.702454	27.275208	1.700409	72.846901	67.024529	1545.635864	50.447357	64.284225	8.257854	16.098749	0.669695	2.000000
2021-08-01 00:20:00	903.304626	27.382696	1.698750	73.596794	67.804062	1644.362915	50.630676	67.102875	8.658990	16.597500	0.372648	2.000000
2021-08-01 00:30:00	972.517456	27.379818		74.379807	68.635666	1675.802002	51.052578	67.641457	8.786656	16.698750	0.426742	2.000000
2021-08-01 00:40:00	926.988831	27.404701		74.827515	69.103333	1654.004639	51.580418	68.477455	8.645920	16.597500	0.314328	2.000000
2021-08-01 00:50:00	860.175171	27.476120		74.928558	69.256081	1615.758057	52.051991	67.956062	8.434026	16.297501	0.417091	2.000000
2021-08-01 01:00:00	816.798218	27.480196		74.917229	69.311264	1587.907959	52.392345	67.746750	8.314939	16.098749	0.329532	2.000000
2021-08-01 01:10:00	656.431152	27.532448	1.699088	74.436691	68.886093	1471.162231	52.575783	63.934731	7.841919	15.397500	0.524935	2.000000
2021-08-01 01:20:00	478.521393	27.882683		73.345276	67.774559	1338.034424	52.545467	64.300339	7.116052	13.698750	0.411407	2.000000
2021-08-01 01:30:00	540.342468	28.227644	1.843801	73.024338	67.297554	1395.469849	52.411133	69.079048	7.322469	14.497500	0.426209	2.000000
2021-08-01 01:40:00	695.324585	27.919245	1.698750	73.363579	67.610146	1512.486572	52.293152	68.182571	7.930158	15.798750	0.321417	2.000000
2021-08-01 01:50:00	754.499878	27.608959	1.700578	73.807373	68.139816	1544.294800	52.209194	64.959358	8.171227		0.553476	2.000000
2021-08-01 02:00:00	821.304138	27.080183	1.703250	74.021599	68.427269	1592.525024	52.201916	66.658089	8.231275	16.500000	0.692154	2.000000
2021-08-01 02:10:00	924.404358	26.727304	1.705200	74.579636	68.988853	1638.902954	52.302460	71.964958	8.758340	16.698750	0.817265	2.000000
2021-08-01 02:20:00								72.696518	8.980074		0.853070	2.000000
2021-08-01 02:30:00	916.276978	26.400299	1.714556	75.372749	69.731293	1628.300293	52.648254	73.699417	8.638401	16.698750	0.920449	2.000000
2021-08-01 02:40:00	983.373169	26.147018	1.699913	75.267410	68.564377	1657.592529	52.695312	73.580254	9.002360		0.722671	2.000000
2021-08-01 02:50:00	1125.939453	25.787708	1.701375	74.815498	67.948296	1676.992554		70.096199	9.474881		0.704976	2.000000
2021-08-01 03:00:00								73.238907	10.675868		0.913808	2.000000
2021-08-01 03:10:00								72.856850	10.637619		0.699475	2.000000
2021-08-01 03:20:00	1653.353760	23.926706	1.854140	75.692093	68.937500	1679.978149	53.222046	75.259285	10.713897	16.698750	0.661013	2.000000

Table 4.3. WTG 2: Data set after the cleaning operations.

### 4.3 Data Preprocessing

After the data extraction operation, a filtering process is required to make the data adequate to represent the condition of the "healthy behavior" of the turbines. Initially, a data cleaning operation was required to filter non-valid samples, subsequently all downtime was eliminated to filter period where a turbine is in a production phase, and then the various outliers were analyzed. The presence of outliers is quite a dangerous situation and their presence in the data may affect the accuracy of the model, so a filtering operation is extremely useful.

This operation was performed in different steps. Starting from the data of each wind turbine, it was possible to represent the relative power curve. Wind turbine 6 will be used to show the filtering operations carried out on the different WTGs. its initial power curve is illustrated in Figure 4.1. It is possible to notice the presence of many outliers: after each filtering operation, the power curve will gradually tend to resemble the ideal power curve.

Furthermore, in Table 4.4 it is possible to see how the various filtering operations resulted in a decrease of valid samples at training and how the number of values at NULL decreased significantly.

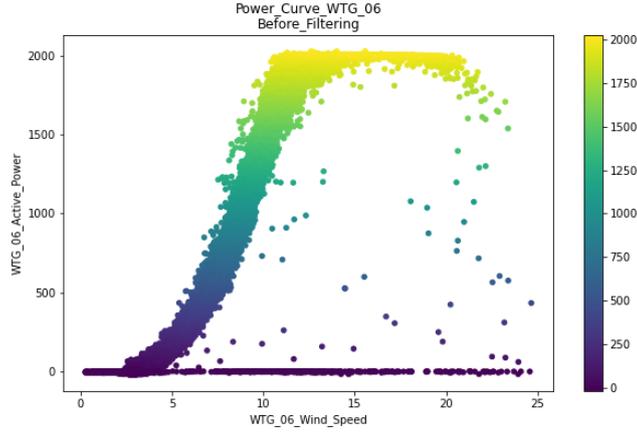


Figure 4.1. WTG 6: Power curve before filtering.

Dataframe	Samples	Null Samples
Dataframe Before Filtering	34519	17404
Dataframe Resampled	34519	7836
Dataframe Filt. Limitation	34387	7836
Dataframe Filt. Status	26961	0
Dataframe Filt. Neg. Power	26302	0
Dataframe Rem. Low Power	26232	0
Dataframe Filtered	24961	0

Table 4.4. WTG 6: Number of sample and null values for each filtering operation.

### 4.3.1 Data Cleaning

#### Elimination of NULL Samples and Resampling

If a condition of data loss occurs due to sensor transmission errors and at least one data item within all samples has a missing value, a NULL value is imposed on all samples for all data which is then deleted or replaced. If one or more samples have a NULL value for short period they are replaced with the previous valid sample. Resampling operation is performed for up to 12 consecutive samples ( $12 \times 10 = 120 \text{ min} = 2\text{h}$ ). For long periods where information is missing, the data are not subjected to this operation and the samples remain with all NULL values and are subsequently deleted. This operation does not generate any difference in the power curve of the wind turbine as the total of the NULL values concerns the sensor data on the turbine components, since a reference value for the power produced is always available. Nevertheless, the number of NULL samples is significantly reduced.

### 4.3.2 Downtime Filtering

#### Filtering Periods Of Limitations

The power limitations may be due to maintenance needs or constraints imposed by Terna, which is the manager of the electricity grid. For certain periods, turbines in an entire wind farm may need to run in depowered mode and feed less energy into the grid. During these periods the turbines assume a different behavior than expected: for this reason, the affected samples can be removed. This information is not encoded with a T-Status code and had to be extracted from an additional data set provided by the company.

Table 4.5 shows the restriction periods affecting the wind farm under consideration.

Date_Start	Date_End
01/09/2021 09:06	01/09/2021 14:40
19/12/2021 10:30	19/12/2021 14:40
07/02/2022 18:30	07/02/2022 19:20
01/04/2022 12:40	01/04/2022 13:10
01/04/2022 13:10	01/04/2022 14:45
01/04/2022 14:45	01/04/2022 19:25
03/04/2022 09:30	03/04/2022 14:00
03/04/2022 14:00	03/04/2022 15:15
03/04/2022 15:15	03/04/2022 16:15
21/04/2022 13:45	21/04/2022 14:10
21/04/2022 14:10	21/04/2022 16:25
21/04/2022 16:25	21/04/2022 16:45

Table 4.5. Restriction periods.

Figure 4.2 reports the power curve after this filtering operation.

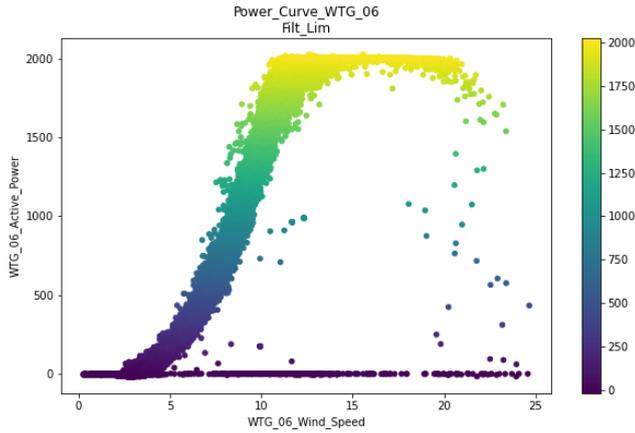


Figure 4.2. WTG 6: Power curve without limitation periods.

### Status Filtering

As specified earlier thanks to data cleaning operations, each sample is associated with a status code. Thus, the presence of the status information indicates when the turbine is producing correctly, and therefore all samples associated with a downtime category are eliminated. In addition, the presence of T-Status values attributable to an alarm, fault, or communication error is of sufficient worry and may compromise the definition of healthy behavior. As a precautionary measure, all samples preceding the 4 hours in which a problem of those mentioned above occurs are also eliminated, even if the turbine was producing. Thus, 24 samples prior to a sample with a different state than normal production, which can be defined as P, are eliminated ( $24 \times 10 = 240 \text{ min} = 4 \text{ h}$ ). This operation is schematically represented in Table 4.6. Instead, Figure 4.3 reports the power curve after filtering the status.

Although these operations are often sufficient to eliminate most of the outliers, further considerations can be made by analyzing more in detail the remaining values of the active power produced by the wind turbines. More specific techniques ensure that all outliers are eliminated so that the dataset input into the model will guarantee better performance.

Date	P1	P2	P3	P4	P5	Status
1/1/22 0.00	x	x	x	x	x	P
1/1/22 0.10	x	x	x	x	x	P
1/1/22 0.20	x	x	x	x	x	P
1/1/22 0.30	x	x	x	x	x	P
1/1/22 0.40	x	x	x	x	x	P
1/1/22 0.50	x	x	x	x	x	P
1/1/22 1.00	x	x	x	x	x	P
1/1/22 1.10	x	x	x	x	x	P
1/1/22 1.20	x	x	x	x	x	P
1/1/22 1.30	x	x	x	x	x	P
1/1/22 1.40	x	x	x	x	x	P
1/1/22 1.50	x	x	x	x	x	P
1/1/22 2.00	x	x	x	x	x	P
1/1/22 2.10	x	x	x	x	x	P
1/1/22 2.20	x	x	x	x	x	P
1/1/22 2.30	x	x	x	x	x	Alarm/Error
1/1/22 2.40	x	x	x	x	x	Alarm/Error
1/1/22 2.50	x	x	x	x	x	P
1/1/22 3.00	x	x	x	x	x	P
1/1/22 3.10	x	x	x	x	x	P

4 h

Table 4.6. Status Filtering: Operation of filtering based on status code value. All last 4h samples before an alarm, fault, or error are eliminated. The row with the red background represents the samples affected by an alarm or error, instead, the row with font red represents the samples eliminated.

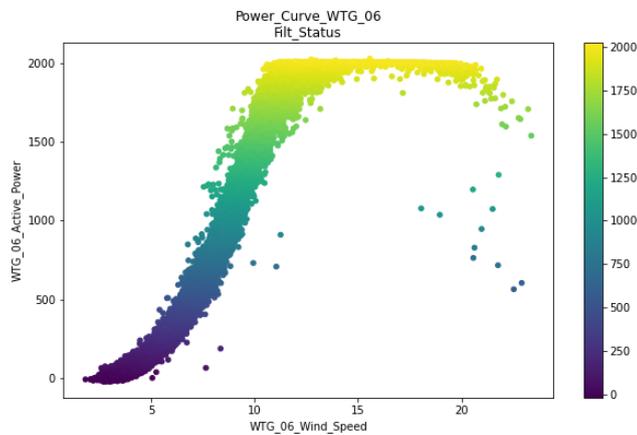


Figure 4.3. WTG 6: Power curve after filtering based on T-Status.

### Output Power Zero or Negative

After filtering based on production status the remaining samples where the turbine power is still zero or less than zero are eliminated. Under these conditions, the turbine does not produce. This operation also eliminates all samples where the wind speed is below the **cut-in-speed** limit, below which the rotor cannot rotate. Figure 4.4 reports the power curve after removing these values.

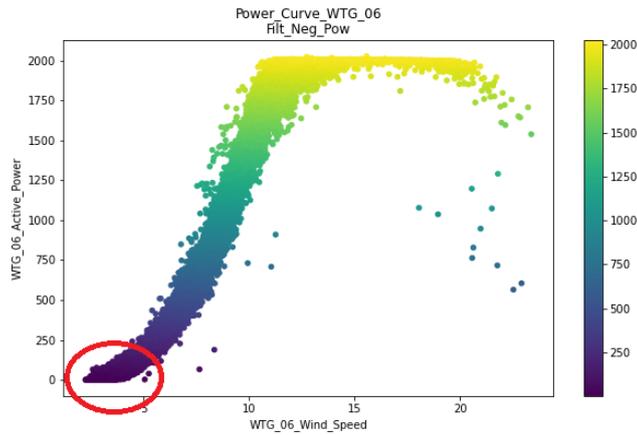


Figure 4.4. WTG 6: Power curve without negative or null power in output.

### 4.3.3 Outliers Filtering

#### Unrealistically Low Power

As can be seen in the lower right area of the power curves represented so far, some samples have a very low power output for very high wind speeds. This non-linear behavior is then removed employing a specific filtering operation. Figure 4.5 reports the power curve after filtering these unrealistic low power situations.

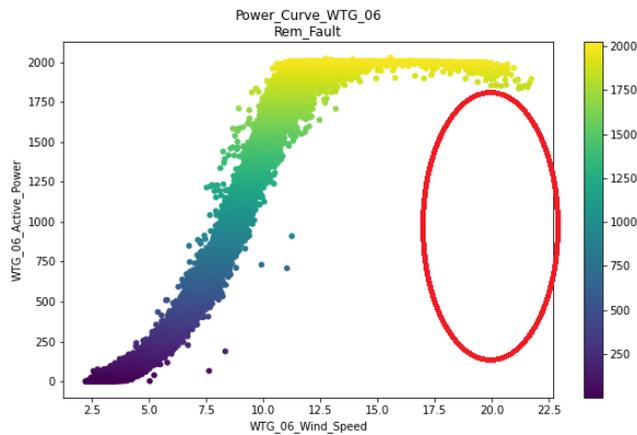


Figure 4.5. WTG 6: Power curve without unrealistic low power.

#### Data Smoothing

At this stage, filtering based on the "Binning method for data smoothing" was carried out. Starting from the solution proposed in the research paper "Wind energy analytics toolbox: Iterative power curve filter" [25], the code was adapted to make it functional for the turbines under consideration. In general, the data were divided into clusters based on wind speed.

Considering that the maximum wind speed corresponds to 25 m/s, it was decided to set an interval width of 0.5 m/s, obtaining a total of 51 bins. Then, for each bin, the power output values were taken into analysis and, for these, standard deviation and arithmetic mean were

calculated. Subsequently, with a reference to the control chart principle, a z-score value of 2.5 is set to calculate the limits value LCL and UCL for each bin, and as result power values above them were eliminated. So, UCL and LCL were fixed as follows:

- UCL:  $\mu + 2.5\sigma$
- LCL:  $\mu - 2.5\sigma$

Figure 4.6 shows the deleted data in orange, while Figure 4.7 reports the power curve after data smoothing.

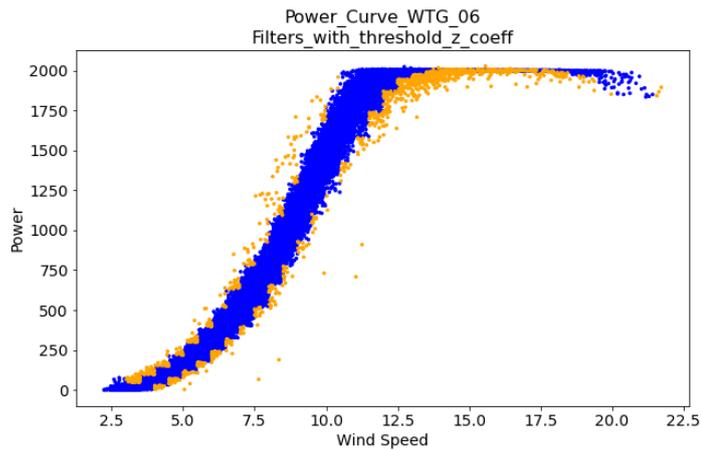


Figure 4.6. WTG 6: Result of Data smoothing filter on the Power Curve. In orange are represented the samples eliminated.

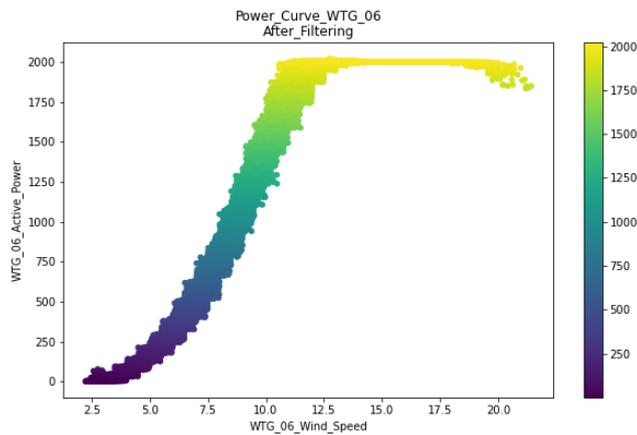


Figure 4.7. WTG 6: Power curve after data smoothing operation.

Analyzing Figures 4.8 and 4.9, which depict the correlation matrix between data before and after filtering, it can be seen that the relationship between them is stronger.

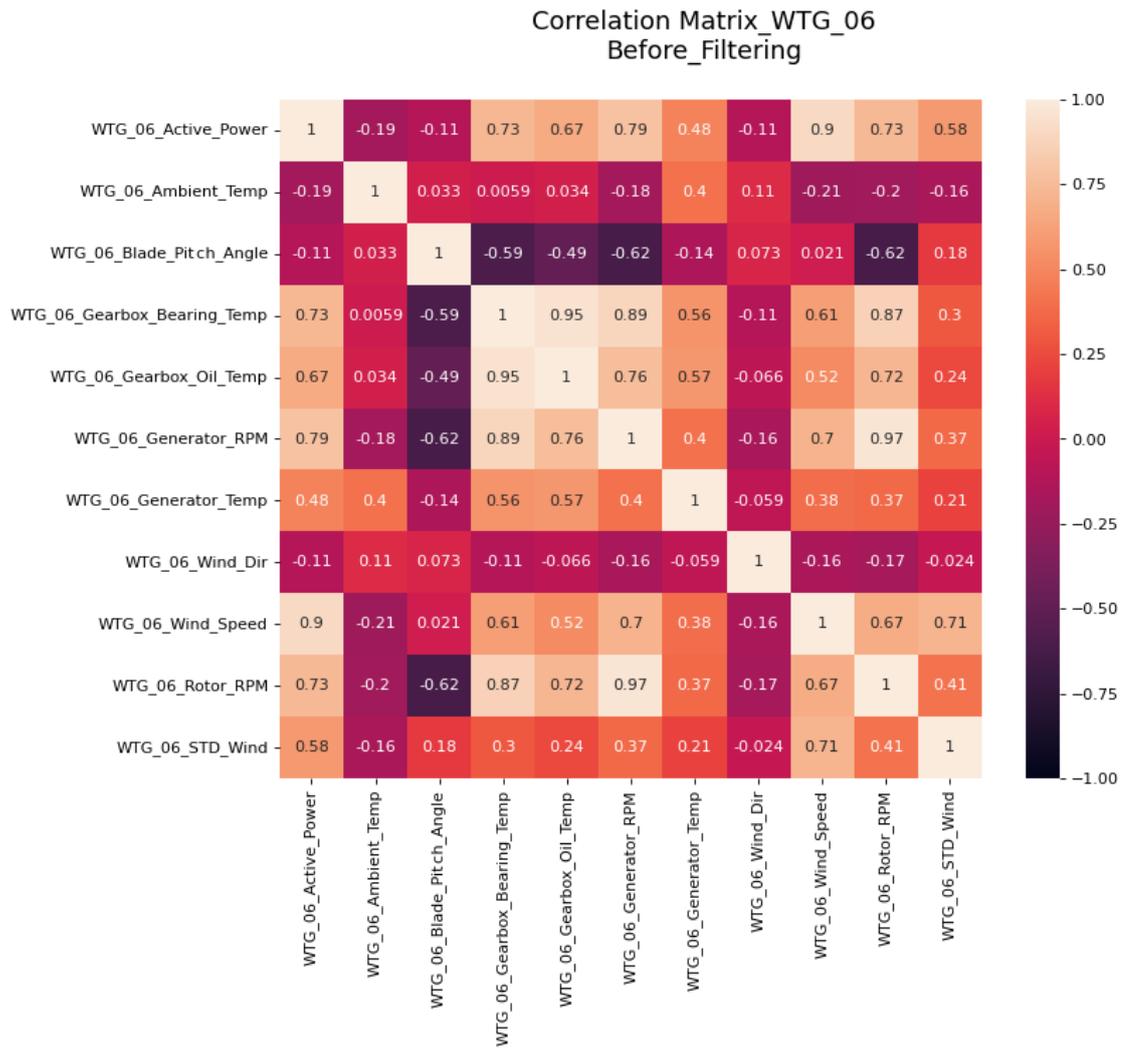


Figure 4.8. WTG 6: Correlation matrix of the dataset before filtering.

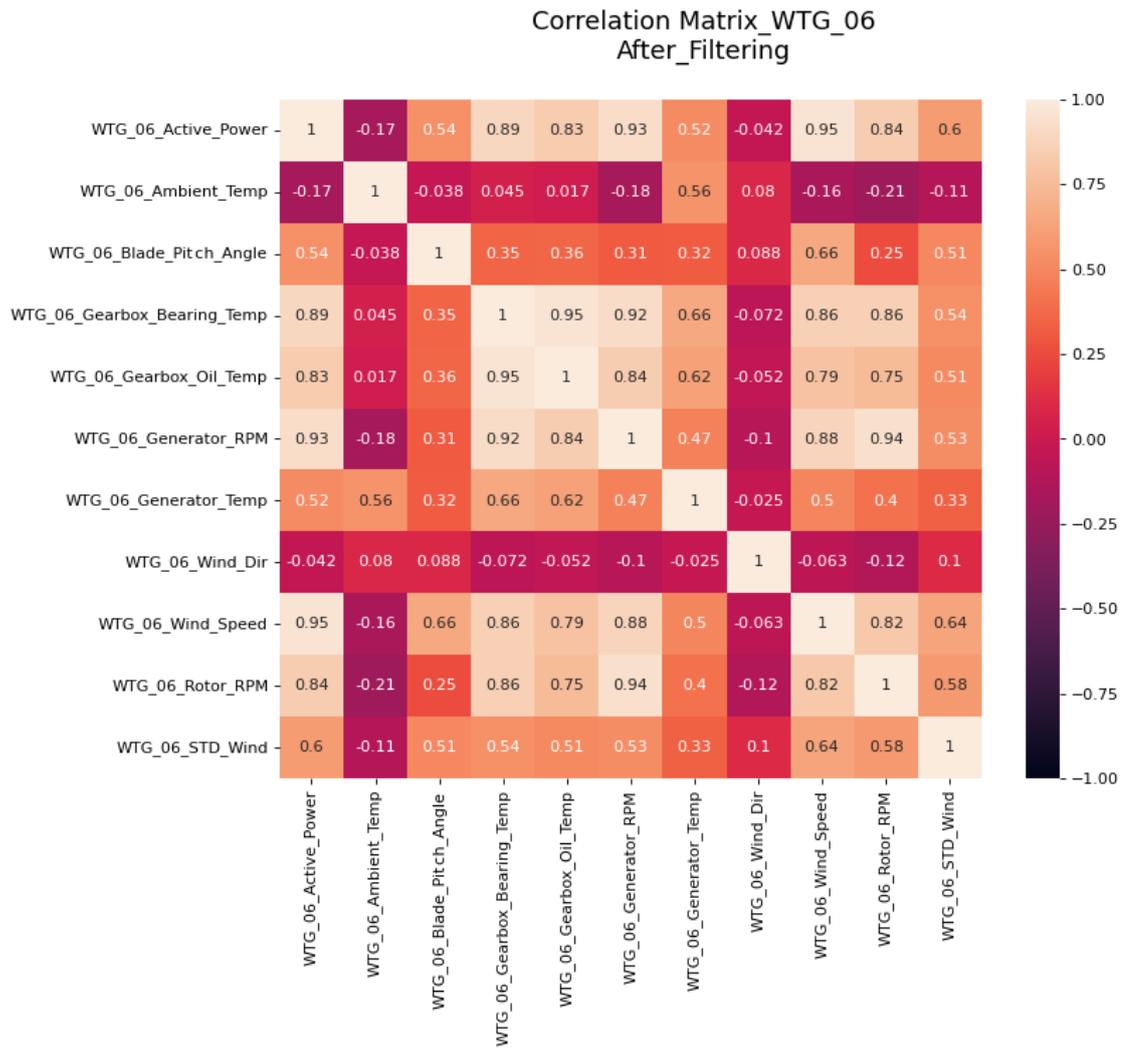


Figure 4.9. WTG 6: Correlation matrix of the dataset after filtering operation.

## 4.4 Wind Direction Analysis

Wind direction can play an important role in the analysis of output power. After the data preprocessing operation, the wind direction of each sample was analyzed in order to see if the different turbines of the wind farm work better when the wind comes from certain directions. In the dataset, the wind direction is provided in polar coordinates, thus as an angle expressed in degrees. A Cartesian coordinate system is a three-dimensional space based on three perpendicular axes and can be useful to represent the 3D power curve (wind speed, wind direction, and active power).

In Figure 4.10 it is possible to observe the difference in the power curve on the unfiltered and filtered data of WTG 6 turbine, with the outliers eliminated by the filtering operation. The graphs show that there are denser regions along the wind direction that contribute to maximum energy production and other directions where the wind speed remains limited.

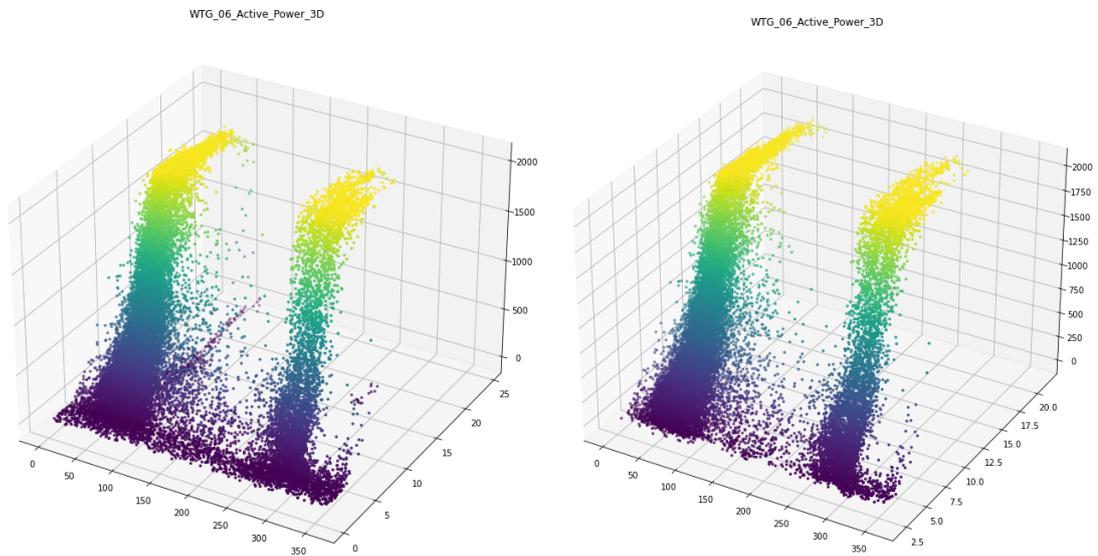


Figure 4.10. WTG 6, from left to right: 3D power curve before and after filtering.

Figure 4.10 (only for the filtered version), alongside Figure 4.11, represent the three Cartesian plots of the power output of WTG 2, WTG 6, and WTG 7, the 3 turbines used as inputs for the neural network training. It is also possible to see that these regions are different for each turbine.

As a result, in order to conduct a more in-depth analysis, it was decided to cluster the data based on the cardinal points of the wind direction. An image of the compass rose is shown in Figure 4.12. The transformation was carried out in accordance with the following rule, which is based precisely on the compass rose:

- if Wind Dir  $> 348.75^\circ$  or Wind Dir  $< 11.25^\circ$ : return N.
- if Wind Dir  $< 33.75^\circ$ : return NNE.
- if Wind Dir  $< 56.25^\circ$ : return NE.
- if Wind Dir  $< 78.75^\circ$  return ENE.
- if Wind Dir  $< 101.25^\circ$ : return E.
- if Wind Dir  $< 123.75^\circ$ : return ESE.
- if Wind Dir  $< 146.25^\circ$ : return SE.
- if Wind Dir  $< 168.75^\circ$ : return SSE.

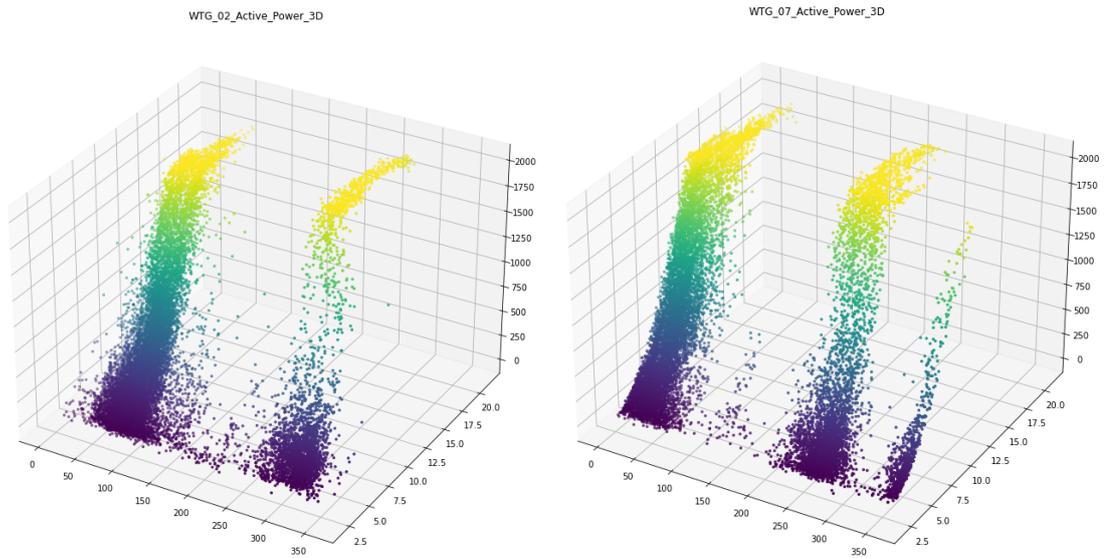


Figure 4.11. WTG 2 and WTG 7: 3D power curve after filtering.

- if Wind Dir < 191.25°: return S.
- if Wind Dir < 213.75°: return SSW.
- if Wind Dir < 236.25°: return SW.
- if Wind Dir < 258.75°: return WSW.
- if Wind Dir < 281.25°: return W.
- if Wind Dir < 303.75°: return WNW.
- if Wind Dir < 326.25°: return NW.
- else: return NNW.

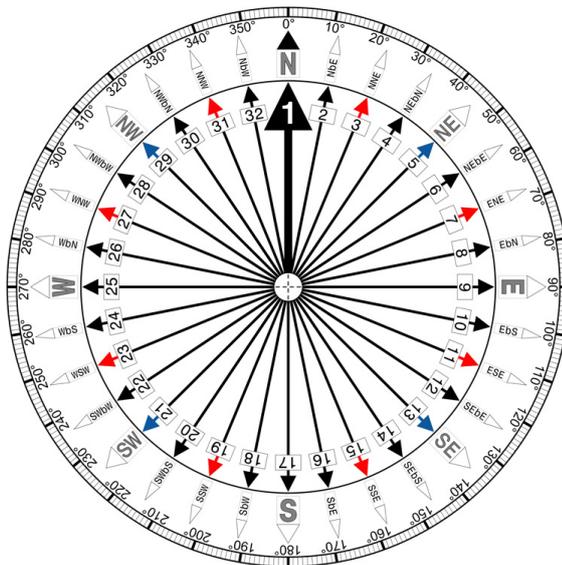


Figure 4.12. Compass rose [28].

Figure 4.13 shows a comparison of the power curves of WTG 6 and WTG 7 obtained when the wind comes from East-Northeast (ENE) and North-Northeast (NNE) directions. This analysis clarifies how some turbines from the same wind farm perform better in certain wind directions. Considering that the thesis model was designed to work on a generic turbine, this analysis is further confirmation that the merge of more data is the right choice to best represent a generic turbine.

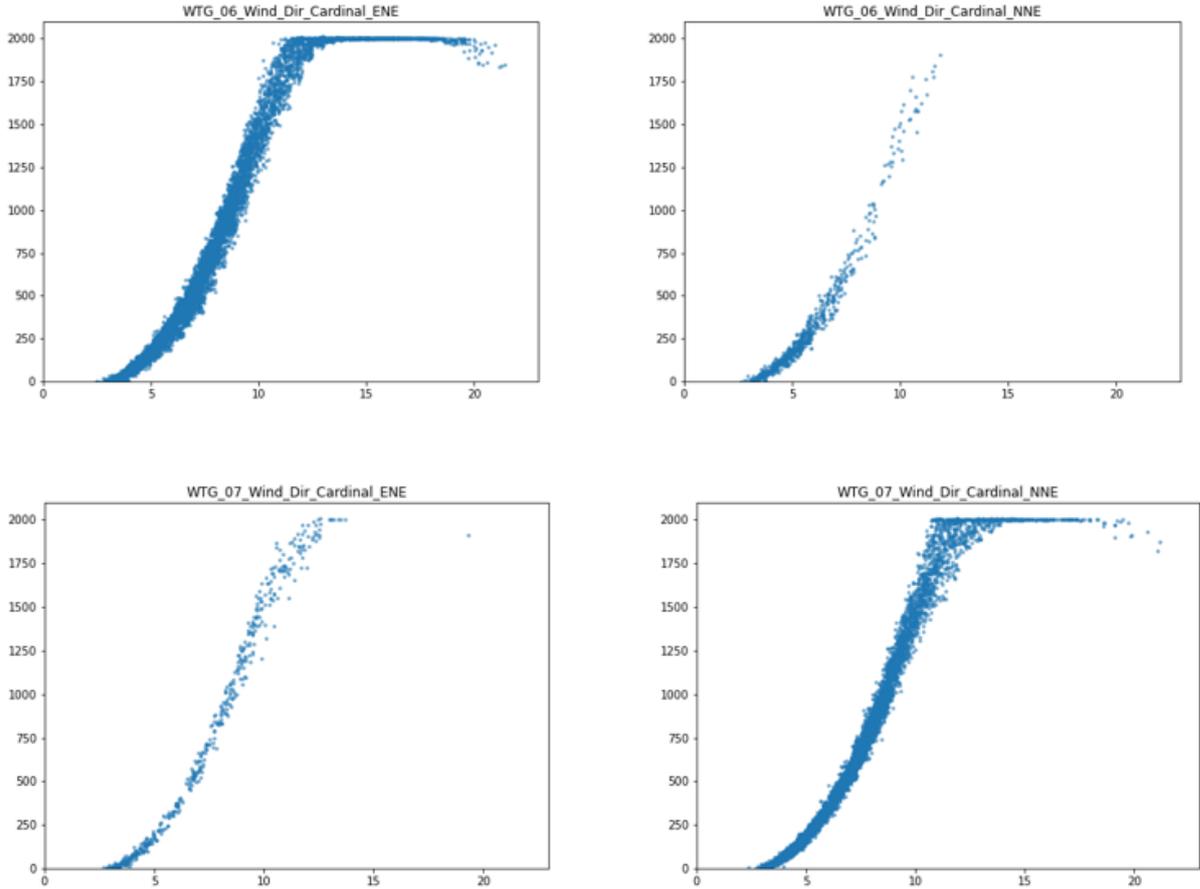


Figure 4.13. Comparison of the power curves of the WTG 6 and WTG 7 considering only the wind direction ENE and NNE.

Subsequently, starting from the wind speed and the angle in degrees representing its direction, the X and Y components of the wind vector were calculated for each sample of the dataset using the following formulas:

$$angle\ radians = angle\ degrees * \frac{\pi}{180}$$

$$X\ Component = Wind\ Speed * \cos(angle\ radians)$$

$$Y\ Component = Wind\ Speed * \sin(angle\ radians)$$

## 4.5 Models Processing

To simulate the behavior of a healthy state of a turbine several models were considered with the aim to predict different output variables.

Component	Output	Inputs
<b>Wind Turbine Models</b>		
Wind Turbine	Active Power	Wind Speed
		Ambient Temperature Wind Speed STD Wind Speed Wind Direction
		Ambient Temperature STD Wind Speed Wind X Component Wind Y Component
<b>Gearbox Oil Models</b>		
Gear Box	GearBox Oil Temperature	Active Power Ambient Temperature Rotor RPM
		Active Power Ambient Temperature Rotor RPM Wind Speed Blade Pitch Angle
		Active Power Ambient Temperature Rotor RPM Wind Speed Blade Pitch Angle Gear Box Bearing Temperature Generator RPM Generator Temperature
<b>Gearbox Bearing Models</b>		
Gear Box Bearing	Gear Box Bearing Temperature	Active Power Ambient Temperature Rotor RPM GearBox Oil Temperature
		Active Power Ambient Temperature Rotor RPM GearBox Oil Temperature Wind Speed Blade Pitch Angle Generator RPM Generator Temperature
<b>Generator Models</b>		
Generator	Generator Temperature	Active Power Ambient Temperature Wind Speed Rotor RPM Generator RPM GearBox Oil Temperature Gear Box Bearing Temperature

Table 4.7. Inputs and outputs used for the different models, divided for categories.

#### 4.5.1 Features Selection

The variables must be chosen to ensure that the healthy behavior of the system can be clearly illustrated. Its ability to monitor components and identify faults will be strongly influenced by

inputs and outputs.

The right choice can enable proper recognition of abnormal behaviors, but they are not easy to determine. The assistance of literature research is essential to understand which of the available variables, among those handled by SCADA systems, are most suitable for implementing a wind turbine monitoring system by monitoring its components, mainly by focusing on the most common types of faults. The identification of the link between variables and faults is generally based on engineering knowledge. An important role in the choice of output variables is also due to the types of stops that occurred in the test turbines.

Once the output variable for each model was chosen, different combinations of input variables were tested to best predict the output. They were based not only on literature but also on the intuition derived from the autonomous study of the functioning of wind turbines. The different inputs and outputs for the different proposed models are displayed in Table 4.7.

For the choice of inputs and outputs, an attempt was made to adapt the following studies [55] [56] to the case under consideration. In particular, the choice of using active power as input for the prediction of component temperatures is due to the fact that the temperature inside the nacelle is directly proportional to the power produced. Furthermore, the oil flow on the components depends on the rotor speed and for this reason, it has also been included as an input parameter [54].

#### 4.5.2 Datasets

After feature selection, different datasets for each turbine were created with the right variables of interest. As specified in Chapter 3 two types of ANNs have been implemented: FNN and LSTM. In FNNs, the prediction of the output at instant  $t$  depends on the value of input data at the same instant  $t$ , while for LSTM networks timeseries were created based on the rolling window principle: the value of output at instant  $t$  depends on the value of each input in the previous 2 hours. Two different approaches were considered for the realization of the timeseries needed to train the various LSTMs and predict the output value:

- In the first case, called  $L1$ , were considered all inputs from the previous two hours ( $t-12$  to  $t$ , sampled every 10 minutes) up to the last measurement. Total timestamps = 13.
- In the second case, called  $L2$ , the output was considered in addition to all inputs from the previous two hours up to the measurement preceding the last ( $t-12$  to  $t-1$ ). In other words, in order to predict the value of the output at instant  $t$  its previous measurements were also taken into account. Total timestamps = 12.

Figure 4.14 shows the creation process of each window considering a generic timestamp  $t$ .

Since the process of creating the single window involves having data available for 2 consecutive hours without interruption, this will reduce the quantity of data available in the datasets. The entire window is deleted even if a single timestamp has NULL values. For the training and testing of the models, it was necessary to divide the remaining data between training, validation, and test. In this regard, it was chosen to adopt a 80% -10% -10% subdivision.

The training and validation datasets of the 3 turbines are concatenated and used to train the models, while the test sets remain independent and are used after the training is complete to check the performance of the models.

A diagram of this operation is displayed in Figure 4.15.

In Figure 4.16 it is possible to notice the value of the Active Power output variable for WTG 6 and, based on the color, understand in which set each timestamp is inserted. For recurring neural networks, the division takes place after the rolling windows creation process, which involves the elimination of some output values. Therefore, the subsequent division will be different. Figure 4.17 shows the division of the same output variable for prevision based on the timeseries forecasting principle.

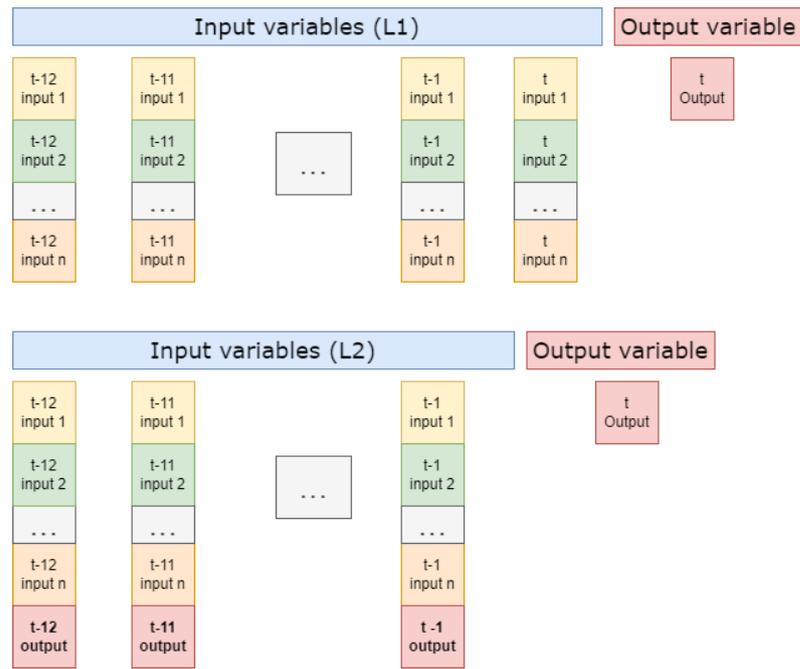


Figure 4.14. The 2 different approach (L1 and L2) for creating the rolling window.

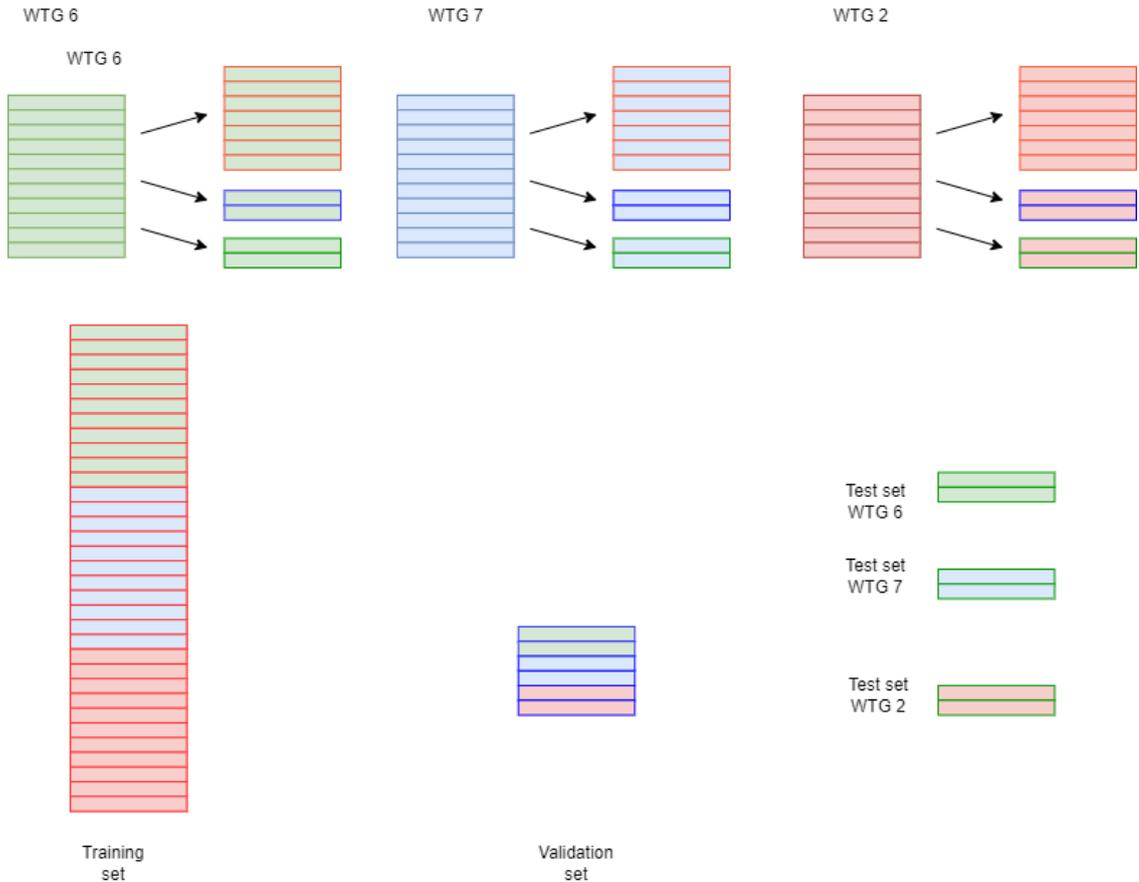


Figure 4.15. Phases of creation of training, validation, and test data sets.

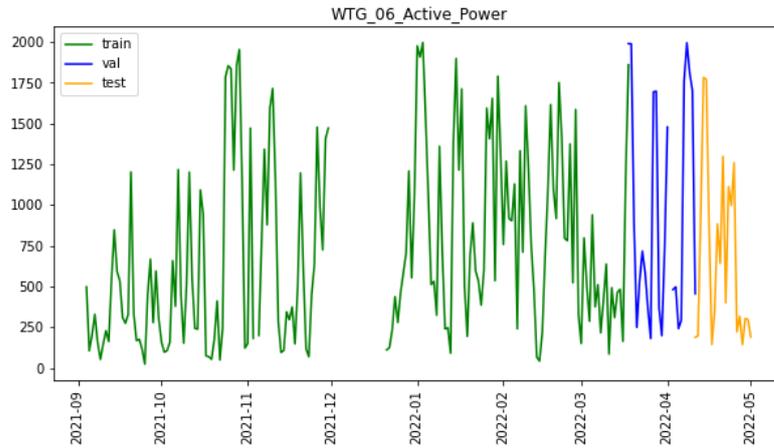


Figure 4.16. Active power output variable for WTG 6 and division of timestamp values in training (green), validation (blue), and test dataset (orange).

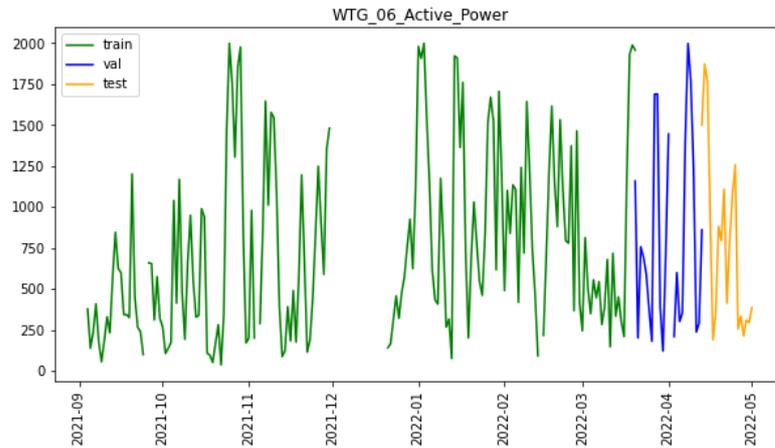


Figure 4.17. Active power output variable for WTG 6 and division of timestamps value in the training set (green), validation set (blue) and test set (orange) for timeseries forecasting approach.

Furthermore, Table 4.8 shows for each turbine the number of timestamps available in each dataset and the number of samples after the concatenation process in the training and validation dataset.

Since machine learning algorithms work best with scaled data, in addition to training the various neural networks with the original data, it was probed the use of techniques such as:

- Scaling to a range 0-1 (Scikit-learn object `MinMaxScaler`).
- Z-score standardization (Scikit-learn object `StandardScaler`).

### 4.5.3 Artificial Neural Networks Architectures

This section discusses the architectures of the different ANNs developed, which, as already mentioned, are part of the categories FNN and LSTM Multivariate with multiple input time series. In the regression problem loss function, MSE is the default choice and is mainly used after removing the outliers [53].

Number of samples in dataset for prediction based on the same timestamp				
	Training set	Validation set	Test set	Total samples turbine
WTG 6	19968	2496	2497	24961
WTG 7	11779	1472	1473	14724
WTG 2	12116	1514	1515	15145
Total	43863	5482		

Number of samples in dataset for prediction based on timeserie				
	Training set	Validation set	Test set	Total samples turbine
WTG 6	13602	1700	1701	17003
WTG 7	7652	956	958	9566
WTG 2	7947	993	994	9934
Total	29201	3649		

Table 4.8. Number of samples in training, validation, and test sets for each turbine, for both the two approaches of prediction.

### Feed Forward Neural Networks Architecture

As for the FNN typology, two types of neural networks have been trained. These can be considered like two different configurations (ANN Models) of Multilayer Perceptron:

- First configuration:
  - 1 input layer with a number of neurons equal to the number of input variables.
  - 1 hidden layer with 50 neurons.
  - 1 output layer with 1 neuron since we need to predict only one output variable.
- Second configuration:
  - 1 input layer with a number of neurons equal to the number of input variables.
  - 1 hidden layer with 50 neurons.
  - 1 hidden layer with 18 neurons.
  - 1 output layer with 1 neuron since we need to predict only one output variable.

In these solutions, ANNs receive input data of various variables at instant  $t$  and predict as output the value of an additional variable at the same time  $t$ . Figures 4.18 and 4.19 show two examples of the architecture of the two networks assuming 5 variables as input.

The models are composed of all interconnected neurons with RELU [51] activation function. The following hyperparameters were set:

- Loss function: MSE.
- Metrics: RMSE,  $R^2$ .
- Hyper-parameters:
  - Learning rate: 0.0005, 0.005.
  - Epochs: 100.
  - Batch size: 128, 64, 32.

As it is possible to see here above, neural networks were trained with different learning rates and batch sizes. The callback parameters were set as follows:

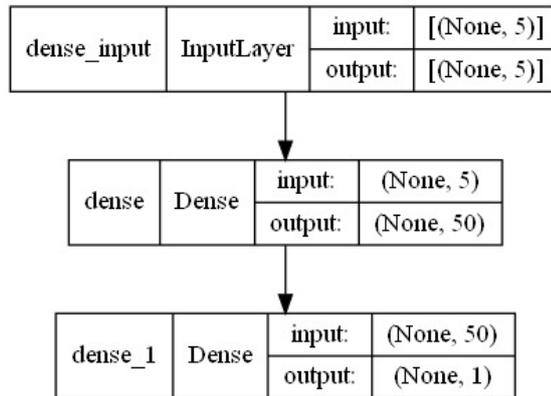


Figure 4.18. ANN Model of MLP Architecture with an input layer with 5 neurons and 1 hidden layer of 50 neurons.

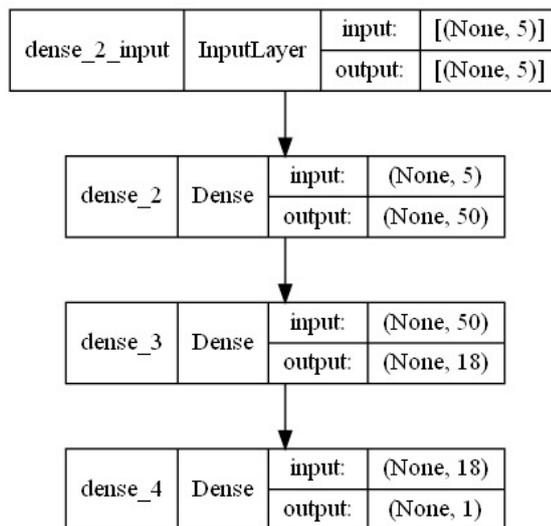


Figure 4.19. ANN Model of MLP Architecture with an input layer with 5 neurons and 2 hidden layers of 50 and 18 neurons.

- Early Stopping:
  - Parameter: MSE validation loss.
  - Patience: 20 epochs.
  - Monitoring mode: reduce value.
- ReduceLROnPlateau:
  - Parameter: loss function on the validation set.
  - Patience: 4 epochs.
  - Factor: 0.8.
  - Minimum learning rate: 0.01 of initial learning rate.

### Long Short Term Memory Architecture

LSTMs are a category of neural networks that lend themselves well to time series, but they require that the values of the input variables be passed to the machine in the right format:

3D vector with shapes (no of samples, timesteps, features)

where features are the number of variables of input and timesteps are the number of observations for each variable. A single timestep can be defined as a time interval between two timestamps. Also in this case, two LSTM neural networks having different configurations were implemented:

- Single LSTM:
  - 1 input layer with a number of cells equal to the number of timesteps for the number of input variables.
  - 1 hidden layer with 64 cells.
  - 1 output layer with 1 neuron since we need to predict only one output variable.
- Stacked LSTM (4 hidden layers):
  - 1 input layer with a number of cells equal to the number of timesteps for the number of input variables.
  - 1 hidden layer with 64 cells.
  - 1 hidden layer with 50 cells.
  - 1 hidden layer with 25 cells.
  - 1 hidden layer with 1 cell.
  - 1 output layer with 1 neuron since we need to predict only one output variable.

Figures 4.20 and 4.21 show two examples of the architecture of the two networks assuming 5 variables as input and 13 timesteps.

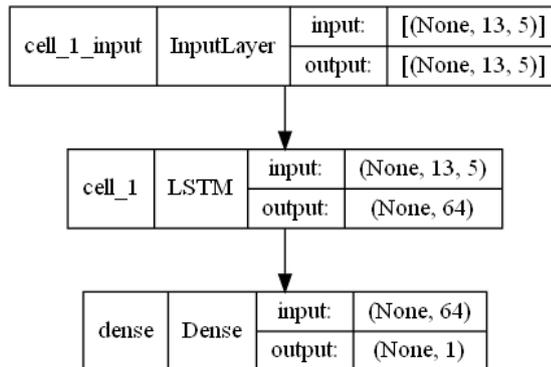


Figure 4.20. ANN Model of LSTM Architecture with input shape [5, 13].

All the parameters, even the hyperparameters and callback parameters, used during the training of the machines are the same as the FNN models. The only difference concerns the patience value of Early Stopping that is 4 epochs.

#### 4.5.4 Training and Test

This section analyzes the results of the training process for each model, displaying the results. All 4 types of ANNs were trained for each model of all categories, for each combination of inputs associated with it, and for each type of scaling (no scaling, minmax, and standard) of the data, using also different combinations of Batch size and Learning Rate.

After the training of all the different networks, it was necessary to test their performance on all 3 test sets of the 3 turbines used in this phase. The performance is measured through the RSME and R2 metrics for each individual turbine. Then an average of the RMSE values is calculated and the ANN model with the lowest average, for each category, will be selected as the best.

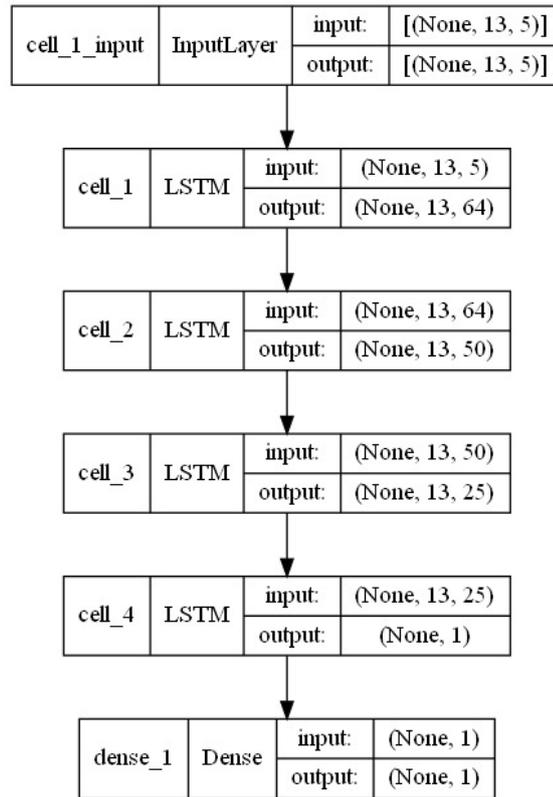


Figure 4.21. ANN Model of 4 Stacked LSTM Architecture with input shape [5, 13].

Below the best ANN for each category is presented with the relative loss graph and metrics graphs on both training and validation sets. In all the examples, the loss function and the validation loss function tend to zero together as the number of epochs increases. This behavior indicates that the training is done well and there are no overfitting or underfitting problems. Furthermore, the RMSE functions also tend to zero, so the average error on the forecast tends to decrease, and the values of  $R^2$  tend to 1, which represents the optimal value. The best models for each category are summarized below.

### Wind Turbine Model

Result training parameters:

- Type of network: FNN 2 Layers.
- Numbers of hidden layers and node for each: 2 (50 18).
- Inputs: Ambient Temperature, Wind Speed, STD Wind Speed, Wind Direction.
- Scaling: MinMax.
- Minimum validation loss: 0.00055 at epoch 1.
- Epochs of training: 37.
- Min validation RMSE: 0.01630.
- Max validation  $R^2$ : 0.99399.
- Batch size: 64.

- Starting Learning rate: 0.0005.
- Learning Rate Reducer: Monitoring validation loss Min: 0.00010.

Figure 4.22 shows the graphs of loss functions MSE, instead Figures 4.23 and 4.24 show RMSE and  $R^2$  metrics functions. Table 4.9 shows the results of the testing phase for the best model. The average RMSE is about 50 watts. Figure 4.25 compares the real value and the predicted value of the output for the WTG 6 test set. Instead, Figures 4.26 and 4.27 represent the power curves real and predicted, for the same turbine.

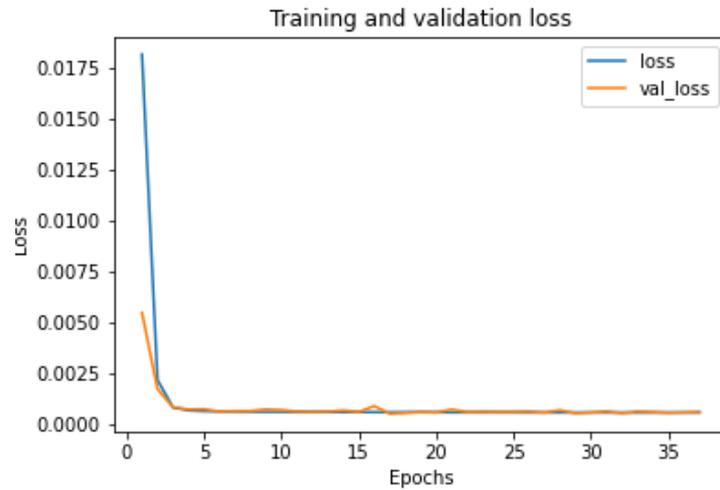


Figure 4.22. Loss of the best network for the wind turbine model.

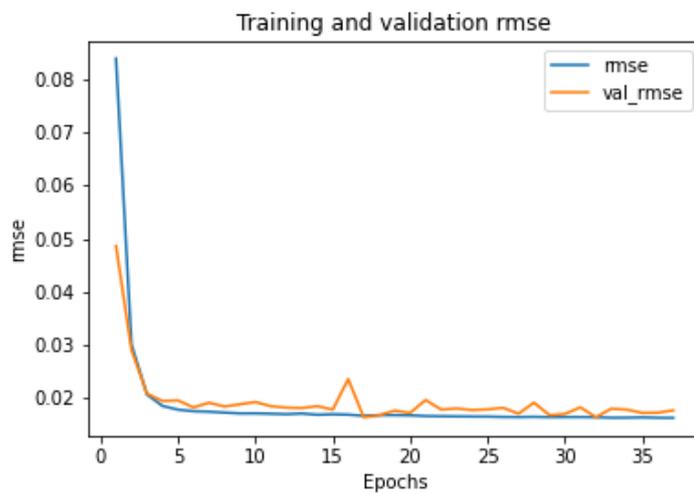


Figure 4.23. RMSE metric of the best ANN of the wind turbine model.

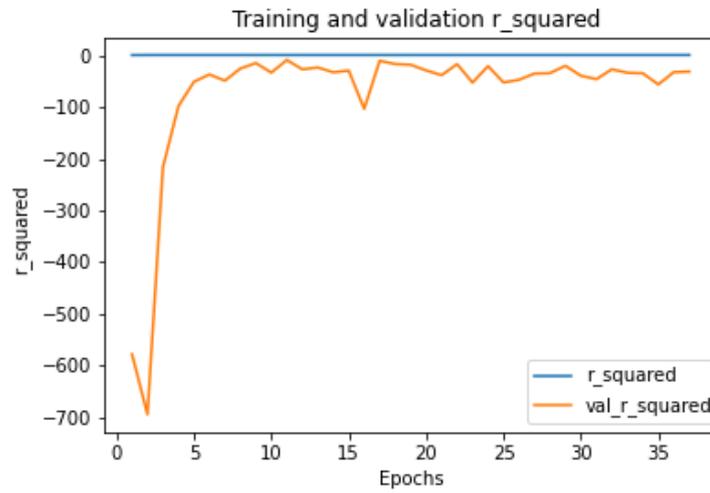


Figure 4.24.  $R^2$  metric of the best ANN of the wind turbine model.

Output: Active_Power							
WTG 6		WTG 7		WTG 2		Avg RMSE	Avg R squared
RMSE	R square	RMSE	R square	RMSE	R square		
46,523	0,994	59,347	0,992	46,177	0,996	50,683	0,994

Table 4.9. Metrics evaluation on the 3 different test sets for the wind turbine model.

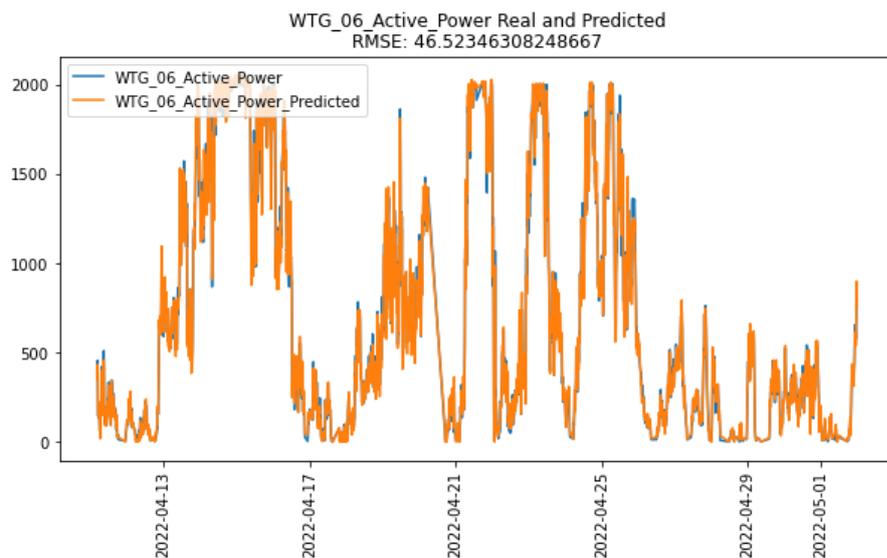


Figure 4.25. Real and predicted value of active power for the WTG 6 test set.

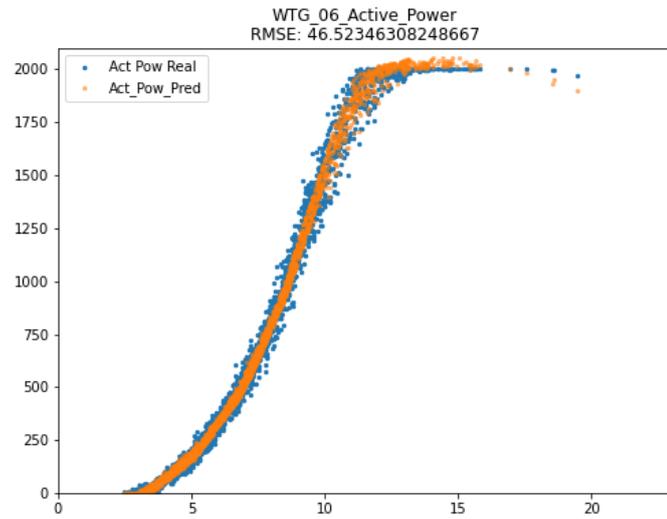


Figure 4.26. Power curves real and predicted of the test set of the WTG 6 turbine.

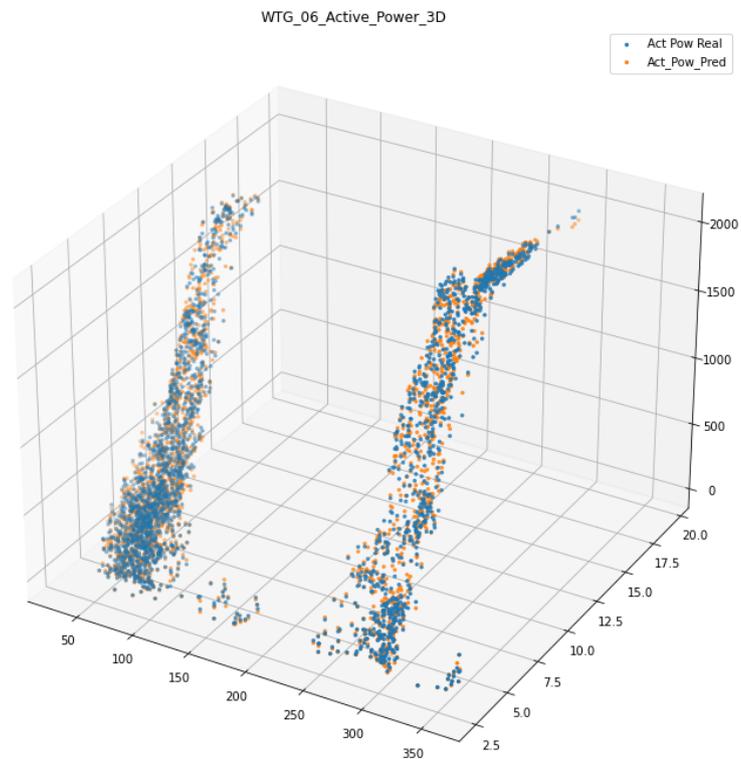


Figure 4.27. Power curves 3D (with Wind Direction) real and predicted of the test set of the WTG 6 turbine.

## Gearbox Oil Model

The training parameters used were the following:

- Type of network: Single LSTM.
- Numbers of hidden layers and node for each: 1(64).
- LSTM approach: L1.
- Inputs: Active Power, Ambient Temperature, Rotor RPM, Wind Speed, Blade Pitch Angle, Gear Box Bearing Temperature, Generator RPM, Generator Temperature.
- Scaling: Standard.
- Minimum validation loss: 0.02238 at epoch 10.
- Epochs of training: 10.
- Min validation RMSE: 0.08895.
- Max validation  $R^2$ : 0.98632
- Batch size: 64.
- Starting Learning rate: 0.0005.
- Learning Rate Reducer: Monitoring validation loss Min: 0.00050.

Figure 4.28 shows the graphs of loss functions MSE, instead Figures 4.29 and 4.30 show the metrics functions RMSE and  $R^2$ . Table 4.10 shows the results of the testing phase for the best model. The average RMSE is about 0.47 degrees Celsius. In Figure 4.31 the real value and the predicted value of the output are compared, for the WTG 6 test set.

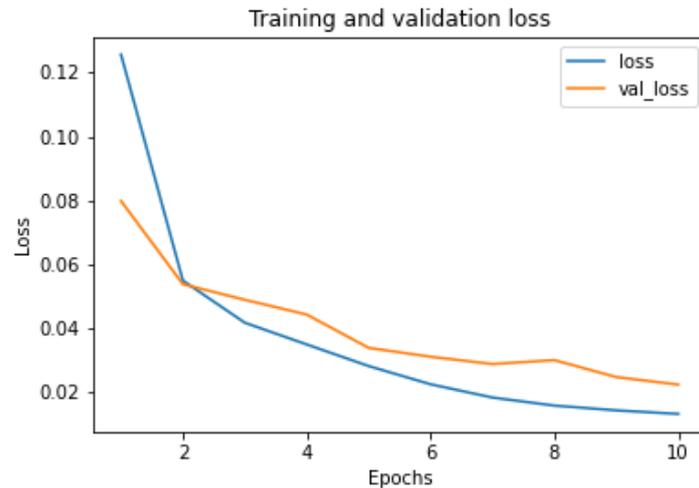


Figure 4.28. Loss of the best network for the gearbox oil model.

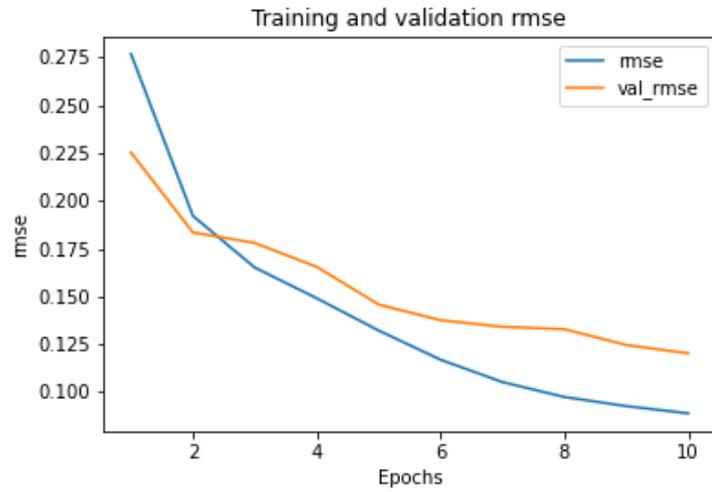


Figure 4.29. RMSE metric of the best ANN of the gearbox oil model.

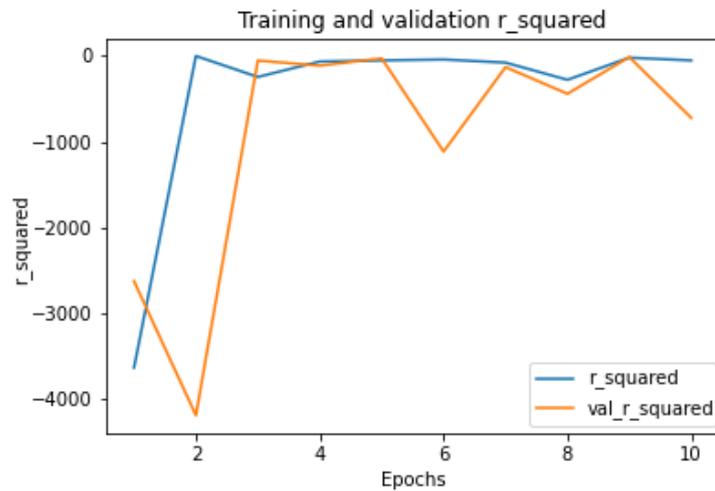


Figure 4.30.  $R^2$  metric of the best ANN of the gearbox oil model.

Output: Gearbox Oil Temperature							
WTG 6		WTG 7		WTG 2		Avg RMSE	Avg R squared
RMSE	R square	RMSE	R square	RMSE	R square		
0,362	0,985	0,396	0,981	0,655	0,934	0,47	0,968

Table 4.10. Metrics evaluation on the 3 different test sets for the gearbox oil model.

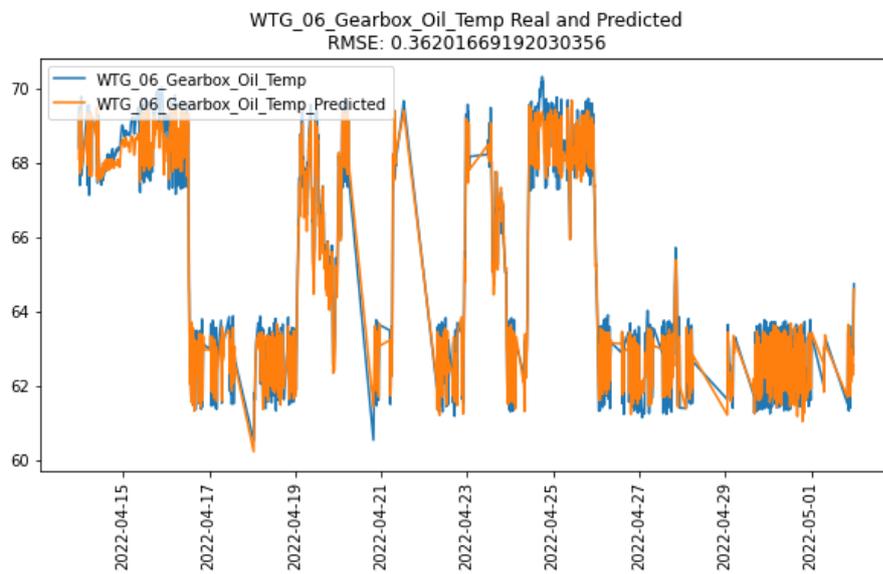


Figure 4.31. Real and predicted value of gearbox oil temperature for the WTG 6 test set.

## Gearbox Bearing Model

The training parameters used were the following:

- Type of network: Single LSTM.
- Numbers of hidden layers and node for each: 1(64).
- LSTM approach: L2.
- Inputs: Active Power, Ambient Temperature, Rotor RPM, Gearbox Oil Temperature, Wind Speed, Blade Pitch Angle, Generator RPM, Generator Temperature.
- Scaling: Standard.
- Minimum validation loss: 0.01345 at epoch 10.
- Epochs of training: 10.
- Min validation RMSE: 0.08957.
- Max validation  $R^2$ : 0.98468
- Batch size: 64.
- Starting Learning rate: 0.0005.
- Learning Rate Reducer: Monitoring validation loss Min: 0.00050.

Figure 4.32 shows the graphs of loss functions MSE, instead Figures 4.33 and 4.34 show the metrics functions RMSE and  $R^2$ . Table 4.11 shows the results of the testing phase for the best model. The average RMSE is about 0.49 degrees Celsius. In Figure 4.35 the real value and the predicted value of the output are compared for the WTG 6 test set.

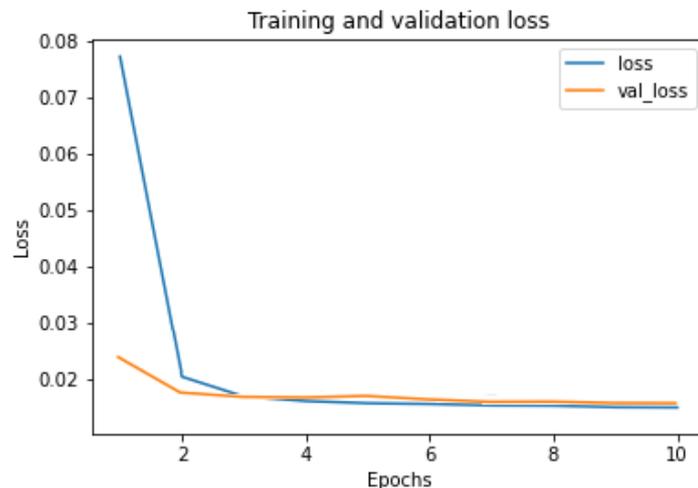


Figure 4.32. Loss of the best network for the gearbox bearing model.

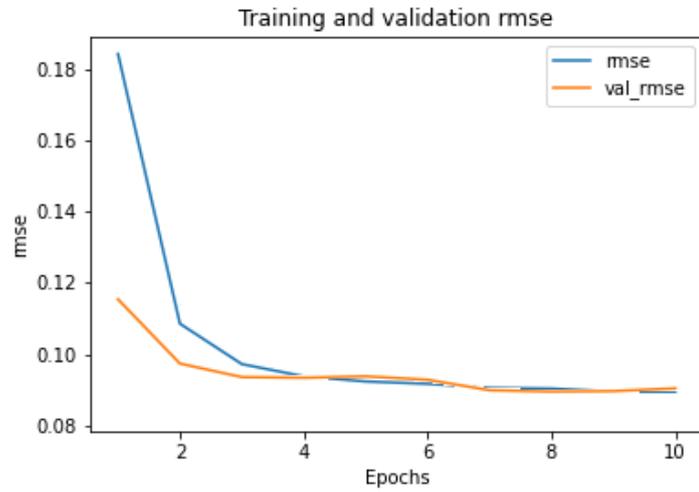


Figure 4.33. RMSE metric of the best ANN of the gearbox bearing model.

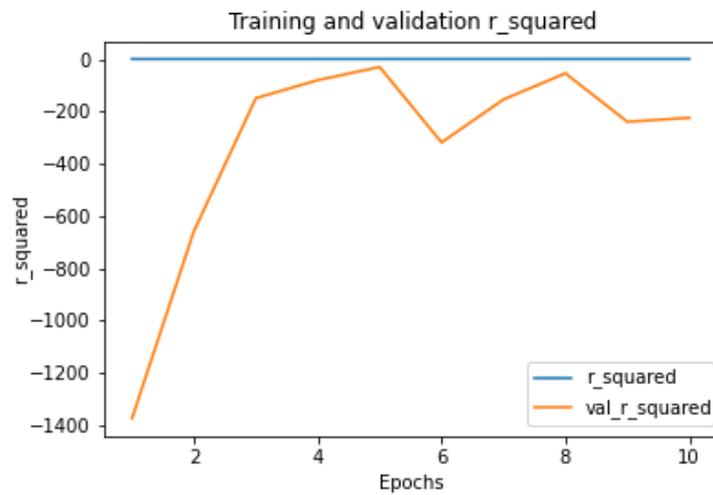


Figure 4.34.  $R^2$  metric of the best ANN of the gearbox bearing model.

Output: Gearbox Bearing Temperature							
WTG 6		WTG 7		WTG 2		Avg RMSE	Avg R squared
RMSE	R square	RMSE	R square	RMSE	R square		
0,408	0,989	0,6682	0,966	0,411	0,982	0,496	0,979

Table 4.11. Metrics evaluation on the 3 different test sets for the gearbox bearing model.

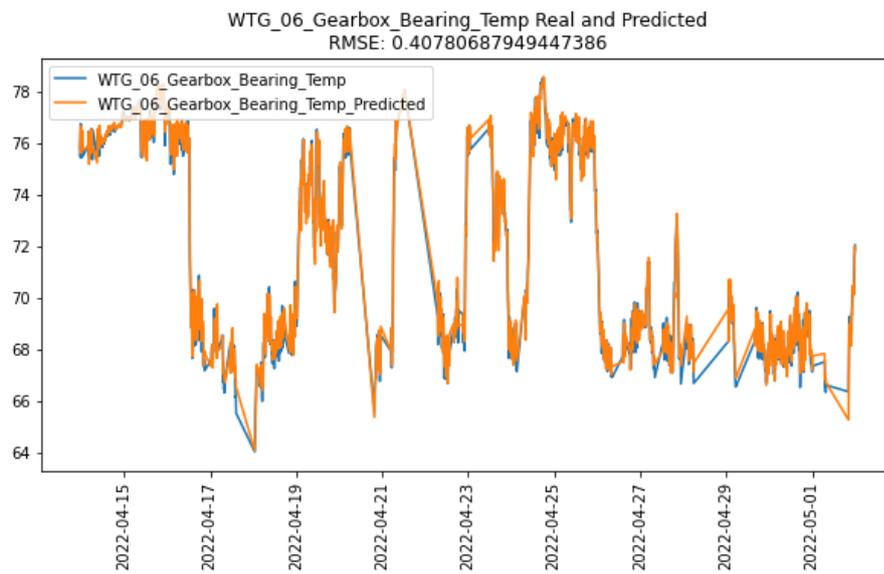


Figure 4.35. Real and predicted value of gearbox bearing temperature for WTG 6 test set.

## Generator Model

The training parameters used were the following:

- Type of network: Single LSTM.
- Numbers of hidden layers and node for each: 1(64).
- LSTM approach: L2.
- Inputs: Active Power, Ambient Temperature, Wind Speed, Rotor RPM, Generator RPM, Gearbox Oil Temperature, Gear Box Bearing Temperature.
- Scaling: Standard.
- Minimum validation loss: 0.00145 at epoch 9.
- Epochs of training: 10.
- Min validation RMSE: 0.03087.
- Max validation  $R^2$ : 0.99611
- Batch size: 64.
- Starting Learning rate: 0.0005.
- Learning Rate Reducer: Monitoring validation loss Min: 0.00050.

Figure 4.36 shows the graphs of loss functions MSE, instead Figures 4.33 and 4.38 show the metrics functions RMSE and  $R^2$ . Table 4.12 shows the results of the testing phase for the best model. The average RMSE is about 0.24 degrees Celsius. In Figure 4.39 the real value and the predicted value of the output are compared for the WTG 6 test set.

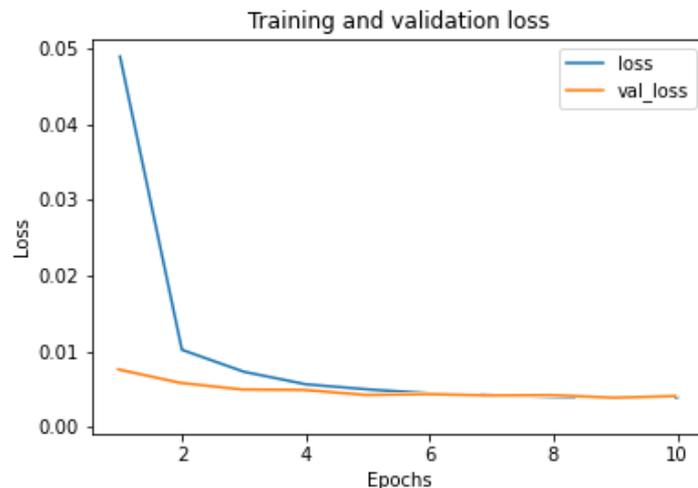


Figure 4.36. Loss of the best network for the generator model.

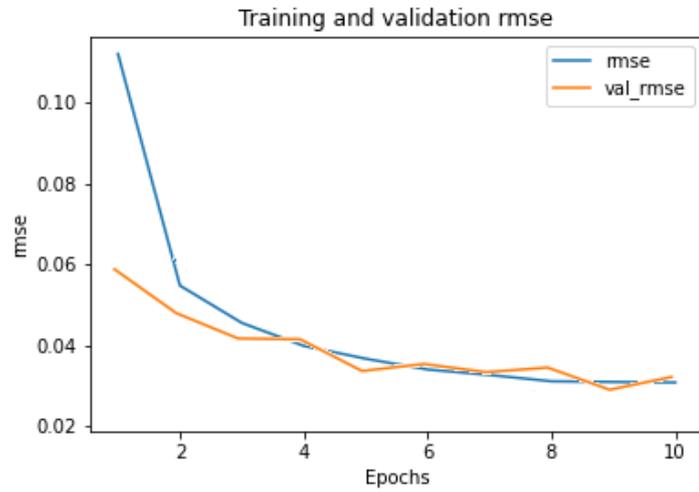


Figure 4.37. RMSE metric of the best ANN of the generator model.



Figure 4.38.  $R^2$  metric of the best ANN of the generator model.

Output: Generator Temperature							
WTG 6		WTG 7		WTG 2		Avg RMSE	Avg R squared
RMSE	R square	RMSE	R square	RMSE	R square		
0,195	0,997	0,29	0,996	0,227	0,997	0,237	0,997

Table 4.12. Metrics evaluation on the 3 different test sets for the generator model.

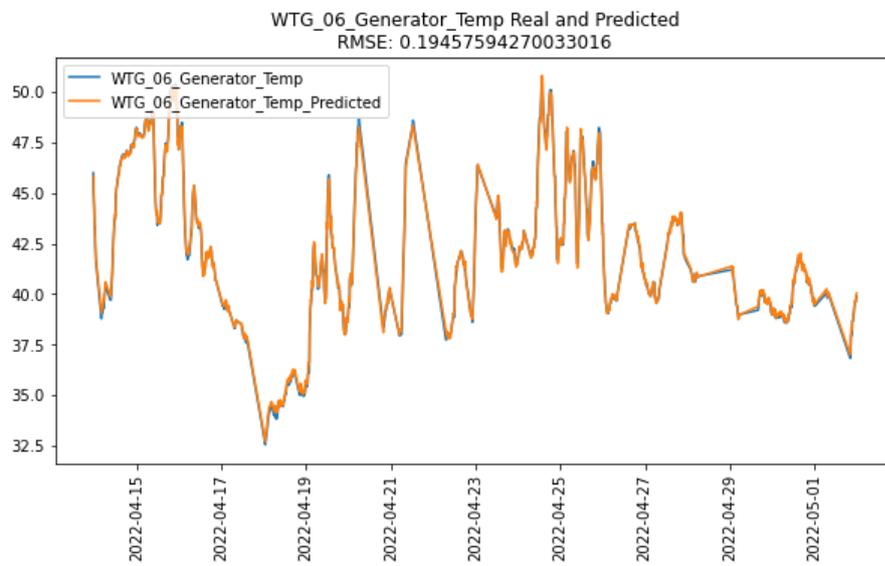


Figure 4.39. Real and predicted value of generator temperature for WTG 6 test set.

An important consideration is that the only model that does not rely on an LSTM approach is the one with the active power in the output, since this is an immediate function primarily of the wind characteristics at a specific instant, whereas for the other models the temperatures at a given instant depend on the history of temperature values.

## 4.6 Validation

After evaluating the ability of the various models in each category to predict the output variable correctly and selecting the best one, the error of the prediction for each sample is calculated, which is named **delta** or **deviation** and is denoted by the Greek letter  $\Delta$ :

$$\Delta = \text{actual value} - \text{estimated value}$$

At this stage it is important to choose the values of the UCL and LCL limits correctly, to implement a correct control chart, so the delta values of the test sets of the 3 turbines are merged and analyzed together.

Initially, an analysis of the distribution of delta values is performed to determine that the curve is similar to a random distribution and its mean value is around zero. This information is useful in understanding that the turbines are not affected by abnormal behavior. An attempt is made to choose limit values based on the mean and the standard deviation of delta values, trying to fit all delta values between the UCL and LCL. For the calculation of the limits, the same procedures used in creating the individual charts are followed, but the  $3\sigma$  rule is not adequate to include almost all values. Therefore the same formulas are used, adjusting the coefficient for each model. This coefficient is named  $Z$  and the formula for calculating the limits will be:

- UCL:  $\mu + Z\sigma$ .
- LCL:  $\mu - Z\sigma$ .

### 4.6.1 Delta/Output Chart

Furthermore, a delta analysis is performed based on the value of the output both to check if there are any outliers outside the limits and to verify that the deltas are equally distributed for all output data values with no particular anomalies dependent on this data.

### 4.6.2 Individual Chart

Then the delta values of each of the 3 test datasets are entered into the individual chart to perform a time-based evaluation and verify that there are no season anomalies. At this stage, the model can be defined as validated.

### 4.6.3 Results Validation

#### Wind Turbine Model

- $Z$  coefficient: 5.
- LCL: -238 W.
- UCL: 255 W.

Figure 4.40 shows the distribution of the delta values of all the turbines and the UCL and LCL limits. Figure 4.41 shows the delta values compared with the value of the active power, and also the UCL and LCL limits. It allows verifying that all the delta values are included between the 2 limits, so no value is found above UCL or LCL. Figure 4.42 shows the Shewhart charts and relative limits for the deviation data of the WTG 6.

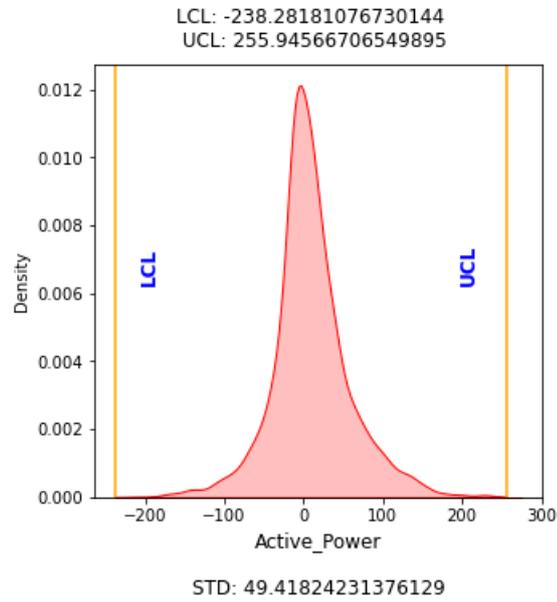


Figure 4.40. Wind turbine model - STD of the delta values of the 3 test set and limit LCL and UCL.

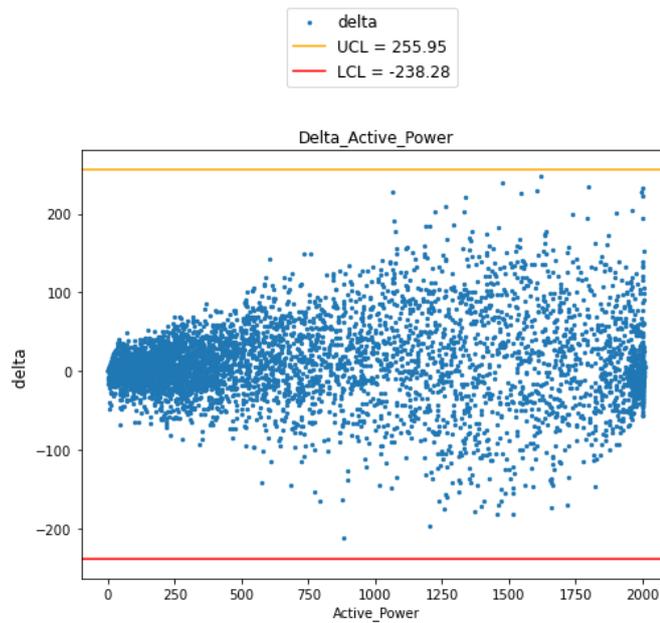


Figure 4.41. Wind turbine model - Delta values compared with output active power.

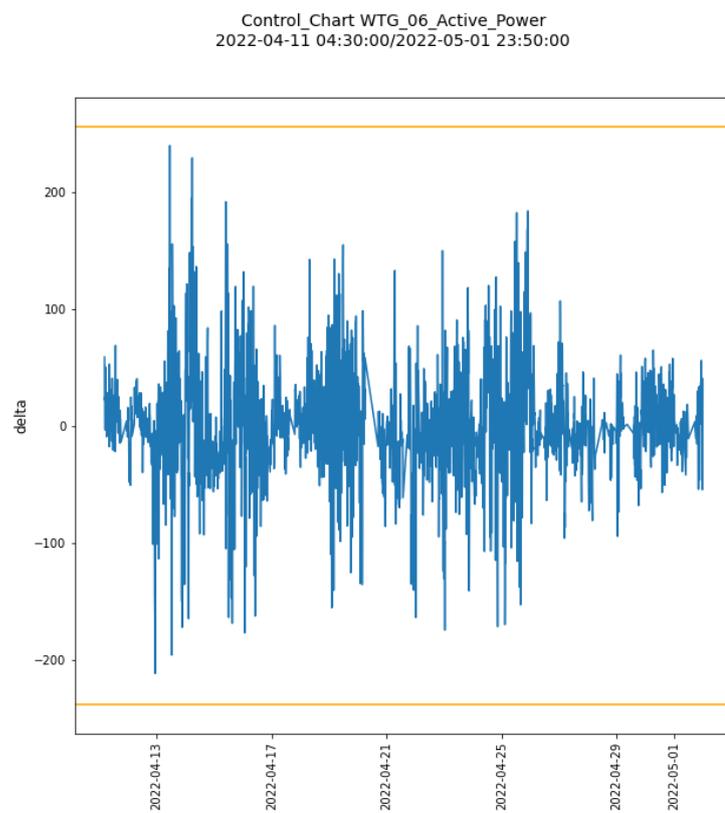


Figure 4.42. Wind turbine model - Shewhart control chart on the WTG 6 test set.

### Gearbox Oil Model

- Z coefficient: 4.
- LCL:  $-1.54\text{ }^{\circ}\text{C}$ .
- UCL:  $1.90\text{ }^{\circ}\text{C}$ .

Figure 4.43 shows the distribution of the delta values of all the turbines and the UCL and LCL limits. Figure 4.44 shows the delta values compared with the value of the temperature of the oil, and also the UCL and LCL limits. It allows verifying that all the delta values are included between the 2 limits, so no value is found above UCL or LCL. Figure 4.45 shows the Shewhart charts and relative limits for the deviation data of the WTG 6.

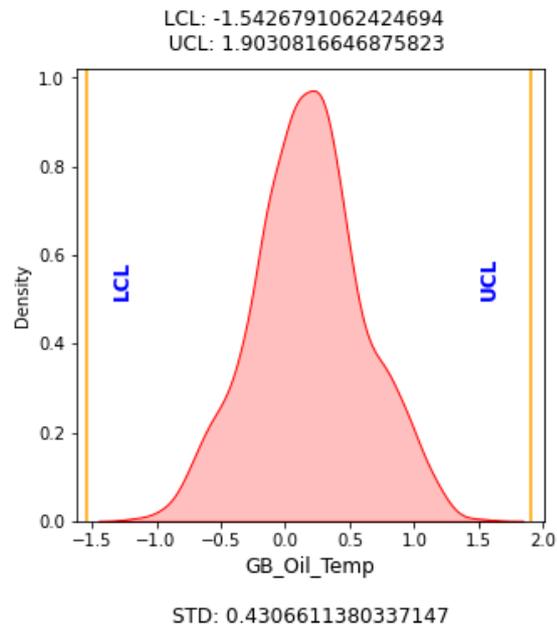


Figure 4.43. Gearbox oil model - STD of the delta values of the 3 test set and limit LCL and UCL.

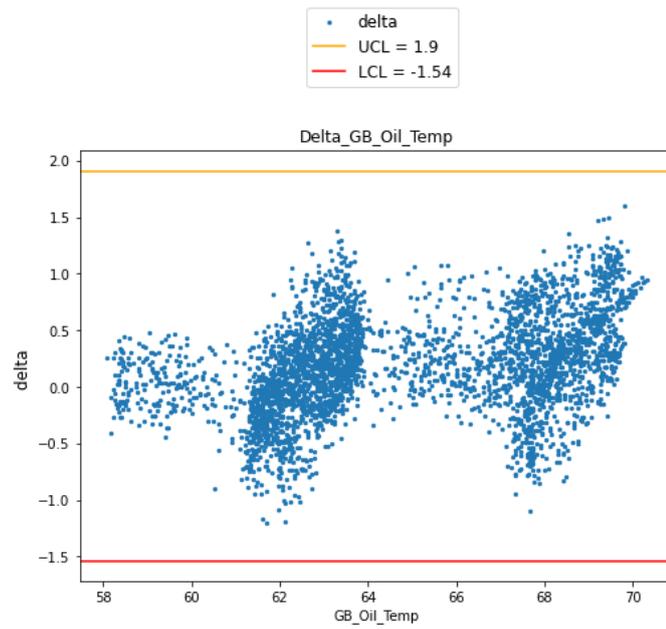


Figure 4.44. Gearbox oil model - Delta values compared with gearbox oil temperature.

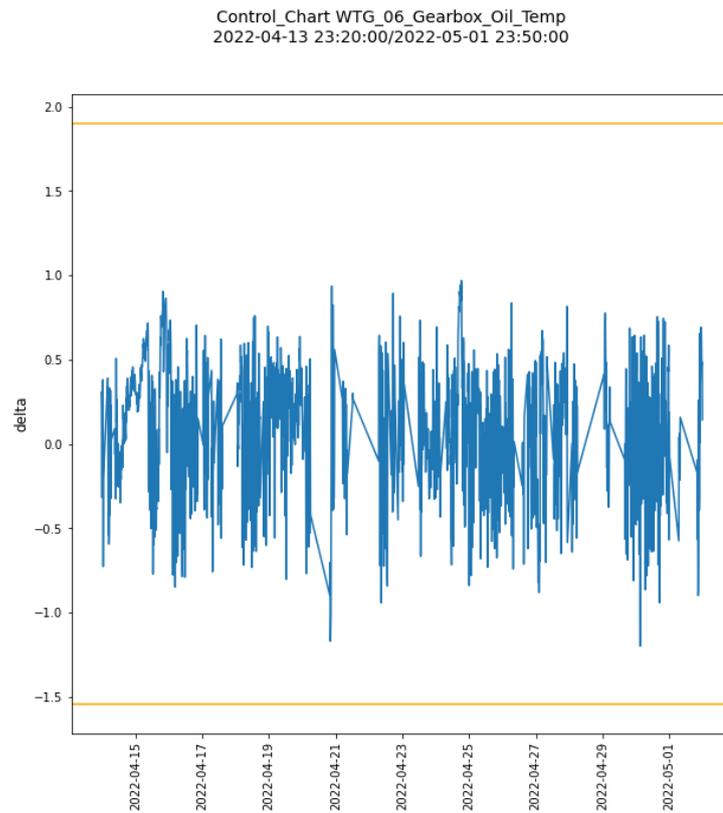


Figure 4.45. Gearbox oil model - Shewhart control chart on the WTG 6 test set.

### Gearbox Bearing Model

- Z coefficient: 5.
- LCL:  $-2.5\text{ }^{\circ}\text{C}$ .

- UCL: 2.36 °C.

Figure 4.46 shows the distribution of the delta values of all the turbines and the UCL and LCL limits. Figure 4.47 shows the delta values compared with the value of the temperature of the bearing, and also the UCL and LCL limits. It allows verifying that all the delta values are included between the 2 limits, so no value is found above UCL or LCL. Figure 4.48 shows the Shewhart charts and relative limits for the deviation data of the WTG 6.

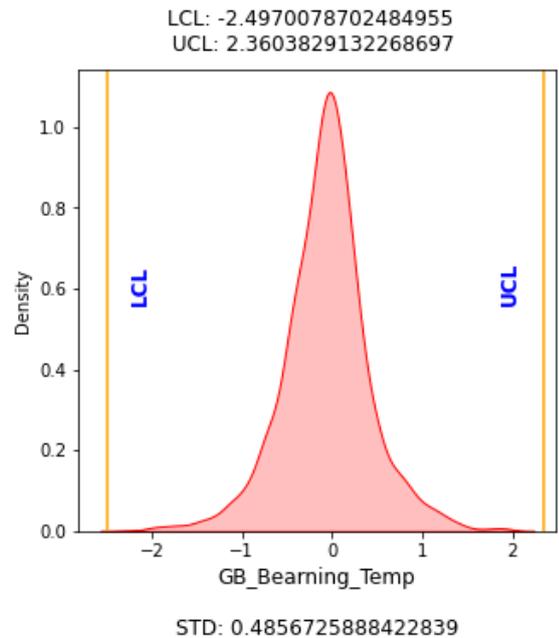


Figure 4.46. Gearbox bearing model - STD of the delta values of the 3 test set and limit LCL and UCL.

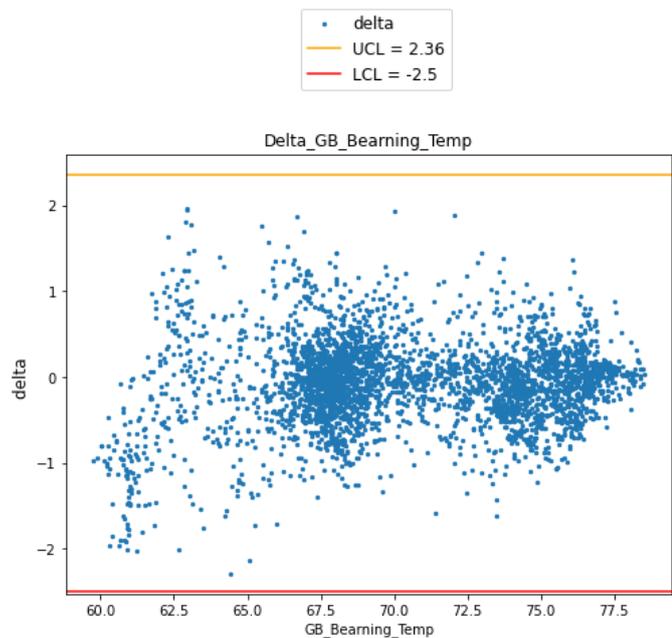


Figure 4.47. Gearbox bearing model - Delta values compared with gearbox temperature value.

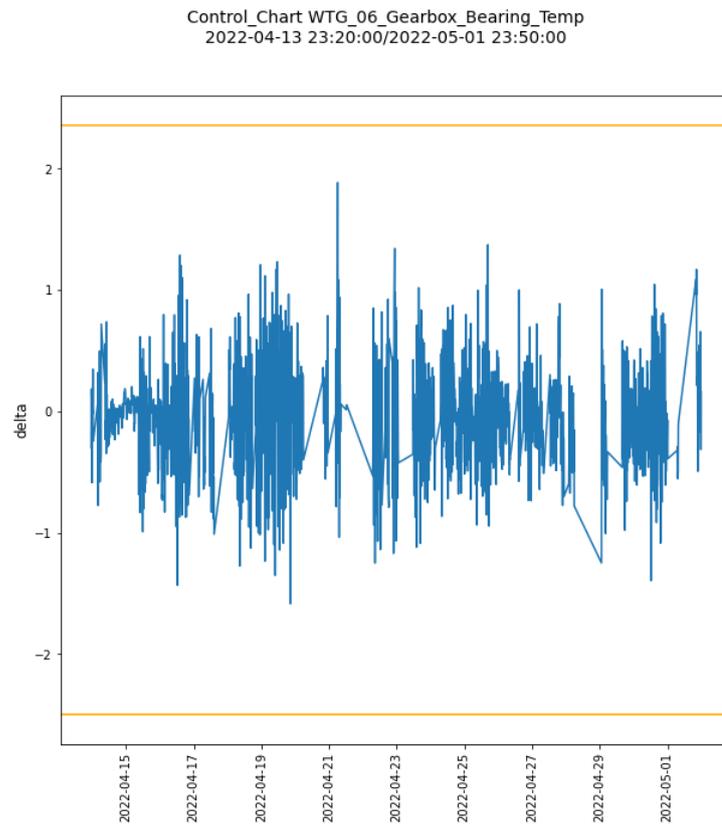


Figure 4.48. Gearbox bearing model - Shewhart control chart on the WTG 6 test set.

### Generator Model

- Z coefficient: 7.
- LCL: 1.56 °C.
- UCL: 1.64 °C.

Figure 4.49 shows the distribution of the delta values of all the turbines and the UCL and LCL limits. Figure 4.50 shows the delta values compared with the value of the temperature of the generator, and also the UCL and LCL limits. It allows verifying that all the delta values are included between the 2 limits, so no value is found above UCL or LCL. Figure 4.51 shows the Shewhart charts and relative limits for the deviation data of the WTG 6.

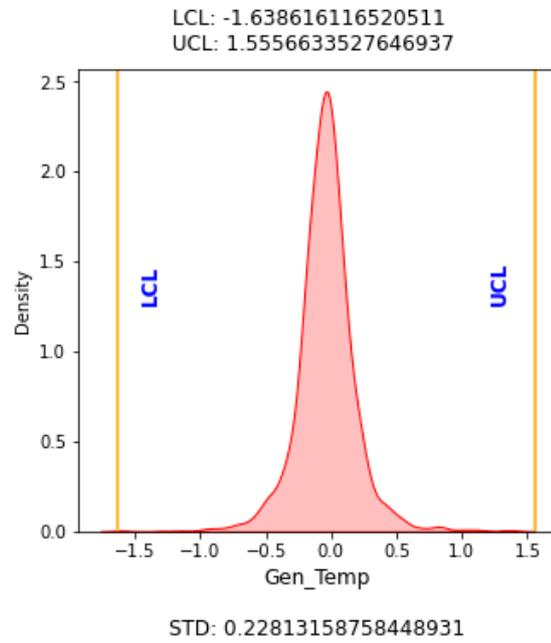


Figure 4.49. Generator model - STD of the delta values of the 3 test set and limit LCL and UCL.

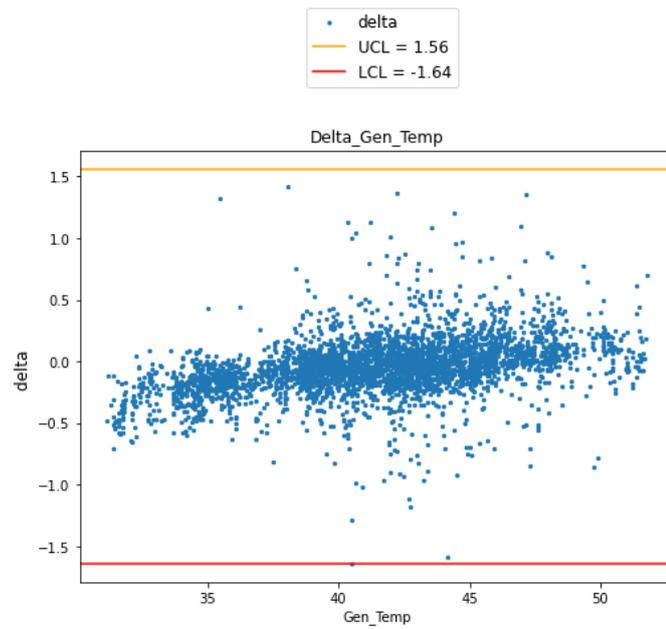


Figure 4.50. Generator model - Delta values compared with generator temperature value.

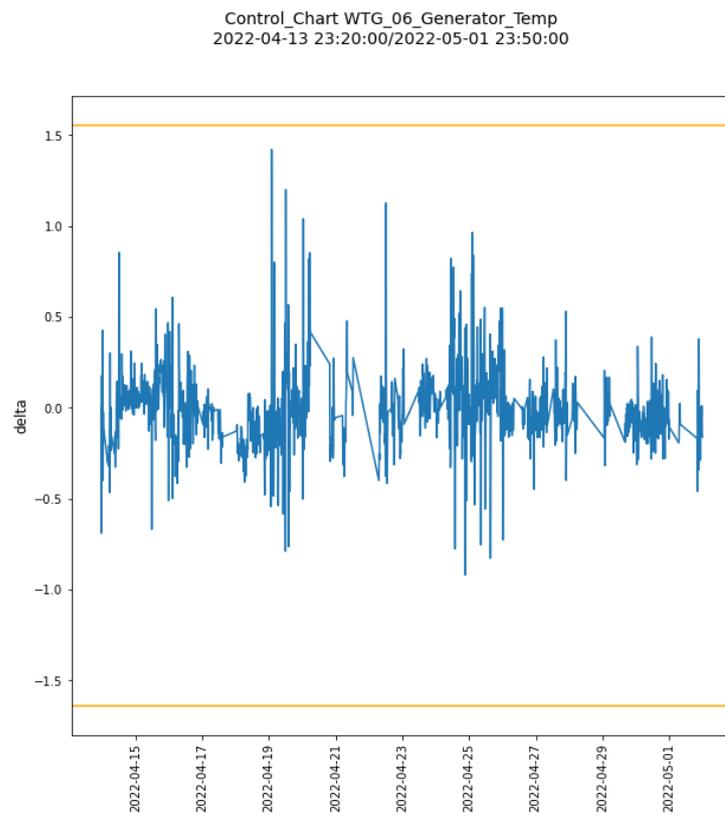


Figure 4.51. Generator model - Shewhart control chart on the WTG 6 test set.

# Chapter 5

## Solution - Control phase

This chapter will discuss the Control phase, showing the different solutions developed to address the problem of anomaly detection of a turbine.

### 5.1 Data Acquisition

Four turbines are used during the Control phase to test the operation of the monitoring process system. They were chosen after a careful examination of the event list of the Downtime Manager to highlight anomalous behavior before an alarm. For this purpose, the following turbines were chosen:

- WTG 1.
- WTG 3.
- WTG 4.
- WTG 5.

The same data acquisition and data cleaning operations were carried out on these turbines without substantial differences. The only notable note is that all possible samples were acquired without excluding any period. Table 5.1 shows the period of the data extracted from each turbine

WTG_01	29/08/2021
	17/04/2022
WTG_03	29/08/2021
	17/04/2022
WTG_04	29/08/2021
	17/04/2022
WTG_05	01/08/2021
	17/04/2022

Table 5.1. WTGs used in Control phase and relative data periods used.

## 5.2 Data Preprocessing

Unlike before, this preprocessing phase is not designed to eliminate outliers, which are instead critical to the purpose of the prediction. The models developed are designed to detect anomalies when the turbine is producing normally, so this operation is performed to exclude all periods of downtime. This procedure was carried out in several steps.

### 5.2.1 Data Cleaning

#### NULL Samples Elimination

In this step, all the samples with at least a NULL value in the data are eliminated.

### 5.2.2 Downtime Filtering

#### Filtering Periods Of Limitations

The same operation described in the model elaboration phase is performed, as the limitations affect the whole park, and the developed models are not designed for forecasting in these anomalous situations.

#### Status Filtering

Filtering the samples based on the T-Status value associated with them is critical for determining when the turbine is actually producing. Unlike before, samples before an alarm or communication error are not deleted.

#### Output Power Zero or Negative

Despite status filtering operation, there may still be some samples where the turbine is not producing, so all data with a power output value less than or equal to zero are removed.

## 5.3 Prediction of Variable Output

During the model elaboration phase, the best model for each category was selected to predict the chosen output variables. The new turbine data have to be submitted to the various models to make the predictions of the relative output variable. As a first step, it is critical to create the various test sets for each model by selecting the features that characterize it and deciding whether or not to proceed with creating the time windows if the model utilizes recurrent neural networks. All of the data available for each turbine is used to create test sets, which are then submitted to the various ANNs to perform forecast operations.

In the Control phase, no data merge or concatenation operation is performed but each turbine is treated individually. The RMSE value is calculated for each forecast. The value of RMSE calculated on the test set of the turbines used in the Control phase is higher than the value calculated on the predictions performed by the different models on the data of these turbines. This property is due to the presence of outliers. Figures 5.1, 5.3, 5.4, 5.5 show the output prediction for each model for WTG1 with the corresponding RMSE value. This value is compared with the average RMSE of the model calculated in the testing phase. Also, in Figure 5.2 the comparison between the real and predicted power curves of the same turbine is shown.

### 5.3.1 Wind Turbine Model

- RMSE: 67.508 W.
- Avg. RMSE: 50.683 W.

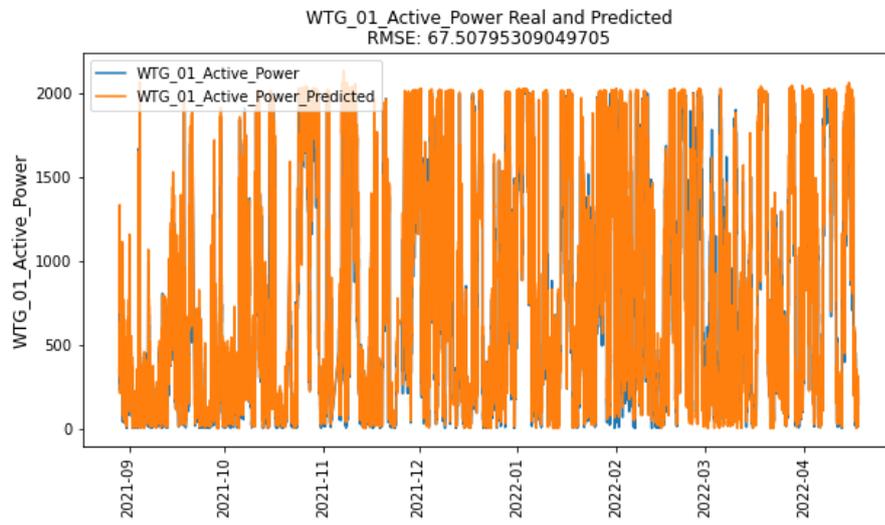


Figure 5.1. Real and predicted values of active Power of WTG 1.

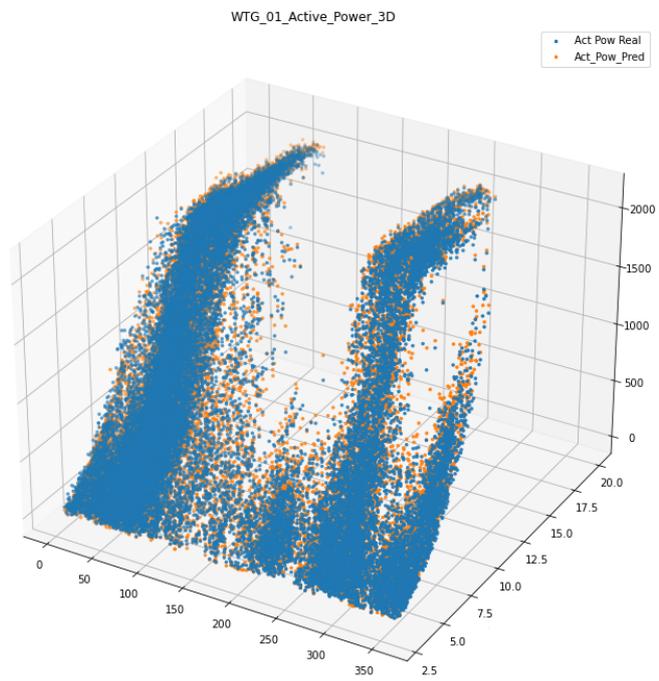


Figure 5.2. 3D power curves with the actual value and predicted value for WTG1.

### 5.3.2 Gearbox Oil Model

- RMSE: 0.519 °C.
- Avg. RMSE: 0.470 °C.

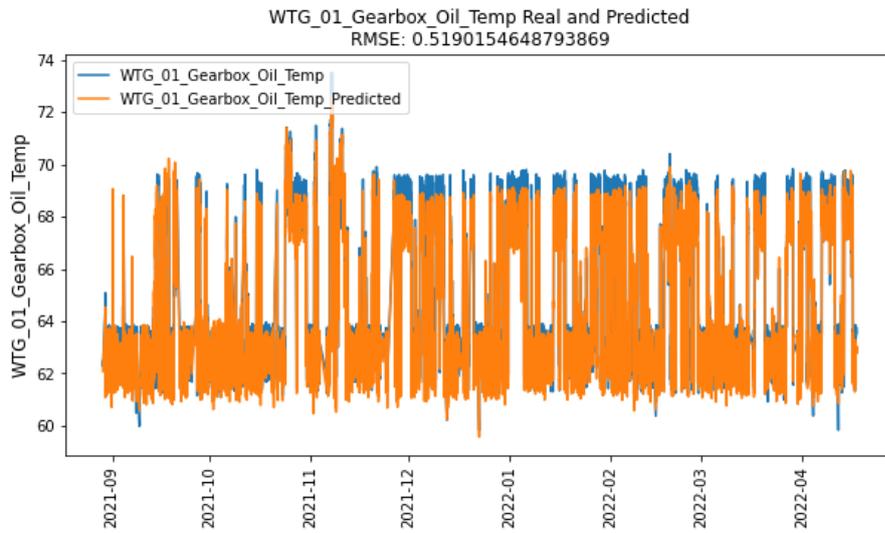


Figure 5.3. Real and predicted values of gearbox oil temperature of WTG 1.

### 5.3.3 Gearbox Bearing Model

- RMSE: 0.500 °C.
- Avg. RMSE: 0.496 °C.

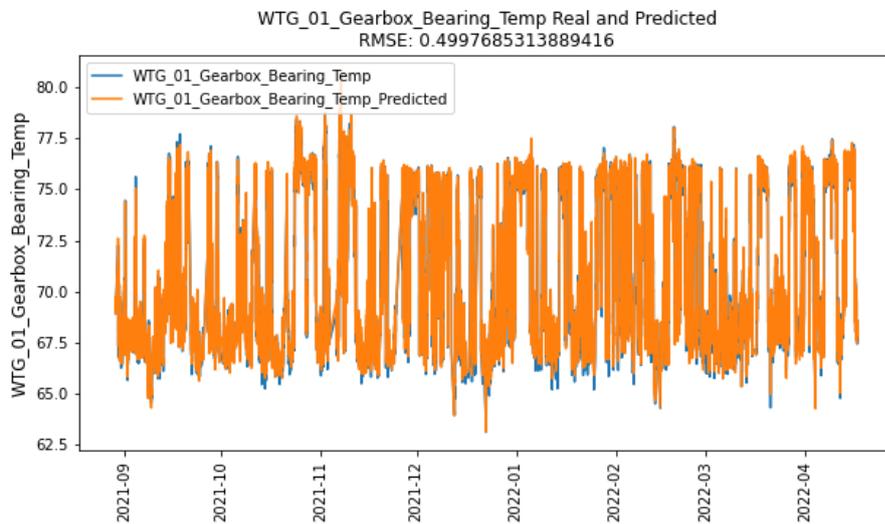


Figure 5.4. Real and predicted values of gearbox bearing temperature of WTG 1.

### 5.3.4 Generator Model

- RMSE: 0.288 °C.
- Avg. RMSE: 0.237 °C.

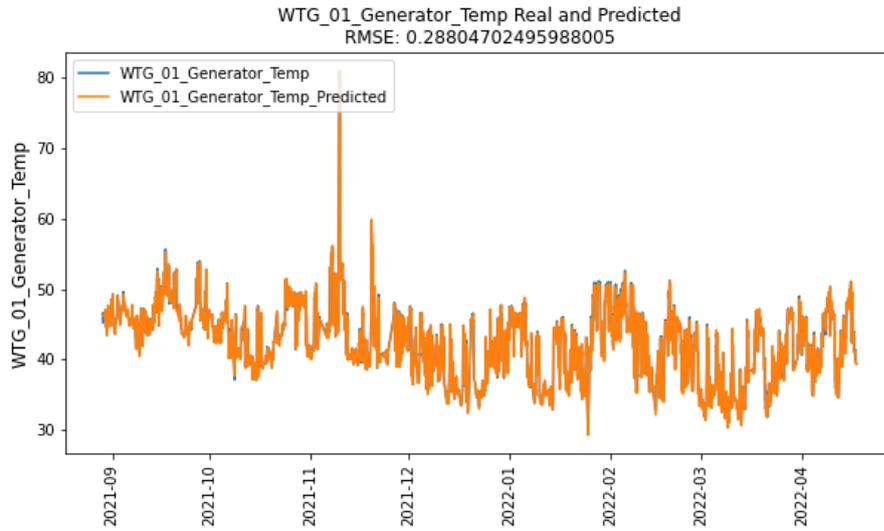


Figure 5.5. Real and predicted values of generator temperature of WTG 1.

## 5.4 Testing

After predicting the output variables for each model and turbine, the deviation values  $\Delta$  were calculated by subtracting the predicted value from the actual value. The deviation values are subsequently inserted into the control charts using LCL and UCL limits, both calculated in the Validation step of the Model Elaborating phase. Outliers (i.e., data greater than UCL or less than LCL) are analyzed to see if they might have any utility in preventing an alarm or representing abnormal turbine operation.

By analyzing the distribution of delta values, it was possible to verify that the standard deviation of the distribution was greater than in the data from the various test sets of the previous phase and, in addition, to get an idea of the number of outliers in each forecast.

Further analysis of the delta values based on the output variable, through a **Delta/Output chart**, allowed us to stable whether there was a relationship between the variables.

### 5.4.1 Power Curve Outliers

For the wind turbine model, the distribution of outliers as a function of wind speed was also studied through a particular representation of the power curve to discover a relationship between outliers and wind speed. In particular, the best predictions were divided by the outliers, and the whole was represented in a graph. Figure 5.6 shows an example for the WTG1; on the left side, there are the predicted and actual values, while on the right side there are only the actual and predicted values of the outliers.

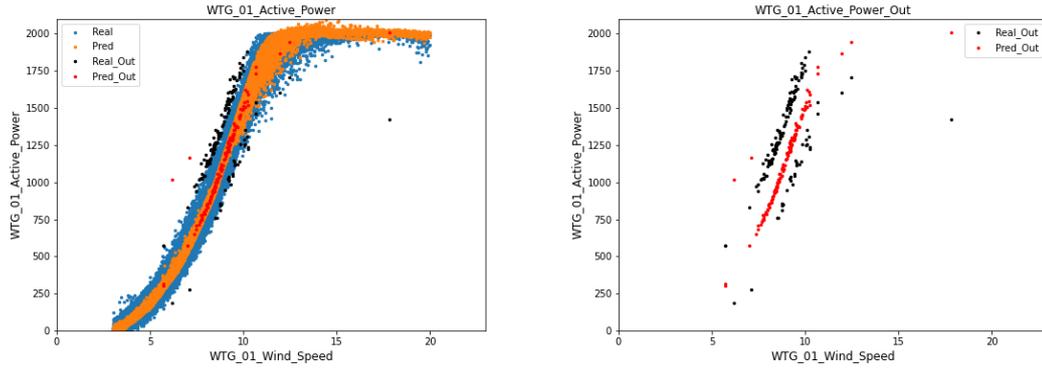


Figure 5.6. Power curves outliers for WTG1, with real (blue) and predicted (orange) values for good predictions, and real (black) and predicted (red) values for outliers.

### 5.4.2 Alternative Control Charts

A standard Shewhart control chart did not always prove to be a useful tool for isolating significant outliers and developing a system to generate alerts, so other solutions were developed. The main problem is that Shewhart control charts do not always allow the visualization of outlier patterns. This study was crucial in the development of alternative solutions. These solutions are designed to try to identify the continuous behavior of outliers over a long period, without focusing on the value of each outlier. A single high delta value could simply indicate an incorrect sensor measurement.

In the first solution, for each timestamp, the **delta value was averaged with the last 8 deltas** of the previous timestamp and this data have been inserted into the chart with the same UCL and LCL.

Next, the outliers above UCL were divided from the outliers below LCL, as outliers with positive or negative delta values were considered to be more or less significant for each model. For the model on the active power output, for example an outlier with a negative delta, indicates that the turbine is producing less than it should; on the other hand, for models based on the temperature of a component the problems are surely found when the temperature is higher than normal and not when the component is colder. Two solutions were developed based on the concept of grouping outliers by period:

- **Sum of  $\Delta_{th}$** : For each outlier, the amount of delta that exceeds the threshold is calculated.

$$\Delta_{th} = \Delta - UCL \text{ (for } \Delta > UCL)$$

$$\Delta_{th} = \Delta - LCL \text{ (for } \Delta < LCL)$$

This value called  $\Delta_{th}$  is summed with the values of  $\Delta_{th}$  of all previous timestamps if they also represent outliers, so if  $\Delta_{th}$  value is different than zero. These values were subjected to a control chart by multiplying the UCL and LCL for a value N. The value N is set to **1.5**. The sum is represented in the graph with the color purple while each outlier with the color black. The latter information represents only the presence of an outlier and is not to be read according to the scale. Figure 5.7 illustrates the process of creation for a "Sum of  $\Delta_{th}$ " chart with only positive outliers.

- **Sum of events**: This solution treats the presence of an outlier as an event and counts consecutive positive (or negative) outliers, signaling an alarm when more than 720 consecutive outliers are found. The sum is represented in the graph with the color red while each outlier is represented with the color blue. Again, the information represents only the presence of an outlier and is not to be read according to the scale. The value 720 corresponds to 10 consecutive days (since in each hour there are 6 samples (6 timestamps x 24 hours x 10 days

= 720 samples). Figure 5.8 illustrates the process of creation for a "Sum of events" chart with only positive outliers and with a threshold of 5 consecutive outliers (instead of 720).

The next chapter discusses the results for each model and each turbine and the various solutions used.

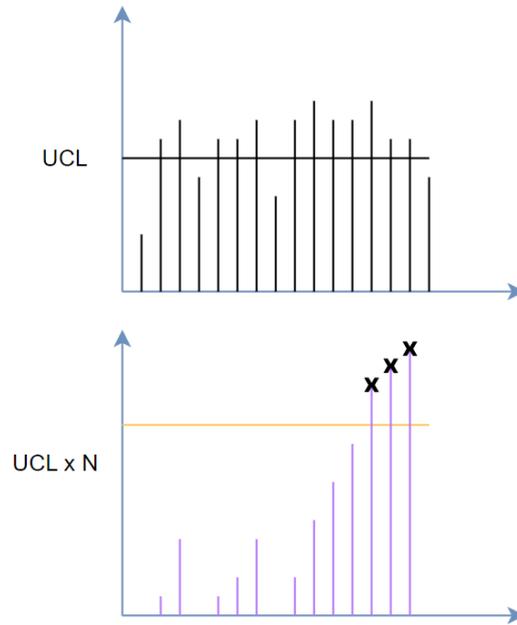


Figure 5.7. Principle of creating the control chart "Sum of  $\Delta_{th}$ " for positive outliers: top figure shows the  $\Delta$  values, while the bottom one shows the possess of sum of  $\Delta_{th}$  values. In the last three samples, the value of  $1.5 \times UCL$  is exceeded and alarms are generated.

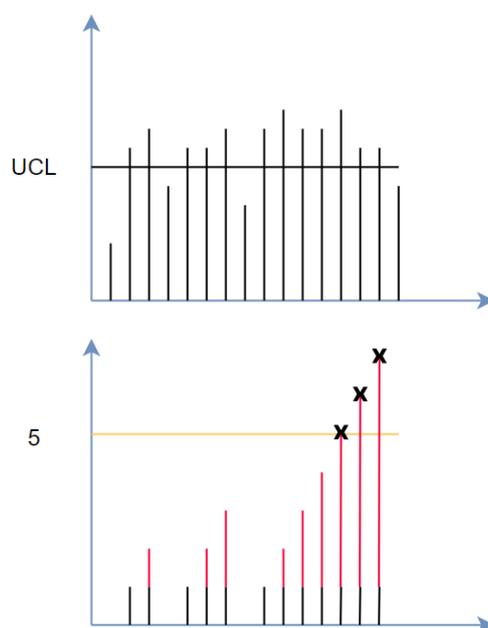


Figure 5.8. Principle of creating the control chart "Sum of events" for positive outliers: top figure shows the  $\Delta$  values, while the bottom one shows the possess of the sum of numbers of outliers. In the last three samples, the value of 5 consecutive outliers is exceeded and alarms are generated.

## 5.5 Results

Results obtained for each model in predicting alarms will be discussed below. In addition, an analysis of anomalous behaviors is also provided.

### 5.5.1 Wind Turbine Model

Analyses of outliers through the standard control chart or the averaging chart were not found to be useful in identifying the known alarms of the various turbines. As specified earlier, the main problems occur when the power produced does not meet the value the ideal value, so the various outliers were divided into positive and negative and treated separately with a Sum of  $\Delta_{th}$  chart. Since no alarms were found for WTG3 and WTG4, only the analysis of the results for WTG1 and WTG5 will be discussed.

#### WTG 1

Downtime Manager shows the following alarms:

- Start: 2022/03/11 15:27:05.
- End: 2022/03/11 17:15:20.
- Duration: 01:48:15.
- Description: Wind turbine in emergency.

Figure 5.9 shows a comparison of delta values with a standard control chart, while Figure 5.10 shows an analysis with an averaging method. In this particular case, it is possible to appreciate how the two graphs are poorly adapted for the purpose, although irregularities can be visualized by analyzing the biggest negative outliers in the standard chart.

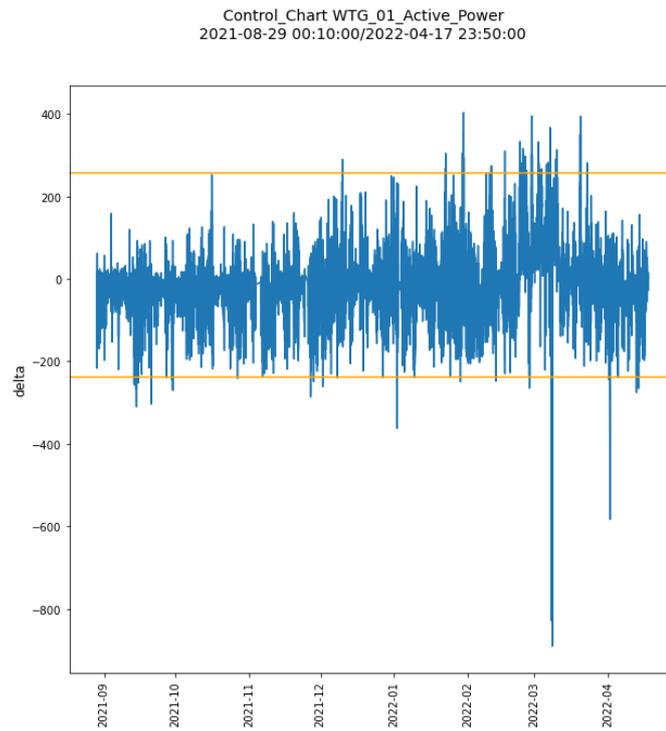


Figure 5.9. Control chart of  $\Delta$  active power for WTG 1.

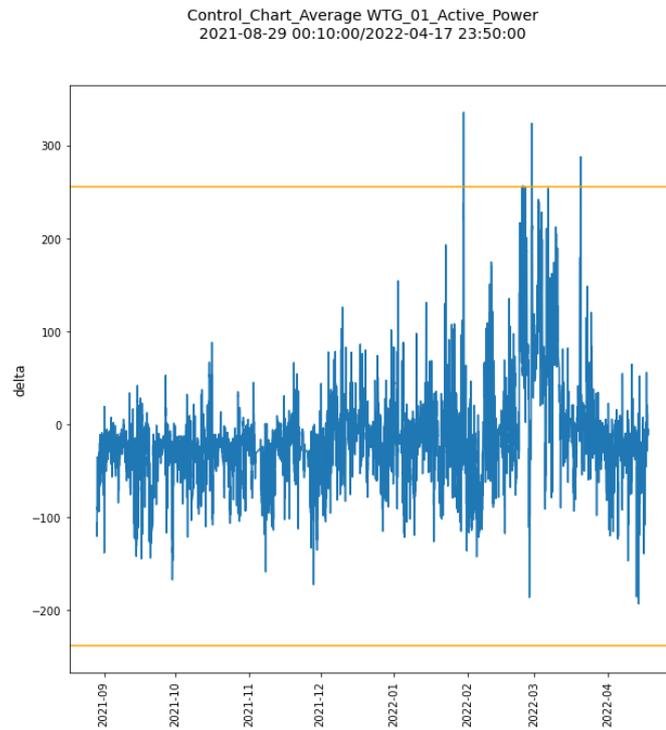


Figure 5.10. Control chart average of  $\Delta$  active power for WTG 1.

Finally, in Figure 5.11 the control graph is illustrated with the method Sum of  $\Delta_{th}$  that gives more precise information. A series of consecutive alarms inform of the abnormal behavior of the turbine. The first of the alarms are generated about 3 days before the problem is detected by

the Downtime Manager and is dated **2022-03-08 04:40:00**. Although above the limit, another significant decrease in power occurred on 01/04/2022, it is due to a restrictive regime the turbine is subjected to, also reported by the Downtime Manager.

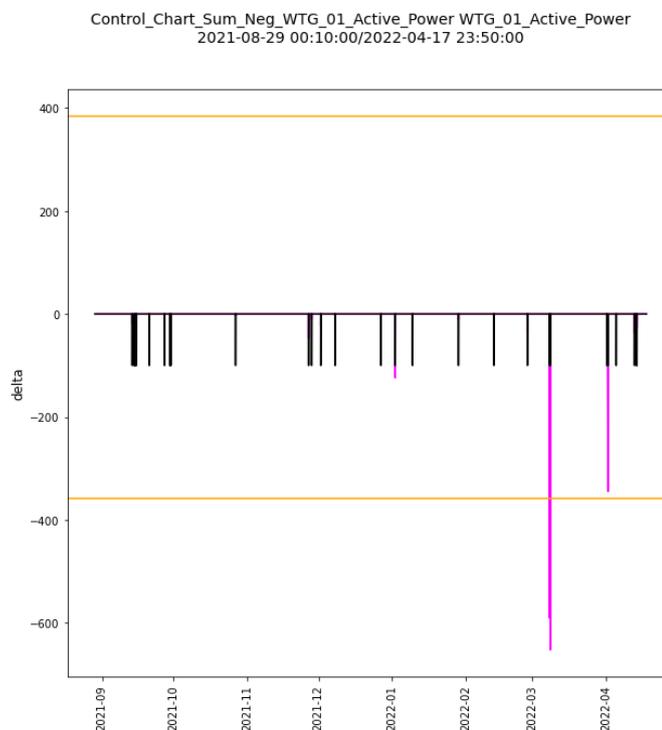


Figure 5.11. Sum of  $\Delta_{th}$  control chart for negative  $\Delta$  values calculated for active power of WTG 1.

## WTG 5

About WTG5, the Downtime Manager shows the following alarm:

- Alarm 1:
  - Start: 2021/12/07 02:26:06.
  - End: 2021/12/07 02:58:24.
  - Duration: 00:32:18.
  - Description: Wind turbine in emergency.
- Alarm 2:
  - Start: 2022/01/27 11:29:35.
  - End: 2022/01/27 12:34:05.
  - Duration: 01:04:30.
  - Description: Wind turbine in emergency.
- Alarm 3:
  - Start: 2022/01/27 14:35:20.
  - End: 2022/01/27 16:26:35.
  - Duration: 01:51:15.
  - Description: Wind turbine in emergency.

Figure 5.12 shows a comparison of delta values with a standard control chart, while Figure 5.13 shows an analysis with an averaging method. From the analysis of the charts it can be seen that the negative delta values are much larger than WTG1, and from the analysis of the average chart a particular pattern can be highlighted, but again both charts do not help in predicting alarms of Downtime Manager and better results can be obtained with the other two techniques.

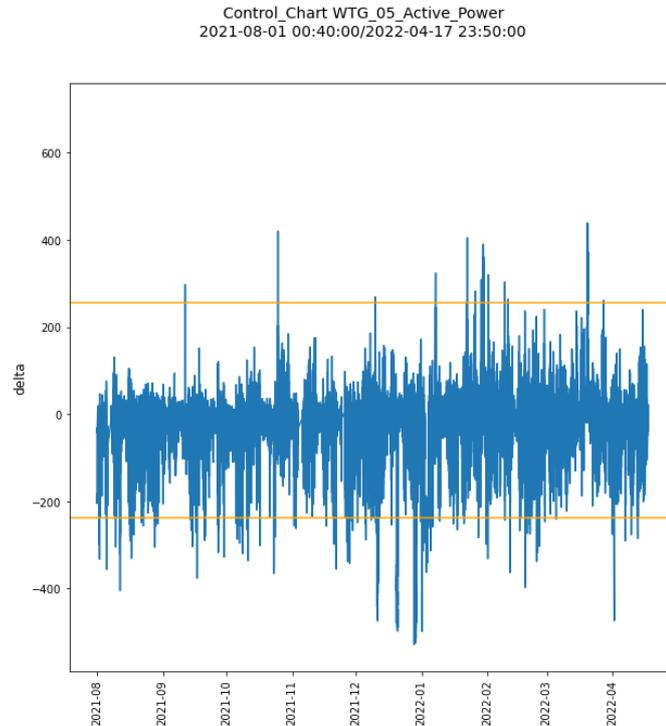


Figure 5.12. Control chart of  $\Delta$  active power for WTG 5.

Figure 5.14 show the Sum of  $\Delta_{th}$  control chart, which provides better results in terms of forecasting. But unfortunately, the alarms are generated after the Downtime Manager reports. A set of alert start to be generated the **2021-12-11 00:20:00**, 4 days after Alarm 1 and a second set the **2022-02-03 17:10:00**, one week after Alarm 2 and 3. In December, for this turbine, there is a huge loss of production of power, probably a symptom of a fault. The analysis through the "Power Curve Outliers" graph (Figure 5.15) has shown that for that period the turbine appears to be depowered, producing up to 500 W less, when it should have been operating at maximum power. Analysis through the delta/output graph (Figure 5.16) confirms that the largest errors occur for output powers between 1.5 MW and 2MW

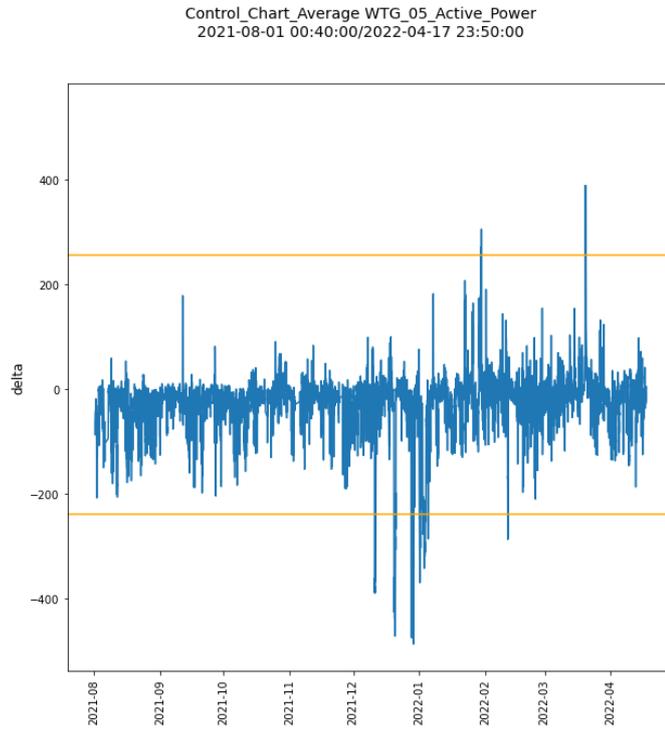


Figure 5.13. Control chart average of  $\Delta$  active power for WTG 5.

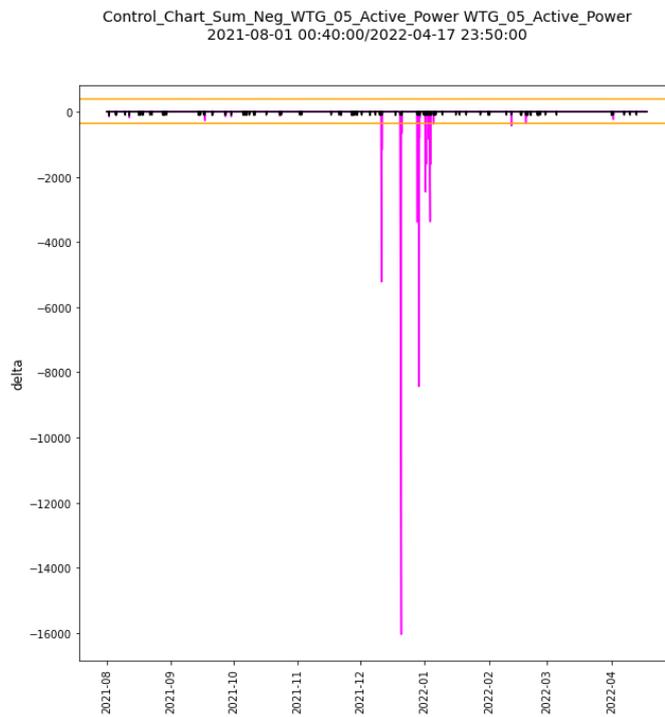


Figure 5.14. Power curves outliers Sum of  $\Delta_{th}$  control chart for negative  $\Delta$  values calculated for active power of WTG 5.

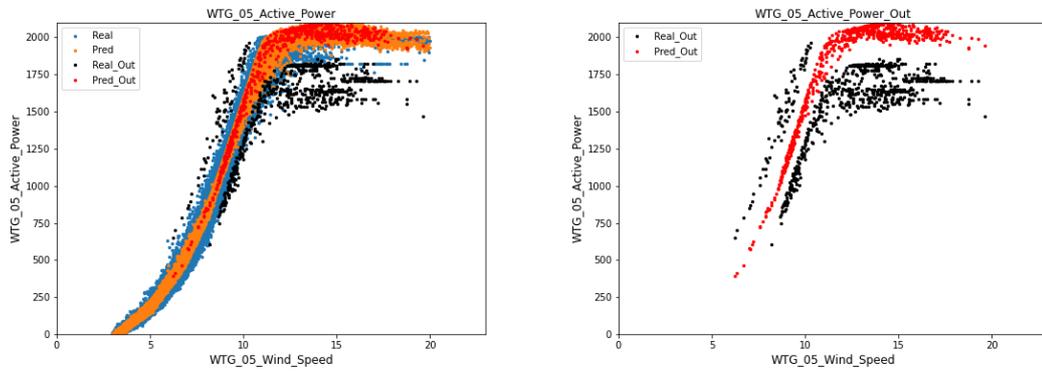


Figure 5.15. Power curves outliers for WTG5, with real (blue) and predicted (orange) values for good predictions, and real (black) and predicted (red) values for outliers.

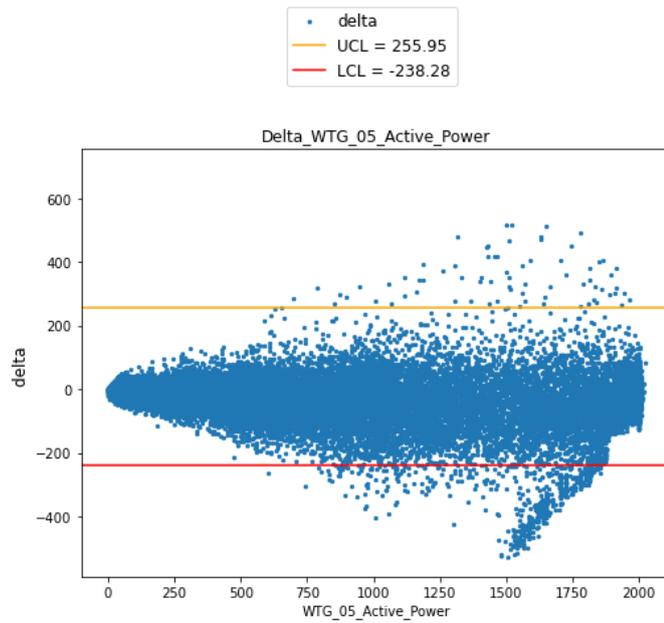


Figure 5.16. Delta/Output chart of  $\Delta$  active power for each output active power of WTG 5.

## 5.5.2 Gearbox Oil Model

This model was designed to try to anticipate high-temperature alarms of the oil in the component. The Downtime Manager analysis showed only one alarm in WTG 5. With the system developed in this thesis, it was possible to identify it after dividing the outliers into positive and negative. Both Sum of  $\Delta_{th}$  and Sum of events charts provided good results but the second one proved to be more efficient. For the same turbine, other abnormal oil temperature values were detected for fairly prolonged periods, and these were not associated with Downtime Manager alarms. For the other turbines, analysis of all charts showed no abnormalities and thus the system did not generate alarms.

### WTG 5

Downtime Manager shows the following alarm:

- Start: 2021/08/01 03:17:07.
- End: 2021/08/01 10:36:15.
- Duration: 07:19:08.
- Description: Gearbox Fan Thermal Relays triggered and High Gearbox Oil Temperature (Oil temperature exceeded 80 degrees for 1 minute and triggering of fan thermal fuse).

Figure 5.17 shows a comparison of the delta values with a standard control chart and did not prove useful in isolating the alarm, but it showed abnormal behavior, particularly markedly elevated oil temperature in the last two weeks of December. Both charts, Sum of  $\Delta_{th}$ , shown in Figure 5.18 and Sum of events, shown in Figure 5.19, proved to detect a problem in the first days of August, but both only few days late. This is probably due to the absence of information on data before August 1; that date corresponds to the same day the problem occurred. However, of the two methods, the latter proved more efficient because it was able to anticipate the reporting of the known problem as well as the abnormal oil temperature in the December period.

The first of the alarms is generated on **2021-08-05 23:50:00** while the last alert is on 2021-08-06 03:50:00 .The problem is detected by the Downtime Manager and is dated **2022-03-08 04:40:00**. For what concerns the abnormal period, the alarms start on **2021-12-10 00:50:00** and end on **2022-01-04 10:10:00**. Analysis with the delta/output graph (Figure 5.20) allows verifying that most of the outliers are positive and relate to higher oil temperatures, so the oil reaches too high temperatures.

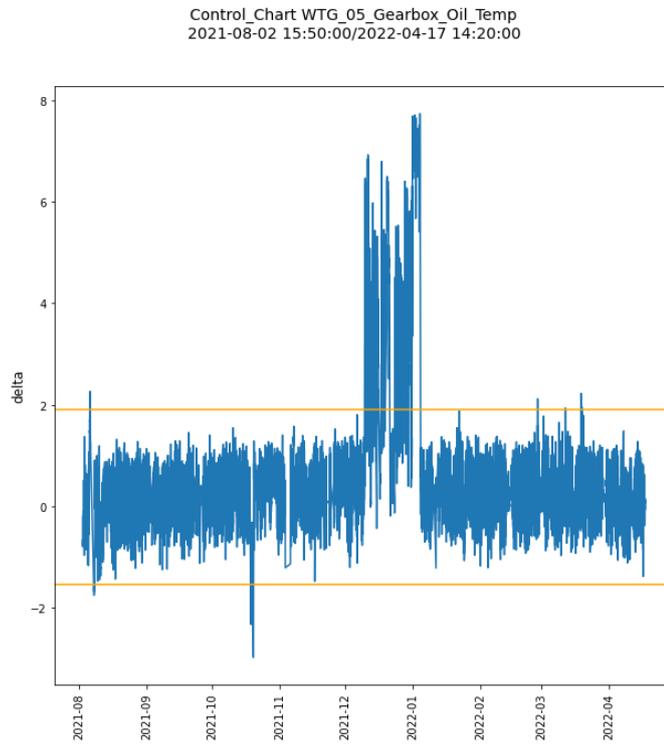


Figure 5.17. Control chart of  $\Delta$  gearbox oil temperature for WTG 5.

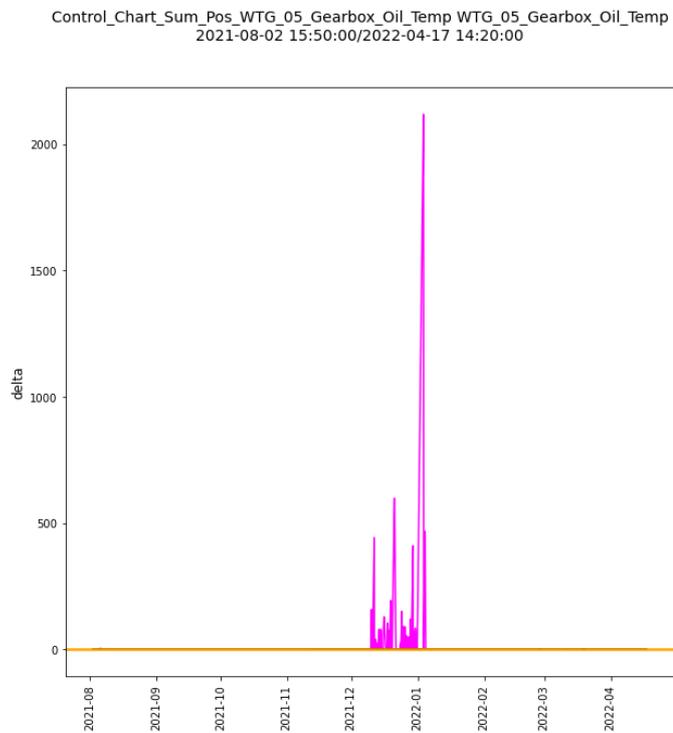


Figure 5.18. Sum of  $\Delta_{th}$  control chart for positive  $\Delta$  values calculated for gearbox oil temperature of WTG 5.

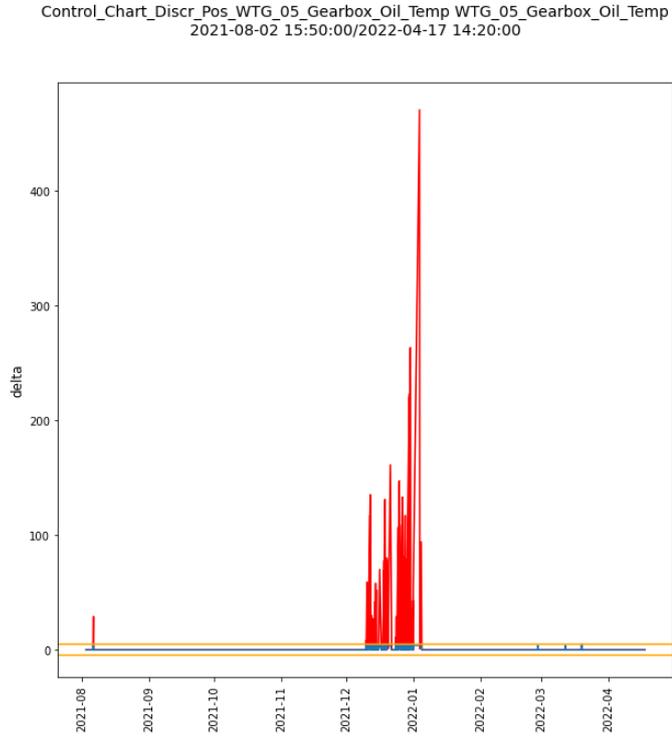


Figure 5.19. Sum of events control chart for positive  $\Delta$  values calculated for gearbox oil temperature of WTG 5.

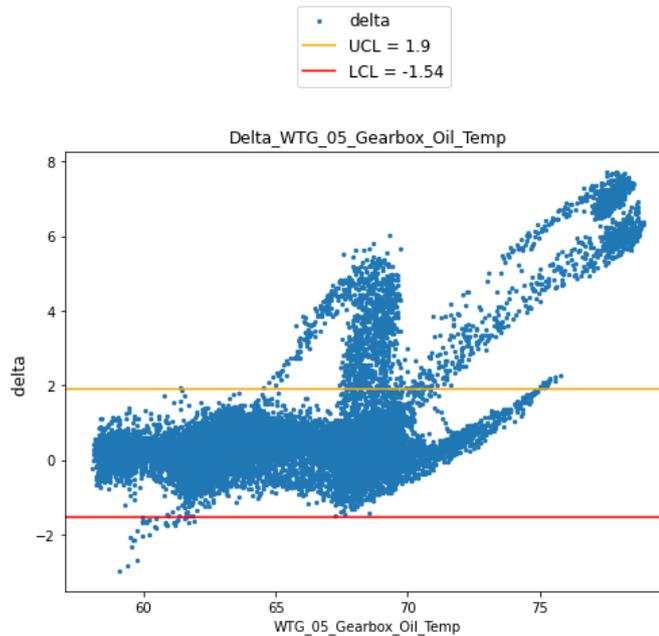


Figure 5.20. Delta/Output chart of  $\Delta$  gearbox oil temperature for each output gearbox oil temperature of WTG 5.

### 5.5.3 Gearbox Bearing Model

This model was designed to try to anticipate high-temperature alarms of the bearing of the gearbox. The Downtime Manager analysis showed a series of alarms in WTG 4, with a duration of a few minutes. With the system developed in this thesis was possible to identify the alarms after dividing the outliers into positive and negative. Both Sum of  $\Delta_{th}$  and Sum of events charts provided good results but the first one proved to be more efficient. For the same turbine, other abnormal gearbox temperature values were detected for fairly prolonged periods, and these were not associated with Downtime Manager alarms. For the other turbines, analysis of all charts showed no abnormalities and thus the system did not generate alarms.

#### WTG 4

Downtime Manager shows the following alarms:

- First alarm: 2021/09/17 00:39:32.
- Last alarm: 2021/10/15 18:48:45.
- Description: High gearbox bearing temperature.

Figure 5.21 shows a comparison of delta values with a standard control chart and did not prove useful in isolating the alarm, but it showed an abnormal behavior in particular a markedly high bearing temperature in September/October and December/January. Both charts, Sum of  $\Delta_{th}$ , shown in Figure 5.22 and Sum of events, shown in Figure 5.23, proved to detect a problem in September and October by going ahead of the Downtime Manager alarms, but also evident abnormal behavior in the period between mid-December and early February. However, of the two methods, the first one proved to be more efficient because it was able to anticipate the reporting of the known problem, instead the second performed well with the abnormal gearbox temperature in the December period.

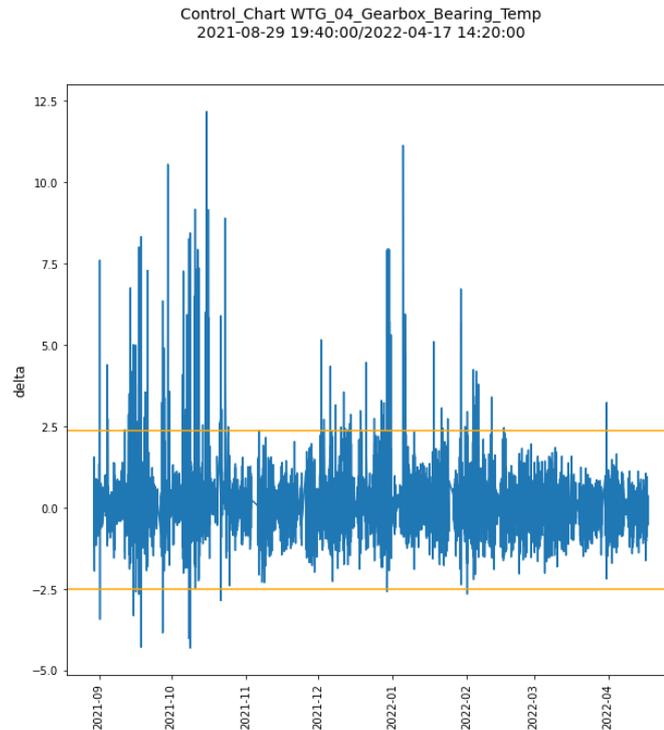


Figure 5.21. Control chart of  $\Delta$  gearbox bearing temperature for WTG 4.

Control\_Chart\_Sum\_Pos\_WTG\_04\_Gearbox\_Bearing\_Temp WTG\_04\_Gearbox\_Bearing\_Temp  
2021-08-29 19:40:00/2022-04-17 14:20:00

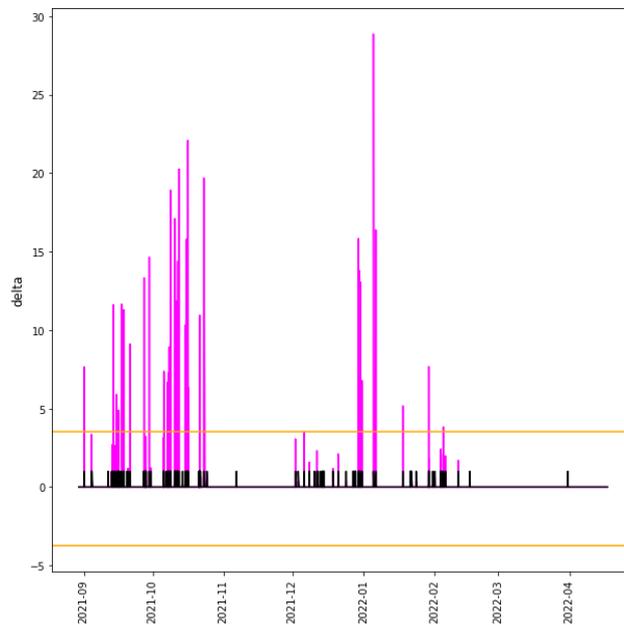


Figure 5.22. Sum of  $\Delta_{th}$  control chart for positive  $\Delta$  values calculated for gearbox bearing temperature for WTG 4.

Control\_Chart\_Discr\_Pos\_WTG\_04\_Gearbox\_Bearing\_Temp WTG\_04\_Gearbox\_Bearing\_Temp  
2021-08-29 19:40:00/2022-04-17 14:20:00

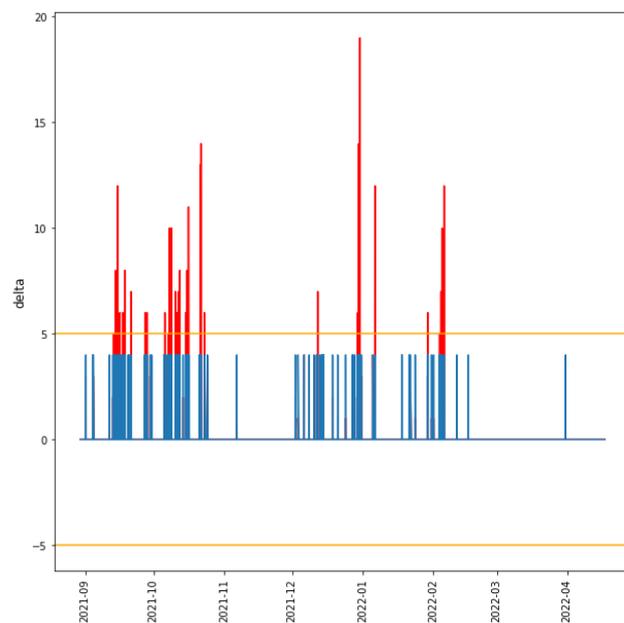


Figure 5.23. Sum of events control chart for positive  $\Delta$  values calculated for gearbox bearing temperature of WTG 4.

The first alarm was generated on **2021-09-01 06:30:00**, and the last alarm was generated on **2021-10-23 09:00:00**. The problem was detected by the Downtime Manager and is dated **2021/09/17 00:39:32** In terms of the anomaly period, the alarms start on **2021-12-12 07:10:00** and end on **2022-02-05 20:40:00**. The analysis of the delta/output graph (Figure 5.24) allows

verifying that most of the outliers are positive and relate to higher bearing temperatures, so this component reaches too high temperatures.

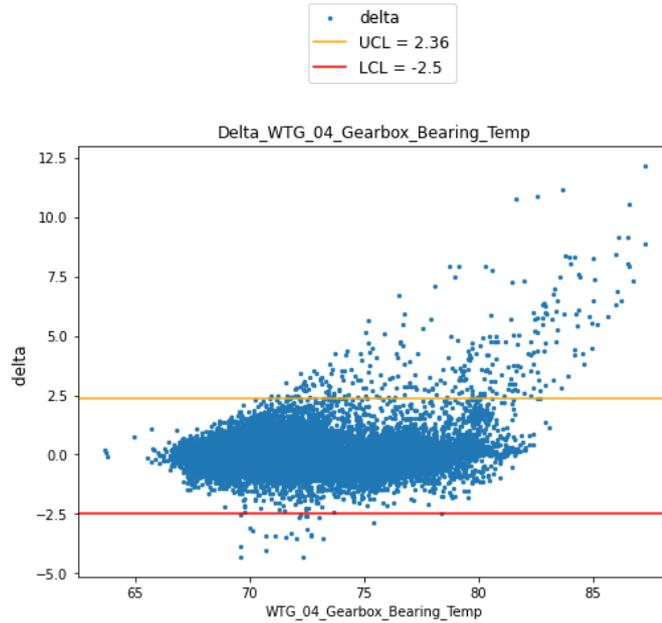


Figure 5.24. Delta/Output chart of  $\Delta$  gearbox bearing temperature for each output gearbox bearing temperature of WTG 4.

### 5.5.4 Generator Model

After the Downtime Manager analysis, as regards the generator, no related alarms were found. However, the system developed in this thesis showed significant results in detecting abnormal behavior for WTG3 and WTG5.

#### WTG 3

Analysis of the standard control chart immediately highlights anomalous patterns. Generator temperature delta values fluctuated strongly both positively and negatively during the periods of late November/early December, mid-January, and early April. Significant alarms could be generated through the two Sum of  $\Delta_{th}$  charts on both positive and negative outliers. Figure 5.25 shows a comparison of delta values with a standard control chart, instead, Figures 5.26 and 5.27 show the generation of alarms through Sum of  $\Delta_{th}$  chart for both positive and negative outliers. Alarms were generated during the following periods:

- Start: **2021-11-24 00:00:00** End: **2021-12-08 07:10:00**.
- Start: **2022-01-07 09:50:00** End: **2022-01-18 05:10:00**.
- Start: **2022-04-03 19:40:00** End: **2022-04-04 17:30:00**.

Delta/output graph analysis shows no significant results.

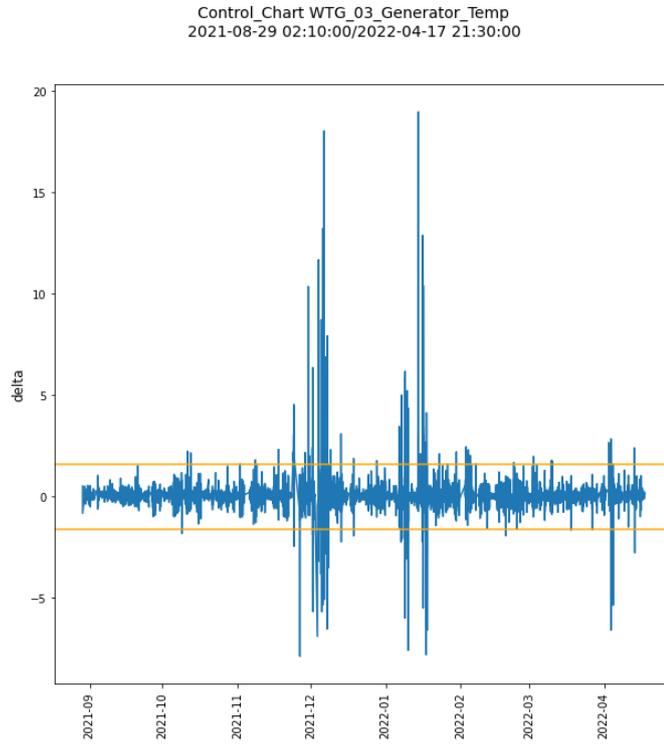


Figure 5.25. Control chart of  $\Delta$  generator temperature for WTG 3.

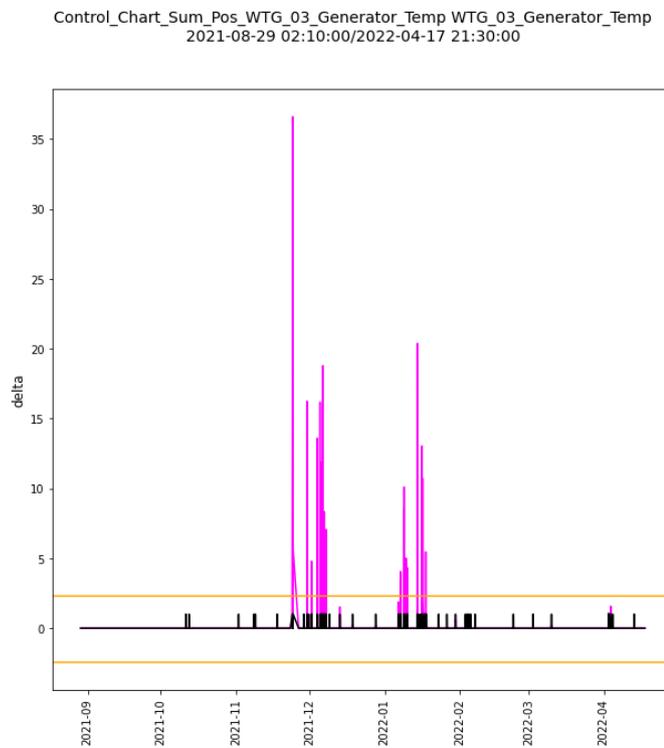


Figure 5.26. Sum of  $\Delta_{th}$  control chart for positive  $\Delta$  values calculated for generator temperature for WTG 3.

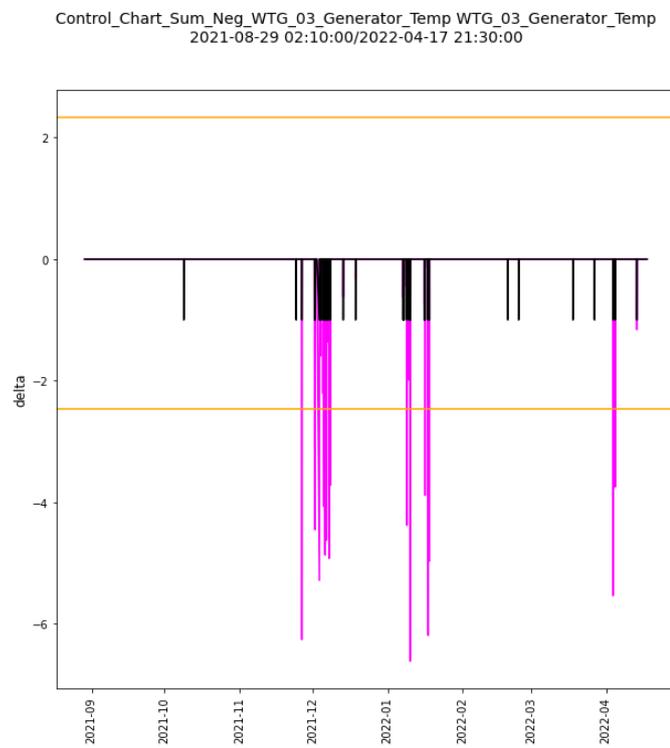


Figure 5.27. Sum of  $\Delta_{th}$  control chart for negative  $\Delta$  values calculated for generator temperature for WTG 3 .

## WTG 5

Analysis of the standard control chart did not immediately detect any abnormal patterns, but analysis using the Sum of  $\Delta_{th}$  method found abnormal behavior with too high a temperature in the mid-October period.

Figure 5.28 shows a comparison of delta values with a standard control chart, instead, Figures 5.29 shows the generation of alarms through the Sum of  $\Delta_{th}$  chart for positive outliers. Alarms were generated during the following period:

- Start: **2021-10-18 21:40:00** End: **2021-10-20 16:10:00**.

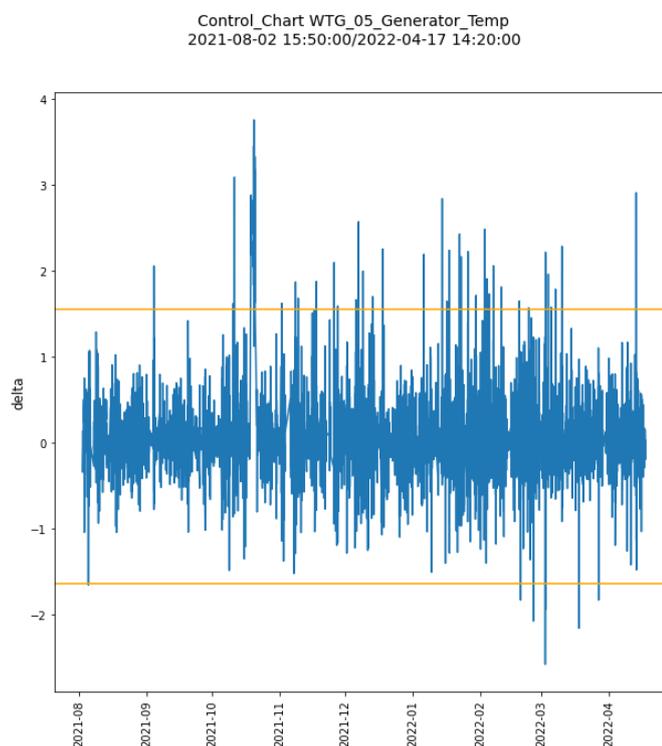


Figure 5.28. Control chart of  $\Delta$  generator temperature for WTG 5.

The analysis of the delta/output graph (Figure 5.30) allows verifying that most of the outliers are positive and relate to higher generator temperatures, so the generator reaches too high temperatures, and it heats more than it should.

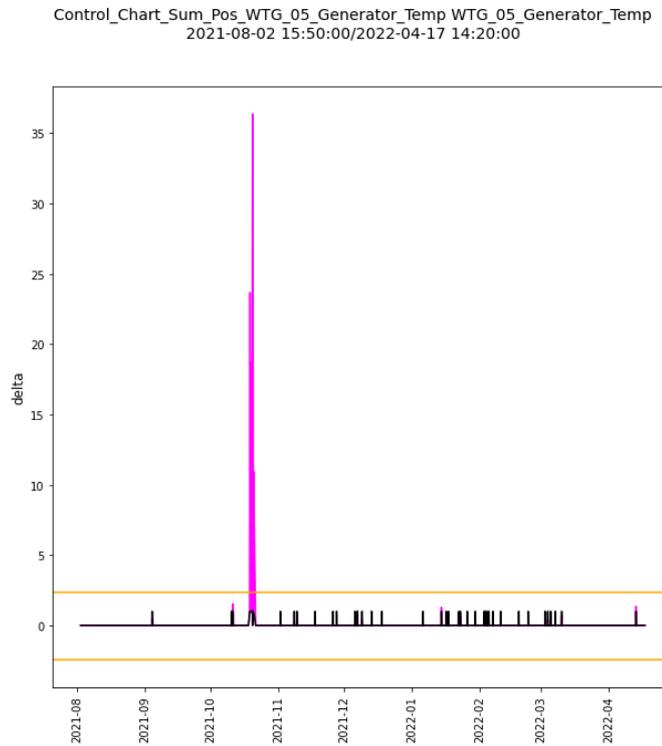


Figure 5.29. Sum of  $\Delta_{th}$  control chart for positive  $\Delta$  values calculated for generator temperature for WTG 5.

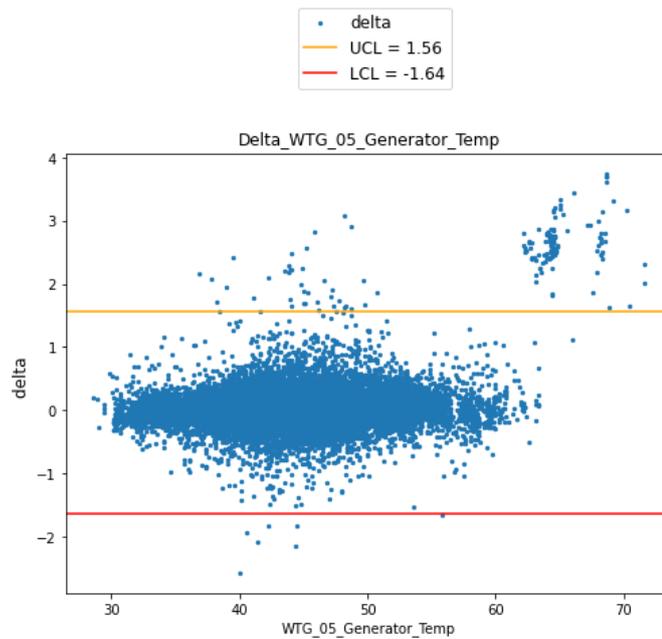


Figure 5.30. Delta/Output chart of  $\Delta$  generator temperature for each output generator temperature of WTG 5.

# Chapter 6

## Conclusions

### 6.1 Discussion

Through the use of ANNs and different statistic control tools, the system developed in this thesis was able to anticipate some alarms reported by the Downtime Manager or otherwise find them after the fact. It also highlighted other anomalous behaviors which are not reported. The results of this project show an obvious malfunction of WTG5 and its components. The model phase is essential for training the networks and should be repeated when analyzing new turbine models and/or new wind farms.

### 6.2 Real-time Prediction

The control phase was implemented to test the system, using datasets. For real-time operation, in order to use the system on a wind farm, the code should be adapted. Specifically, each sample of the individual instant should be filtered, then:

- if the model is based on the use of FNN, a single sample of the current instant should be inserted into the network to predict the output,
- if the model is based on the use of LSTM, a FIFO queue of storage of values in the last 2 hours is made to allow the creation, at each instant, of the correct rolling windows to be inserted into the network.

To apply statistical control with the developed charts and then generate the alarms, additional appropriate FIFO queues with the actual and predicted values of the outputs also should be implemented.

### 6.3 Future Works

In the future, it may be of interest to evaluate the possibility of using other types of RNN such as Gated Recurrent Units (GRUs), or different approaches such as Decision Trees. In addition, neural networks with a single output have been created, but multi-output solutions may be of interest. Finally, the tests were carried out with a data set limited to about one year and using a single wind farm. Using data from multiple years and multiple wind farms, the accuracy of forecasts could be refined, and seasonal traits could be better learned. In addition, the presence of a fault report or multiple alarms belonging to the same category could help in improving the mode, and in selecting the best-performing control charts.

# Bibliography

- [1] Sirius, <https://www.sirius.to.it/>
- [2] Wikipedia, "Multilayer perceptron", [https://en.wikipedia.org/wiki/Multilayer\\_perceptron](https://en.wikipedia.org/wiki/Multilayer_perceptron)
- [3] Van Rossum G, Drake Jr FL, "Python reference manual". Centrum voor Wiskunde en Informatica Amsterdam, 1995, <https://www.python.org/>
- [4] Fortinet, <https://www.fortinet.com>
- [5] NVIDIA, Gamme GeForce RTX 3090, <https://www.nvidia.com/fr-fr/geforce/graphics-cards/30-series/rtx-3060-3060ti/>
- [6] Raybaut, P. "Spyder-documentation". Available online at: [pythonhosted.org](http://pythonhosted.org), 2009, <https://www.spyder-ide.org/>
- [7] Chollet, F. & others "Keras". Available at: <https://github.com/fchollet/keras>, 2015, <https://keras.io>
- [8] J. D. Hunter, "Matplotlib: A 2D Graphics Environment, in Computing in Science & Engineering" vol. 9, no. 3 pp. 90-95, May-June 2007, doi: 10.1109/MCSE.2007.55
- [9] Travis E Oliphant, "A guide to NumPy" volume 1 Trelgol Publishing USA,2006.
- [10] Anon, "Anaconda Software Distribution", Anaconda Inc, 2020 <https://docs.anaconda.com/>
- [11] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015. Software available from [tensorflow.org](http://tensorflow.org).
- [12] Pedregosa, F. et al., "Scikit-learn: Machine learning in Python. Journal of machine learning research", pp.2825–2830, 2011
- [13] McKinney, W. & others, "Data structures for statistical computing in python. In Proceedings of the 9th Python in Science Conference". pp. 51–56, 2010,
- [14] Waskom, M. et al., "mwaskom/seaborn: v0.8.1 (September 2017), Zenodo.",2017, Available at: <https://doi.org/10.5281/zenodo.883859>.
- [15] VireoX, VireoXPower <https://vireox.com/vireoxpower.php>
- [16] Lucas Bauer & Silvio Matysik, "Wind-turbine-models.com, Gamesa G90", <https://it.wind-turbine-models.com/turbines/763-gamesa-g90>
- [17] Irandoost, Hamid, "Fault Diagnosis of a Variable-Speed Wind Turbine via Support Vector Machines". [Laurea magistrale], Università di Bologna, Corso di Studio in Ingegneria dell'automazione [LM-DM270], 2018, <http://amslaurea.unibo.it/view/cds/CDS0931/>
- [18] Youtube, "Cosa sono le TURBINE EOLICHE e come producono l'elettricità? Energia eolica, Componenti, Incidenti", uploaded by JAES Company, <https://www.youtube.com/watch?v=y7wD5uajZBI>
- [19] Ni Engineer Ambitiously, "Wind Turbine Control Method", 2022, <https://www.ni.com/fr-fr/innovations/white-papers/08/wind-turbine-control-methods.html>
- [20] Stefano Carnazzi, LifeGate, "Cos'è una turbina eolica. Funzionamento e dimensioni", 2016, <https://www.lifegate.it/cose-una-turbina-eolica>
- [21] Stephanie Cole, The Roundup, "Wind Turbine Power Curve", <https://theroundup.org/wind-turbine-power-curve/>

- [22] Wikipedia, "Ecart type Wikipedia", <https://fr.wikipedia.org/wiki/>
- [23] Pritha Bhandari, Scribbr, "How to Calculate Standard Deviation (Guide) — Formulas & Examples", 2020, <https://www.scribbr.com/statistics/standard-deviation/>
- [24] Baradaran, Vahid & Dashtbani, Hamideh, "A decision support system for monitoring traffic by statistical control charts". *Management Science Letters*. 4. 10.5267/j.msl.2014.7.019., 2014
- [25] Abiodun Olaoye, Towards Data Science, "Wind energy analytics toolbox: Iterative power curve filter", 2021 <https://towardsdatascience.com/wind-energy-analytics-toolbox-iterative-power-curve-filter-fec258fdb997>
- [26] Zipporah Luna, Geek Culture, "Feature Selection in Machine Learning: Correlation Matrix, Univariate Testing, RFECV", 2021, <https://medium.com/geekculture/feature-selection-in-machine-learning-correlation-matrix-univariate-testing-rfecv-1186168fac12>
- [27] CFI Team, Corporate Finance Institute, "Correlation Matrix", 2022, <https://corporatefinanceinstitute.com/resources/excel/correlation-matrix/>
- [28] SurferToday, "How to read wind direction", <https://www.surfertoday.com/windsurfing/how-to-read-wind-direction>
- [29] Khorava, Liubou; Ny, Ruth Heidi Samuelsen, Norwegian University of Life Sciences, "Stacked LSTM for wind turbine yaw fault forecasting based on SCADA data analysis", 2022
- [30] Kavlakoglu, IBM, "AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?", 2020, <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>
- [31] Kate Strachnyi, Analytics Vidhya, "Brief History of Neural Networks", 2019, <https://medium.com/analytics-vidhya/brief-history-of-neural-networks-44c2bf72eec>
- [32] Yigit, Tuncay & Ersoy, Mevlüt, "The Testing of Wireless Local Area Network with ANN". *Global Journal on Technology*. 4. 621-627, 2013
- [33] Burnett, C. M. L, Wikimedia Commons, "Illustration of the topology of a generic Artificial Neural Network (ANN)", <https://commons.wikimedia.org/w/index.php?curid=1496812> GNU Free Documentation License (accessed: 12/02/2022)
- [34] Alex, Boostedml, "Feedforward Neural Networks and Multilayer Perceptrons", 2020, <https://boostedml.com/2020/04/feedforward-neural-networks-and-multilayer-perceptrons.html>
- [35] Domsoria, "RNN, Come funzionano le Recurrent Neural Network", 2019, <https://www.domsoria.com/2019/11/rnn-recurrent-neural-network/>
- [36] Oinkina, Colah's blog, "Understanding LSTM Networks", <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [37] Turing, "Comprehensive Guide to LSTM & RNNs", <https://www.turing.com/kb/comprehensive-guide-to-lstm-rnn>
- [38] Jason Brownlee, Machine Learning Mastery, "How to Configure the Number of Layers and Nodes in a Neural Network", 2019, <https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/>
- [39] Jason Brownlee, Machine Learning Mastery, Machine Learning Mastery, "Overfitting and Underfitting With Machine Learning Algorithms", 2019, <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>
- [40] Keras, Keras Faq, [https://keras.io/getting\\_started/faq/](https://keras.io/getting_started/faq/)
- [41] Jason Brownlee, Machine Learning Master, "Difference between a batch and an epoch in a Neural Network", 2022, <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>
- [42] Keras, Callback API, <https://keras.io/api/callbacks/>
- [43] Google Developers, "Set di addestramento e test: suddivisione dei dati", <https://developers.google.com/machine-learning/crash-course/training-and-test-sets/splitting-data>
- [44] Juan Orozco Villalobos, BrainsToBytes, "Test, training and validation sets", <https://www.brainstobytes.com/test-training-and-validation-sets/>
- [45] Imron Rosyadi, "Time Series Forecasting", 2021, <https://iroyadi.netlify.app/research/time-series-forecasting/>
- [46] Flaticon, "Icon made by Freepik from [www.flaticon.com](http://www.flaticon.com)" [https://www.flaticon.com/free-icon/server\\_689319](https://www.flaticon.com/free-icon/server_689319)

- [47] Flaticon, "Icon made by Freepik from [www.flaticon.com](http://www.flaticon.com)" [https://www.flaticon.com/free-icon/windmill\\_4914939](https://www.flaticon.com/free-icon/windmill_4914939)
- [48] Flaticon, "Icon made by Vectors Tank from [www.flaticon.com](http://www.flaticon.com)" [https://www.flaticon.com/free-icon/neural-network\\_6461819](https://www.flaticon.com/free-icon/neural-network_6461819)
- [49] Flaticon, "Icon made by Smashicons from [www.flaticon.com](http://www.flaticon.com)" [https://www.flaticon.com/free-icon/magnifying-glass\\_3262128](https://www.flaticon.com/free-icon/magnifying-glass_3262128)
- [50] Luu Khanh Tung, "Units in LSTM", <https://tung2389.github.io/coding-note/unitslstm>
- [51] Jason Brownlee, Machine Learning Mastery, "A Gentle Introduction to the Rectified Linear Unit (ReLU)", 2019, <https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>
- [52] Jason Brownlee, Machine Learning Mastery, "Use Early Stopping to Halt the Training of Neural Networks At the Right Time", 2018, <https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/>
- [53] Prashanth Saravanan, Section, "Understanding Loss Functions in Machine Learning", 2021, <https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/>
- [54] Bangalore, P., Letzgus, S., Karlsson, D., and Patriksson, M., "An artificial neural network-based condition monitoring method for wind turbines, with application to the monitoring of the gearbox". *Wind Energ.*, 20: 1421– 1438. doi: 10.1002/we.2102., 2017, <https://onlinelibrary.wiley.com/doi/epdf/10.1002/we.2102>
- [55] Santolamazza, Annalisa & Dadi, Daniele & Introna, Vito., "A Data-Mining Approach for Wind Turbine Fault Detection Based on SCADA Data Analysis Using Artificial Neural Networks". *Energies*. 14. 1845. 10.3390/en14071845., 2021.
- [56] Wisdom Udo, Yar Muhammad, "Data-Driven Predictive Maintenance of Wind Turbine Based on SCADA Data". *IEEE Access ( IF 3.476 )*, 2021, DOI:10.1109/access.2021.3132684