Data-Driven Wind Turbine Power Anomaly Detection Using SCADA

Douwe Gerard Velds

TU Delft

DTU

Data-Driven Wind Turbine Power Anomaly Detection Using SCADA

by

Douwe Gerard Velds

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on 06-10-2023

Thesis committee:	
Chair:	Prof. Dr. Simon Watson (TU Delft)
Supervisors:	Prof. Dr. Simon Watson (TU Delft)
	Dr. Ir. Nikolay Krasimirov Dimitrov (DTU)
External examiner:	Dr. Ir. Pierre-Elouan Rethore
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	December, 2022 - August, 2023
Student number:	4549244 (TU Delft) - s213892 (DTU)

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Aerospace Engineering · Delft University of Technology

Abstract

In the light of global need for renewable energy, wind energy plays a crucial role. Today, the majority of wind turbines are still being built on land. Given this critical role, accurate monitoring methods are needed to fully understand the performance within wind farms. This thesis aims to create a methodology to evaluate the power performance of wind turbines within an onshore wind farm using SCADA data. The primary objective is to identify deviations from the expected power generation patterns by using a multivariate machine learning approach. Due to the terrain complexity of onshore wind farms, a cluster-based approach is used. In total, 15 clusters consisting of 61 turbines have been evaluated. The training data is filtered by applying multiple filters with the objective of creating a normal behaviour model. The evaluation is based on a feedforward multilayer perceptron regressor to predict the power output for a test data set. A sequential methodology is explored to refine the model performance and is then applied to power performance analysis tasks to detect anomalies in the data sets. The results of the final analysis suggest that the approach taken is capable of detecting anomalies in the data. It is shown that in multiple clusters cases of under- or overperformance can be detected.

Acknowledgements

First and foremost, I would like to thank my supervisors, Nikolay Dimitrov and Simon Watson for their guidance throughout the thesis. Your feedback helped me navigate the many challenges I faced in the past months.

Secondly, I want to express my gratitude to my friends, who kept my spirits high over the past few months. A special thanks goes out to my roommate Koen, who made going to DTU in summer less lonely, and managed every morning to wake me up playing the worst music.

Thirdly, I want to thank my family and especially my parents who have supported me throughout the years. After explaining so many times what my master includes, I hope you have a better understanding of what I'm doing now.

Contents

Nomenclature

List of Abbreviations

AD	Anomaly Detection
AEP	Annual Energy Production
ANN	Artificial Neural Network
IEC	International Electrotechnical Commission
IQR	InterQuartile Range
KNN	k-Nearest Neighbours
KPI	Key Performance Indicator
LOF	Local Outlier Factor
MAD	Mean Absolute Deviation
ML	Machine Learning
QRF	Quantile Regression Forests
RF	Random Forrest
SCADA	Supervisory Control And Data Acquisition
SGBRT	Stochastic Gradient Boosted Regression Trees
SHAP	SHapley additive exPlanations

Greek Symbols

μ	Mean	
Φ	Relative humidity	[%]
ρ	Density	[kg/m^3]
σ	Standard deviation	

Latin Symbols

$\hat{y}_i$	Fitted value of variable	[-]
A	Rotor area	[m^2]
B	Barometric pressure	[Pa]
C_p	Power Coefficient	[-]
e	Error or Residual	[kW]
M	Maximum coalition	[-]
P	Power output	[W]
P_w	Vapour pressure	[Pa]
$R(X)$	Variable rank	[-]
R^2	Coefficient of determination	[-]
R_0	Gas constant of dry air	[J/(kg K)]
R_w	Gas constant of water vapour	[J/(kg K)]
T	Temperature	[K]
U	Wind speed	[m/s]
y_i	True value of variable	[-]
z'	Feature collective	[-]
lrd	Local reachability density	[-]

List of Figures

List of Tables

1 Introduction

This chapter provides the general overview of this Master's thesis. In Section 1.1 the significance and motivation of the study are described. This is followed by Section 1.2, where the research questions that this thesis aims to answer are highlighted. Finally, Section 1.3 gives the general structure of the chapters used in this thesis.

1.1. Motivation

In light of the recent energy crisis, several European countries, including Denmark, The Netherlands, Belgium, and Germany, have committed themselves through the Esbjerg agreements. The agreements state a clear goal: to increase the share of offshore wind energy to at least 65 GW by 2030 [1]. This target builds on the broader European Green Deal, in which the European Union as a whole has pledged to increase the total share of offshore wind energy from the current 16 GW to 60 GW in 2030 and in the long term potentially up to 300 GW in 2050 [2].

Given the critical role of wind energy in meeting the growing energy demands, it is important to develop accurate monitoring methods and effectively detect potential underperformance of wind turbines. Underperformance not only leads to lower power output, it also leads to large revenue losses due to their lower Annual Energy Production (AEP) [3].

Although the future of offshore wind energy is promising, the current landscape of wind energy is heavily reliant on onshore wind energy. As of 2021, 81 % of the installed turbines were constructed onshore, contributing approximately 91 % of the total produced wind power in the European Union is produced onshore [4]. It shows the importance of accurate power performance methods for onshore wind turbines.

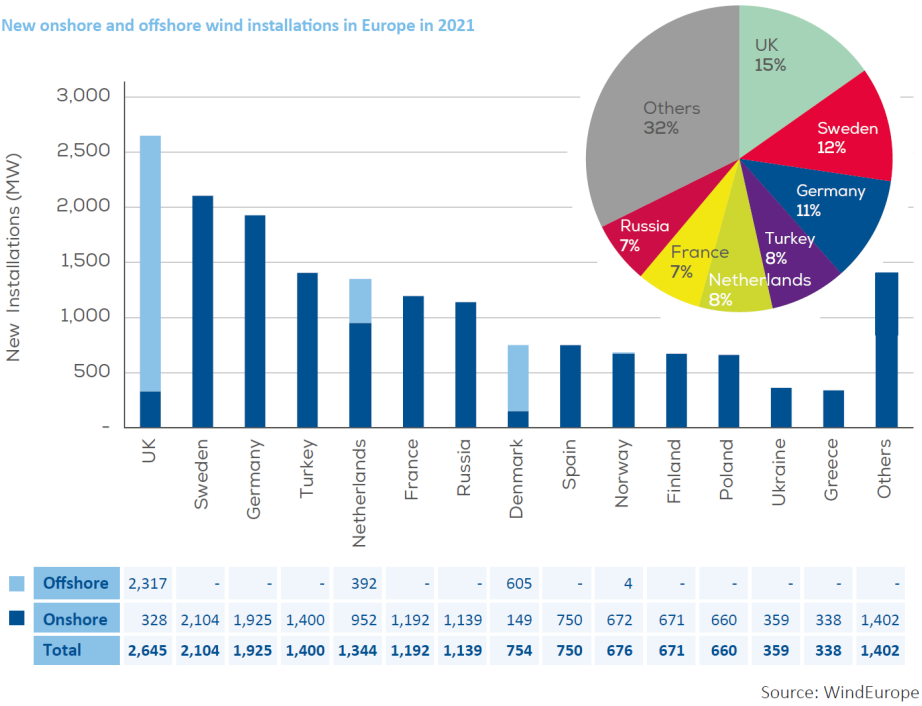


Figure 1.1: Number of wind turbines installed in 2021 [4]

The present study focusses on the evaluation of the power performance evaluation of wind turbines within onshore wind farms. This deliberate choice acknowledges the current reality of the energy landscape, where onshore installations form the majority of the wind energy infrastructure. By focussing on onshore wind farms, this research aims to provide practical insights and solutions that align with the current state of wind energy deployment. Through analysis and investigation, this thesis aims to contribute to the improvement of the efficiency and power output of offshore wind turbines, thus contributing to the broader mission of sustainable energy generation.

1.2. Research Question

This thesis focusses on the viability and validity of multivariate machine learning approaches that use supervisory control and data acquisition (SCADA) data to evaluate the power performance of wind turbines within a wind farm. The primary objective of this study is to assess whether machine learning models are capable of detecting underperformance with a high level of accuracy. To measure accuracy, the study will use mean absolute error and root mean squared error to compare the actual annual energy production with the predicted energy production based on the models. The significance of this research stems from the fact that in recent literature multivariate approaches are typically evaluated on a single wind turbine and are validated by using data from the same turbine. This study will conduct a comparative analysis of discrete approaches with parametric methods. The research questions in this thesis are the following.

1. **How can machine learning (ML) techniques be applied to SCADA data to detect anomalies in power generation within a wind farm?**
   Objective 1: Develop and implement a machine learning algorithm capable of processing SCADA data from multiple wind turbines within a wind farm to detect deviations from expected power generation patterns. Focus on creating models that can be effectively generalised across different turbines.
2. **Can ML improve the accuracy of AEP estimations to detect underperformance in a wind farm?**
   Objective 2: Using machine learning models to predict Annual Energy Production (AEP) for individual wind turbines within the onshore wind farm. Evaluate the model's accuracy by comparing their AEP predictions against the actual energy production.
3. **How can the cause of underperformance be examined within the dataset?**

Develop statistical techniques to analyse the SCADA dataset to identify potential causes of underperformance, considering factors such as environmental conditions, maintenance history, and turbine-specific attributes. Design methods to gain insight from the data that highlight the root causes of deviations in power output.

1.3. Structure

The thesis is divided into seven chapters. In the following section, an overview is given of the material discussed in each chapter and gives a high-level overview of the project's workflow.

In Chapter 2 related work is reviewed. It is divided into three parts, where the first section describes current practises in power curve modelling. Second, methods for detecting anomalies are listed. Finally, the combination of power curve modelling and anomaly detection is highlighted by a section regarding power performance.

Chapter 3 gives an overview of the dataset at hand. First, the layout of the wind farm is described together with the architecture of the SCADA datasets. Second, the distribution of the features in the set of a single turbine is shown. Third, the features are visualised in comparison with the rest of the wind farm. Finally, the down-time across the wind farm is evaluated to prematurely identify problematic wind turbines.

Chapter 4 describes the methodology. It gives an overview of the workflow that has been followed throughout the analysis and provides background information on the applied methods.

Chapter 5 gives the analysis of a single cluster. It lays down the steps taken to identify the performance of the turbines and shows on the hand of figures what the potential causes of the abnormal performance are.

In Chapter 6 an analysis similar to that of a single cluster has been performed accross the wind farm. It shows which turbines in the clusters draw the attention due to significant deviations from expected behaviour.

Finally, Chapter 7 reflects on the work done by summarising the steps taken. Furthermore, it aims to answer the research questions as stated in this introduction.

2

Literature Review

In this chapter, the available literature on power performance has been evaluated. The chapter is divided into three sections. First, the state-of-the-art regarding power curve modelling is described. Second, anomaly detection methods are described. Finally, the power performance section describes how power curve modelling methods can be used together with anomaly detection to detect underperformance in wind turbines or wind farms.

2.1. Power Curve Modelling

Due to the high cost of the O&M department of wind energy companies, the modelling of power curves is an important topic to dedicate studies to. The power curve of a wind turbine is the most commonly used Key Performance Indicator (KPI) since it can be easily read by most users. Furthermore, the change in the measured power curve compared to the predicted power curve can give an early indication of underperformance [5]. Power curve modelling methods can be divided into the following categories: discrete, deterministic, stochastic, parametric, and nonparametric models. This section discusses different power curve modelling techniques used in literature.

2.1.1. Discrete Methods

In order to provide a general method for wind turbine manufacturers, the International Electrotechnical Commission (IEC), has set up a standard IEC 61400 12-1 "*Power performance measurements of electricity producing wind turbines*" [6]. This discrete method is based on the binning method, where the average power is computed in bins of 0.5-1 m/s wind speeds. Since the wind turbines in the farm are pitch controlled, the power curve is based on the normalized wind speed and computed density, which are computed by the formulae described in Eqs. (2.1) to (2.3) [6].

$$\rho_{10\text{min}} = \frac{1}{T_{10\text{min}}} \left(\frac{B_{10\text{min}}}{R_0} + \Phi P_w \left(\frac{1}{R_0} - \frac{1}{R_w} \right) \right) \tag{2.1}$$

$$P_w = 0.0000205 \cdot \exp\{(0.0631846 \cdot T_{10\text{min}})\} \tag{2.2}$$

$$U_n = U_{10\text{min}} \left(\frac{\rho_0}{\rho_{10\text{min}}} \right)^{1/3} \tag{2.3}$$

In Equation 2.1 above ρ is the density, T the temperature, and B the barometric or measured pressure, all averaged over a period of ten minutes. R_0 and R_w are the gas constant of dry air (= 287.05 J/(kg K)) and water vapour (= 461.5 J/(kg K)). Finally Φ describes the relative humidity, which in this wind farm site was found to be approximately 60 % for a city close by. Equation 2.2 describes the vapour pressure P_w, which depends on the mean temperature. Finally, the wind speed, V_n, is normalised, as can be seen in Equation 2.3, by the reference air density ρ_0 and the computed ten-minute average $\rho_{10\text{min}}$.

The main issue with this approach is that it is very site-specific. Furthermore, in the literature, other parameters, such as turbulence intensity [7], have a significant effect on the power curve. Therefore, turbulence intensity normalisation algorithms have been proposed for the IEC method [8].

2.1.2. Deterministic Methods

Deterministic power curve models assume that there exists a fixed relation between power output and wind speed. The most well-known relation is described in Equation 2.4, where the power coefficient C_p in this case is a function of the pitch angle [9].

$$P = C_p \frac{1}{2} \rho V^3 A \tag{2.4}$$

Deterministic models used in power curve modelling can be divided into two main categories: polynomial and sigmoid models. First, in polynomial models the power curve is divided into different regions that can be described by nth order polynomials [10]. Secondly, sigmoid models can either be exponential or algebraic models. In the general form they are described by:

$$P = b_5 + (b_2 - b_5) \frac{(1 + b_6 f(v, b_0, b_1))}{(b_3 + f(v, b_0, b_1))^{1/b_4}} \tag{2.5}$$

Where the b_i values are model parameters and $f(v, b_0, b_1)$ is a function that include model parameters b_0 and b_1. In Villanueva et al. different sigmoid models are compared based on its number of parameters, ease of use and accuracy [10]. The author concludes that consideration of 'best' model is highly dependent on the preference of the user. Taking into account the 3 evaluation criteria mentioned before, it is recommended to apply a 3 or 5 parameter logistic equation model. Further deterministic models include the Blade Element Momentum Equation [9], solving the Navier-Stokes equations [11], and actuator disk models [12].

2.1.3. Machine Learning Methods

Machine learning, which is stochastic in nature, relies on statistical methods to identify patterns and make predictions. This means that the model's output may differ based on the data it is trained on, and there is no guarantee that the model will always produce the same result given the same input. Machine Learning (ML) can be classified into two categories: parametric and non-parametric. In parametric learning, it is assumed that the data follow a particular distribution. The goal is to estimate the parameters of that distribution to build a model. This is done by finding the best fit for the data on the basis of these parameters. On the other hand, nonparametric approaches make no assumptions about the data distribution. The method is intended to learn the distribution of the data itself. Therefore, these methods are effective in solving complex and non-linear problems. In terms of learning, these two categories can be divided into supervised and unsupervised learning. In the following paragraphs, the recent literature on multivariate machine learning approaches for power curve modelling is discussed.

Jannsens et al. [13] stress the importance of accurate performance monitoring. To increase the accuracy of a univariate power curve, six different methods are compared including N-parametric logistic power curve fitting [14], and non-parametric approaches: k-Nearest Neighbours (KNN), Random Forrest (RF) and Stochastic Gradient Boosted Regression Trees (SGBRT). The algorithms are evaluated on a synthetic and a real-world data set and assessed based on the mean absolute error and the root mean squared error. The use of multivariate approaches is found to significantly improve the accuracy of the predicted power compared to using wind speed as the sole input. Incorporating different or additional variables from the SCADA data could be a part of further research.

Gonzalez et al. [15] explore two of the previously described methods in more detail. The Quantile Regression Forests (QRF) approach is compared against two other methods: a nonparametric method of KNN and RF, which is characterised as an ensemble method. The inputs for the three models are the nacelle wind speed, ambient temperature, pitch angle, and rotor speed. The paper confirms the applicability of nonparametric models to model the wind turbine power performance. Moreover, it shows that the empirical approach of the QRF algorithm can take into account the variance of the error term of the residuals.

Ciulia et al [16] propose an ANN with a Genetic Algorithm as optimisation technique. It is shown that the appliance of these neural networks leads to a reduction in standard deviation of the extracted power curve. Furthermore, the ANN allows to incorporate the influence of more parameters compared with the cubic spline method that is also applied in this paper. Both methods show that they can accurately predict the power of the turbine. However, it is concluded that the ANN method can provide higher reliability even

with a limited input data set. More research is needed to validate the applicability on wind turbines under various conditions.

Since power curves rarely describe the standardised conditions described by IEC [6], Pelletier [17] proposes site-specific power curve methods to increase accuracy. Contrary to the previous study, here an artificial neural network (ANN) is applied with the goal of reducing both random and absolute errors. A feed-forward neural network with back-propagation algorithm is applied. The network is trained in two steps to prevent data overload. It is concluded that an ANN model with 6 parameters as input, namely nacelle wind speed, air density, turbulence intensity, wind shear, wind direction, and yaw error, lead to a decrease in mean error compared to nonparametric and discrete models. However, it is only shown for two selected wind turbines in a wind farm, and the results are not validated on different wind turbines at the site.

For obtaining the best results for an ANN model, a prior filtering of a data set is necessary. Manobel et al. [18] propose a Gaussian Process (GP) filtering technique prior to training their ANN model. To benchmark the performance of the GP-ANN model, they make a comparison with the manufacturer, IEC, and a double exponential power curve. The performance of each power curve is calculated based on the root-mean squared error. It is concluded that prior filtering improves the performance for each of the power curves but especially for the ANN power curve. This is due to the fact that there are fewer outliers in the training data. An important note is that the model is purely trained on the basis of the wind speed and direction. In further research, channels can be added to enhance the model's performance.

Barber [19] builds further on methods to improve site-dependent accuracy. He proposes a stochastic, nonparametric use of regression trees in combination with the adaptive boosting method. The advantage of using regression trees is that they are relatively easy to tune and model. Furthermore, due to the non-linear nature of SCADA data, this model is favourable, since it can model complex data based on the input and target variables. However, an important thing to note is that randomly generated combinations are used for the wind speed, turbulence intensity, and shear factors. Therefore, the results do not fully represent the real conditions, which would require further research. It is shown that a regression tree method that can take into account the shear and turbulence intensity can be three times more accurate than the standard binning method [6].

2.2. Anomaly Detection

In this section the state-of-the-art anomaly detection (AD) algorithms are discussed. In short, anomaly detection means the finding of patterns in large sets of data that deviate from the normal behaviour. In some literature, this is used interchangeably with outliers. The detection methods can be divided into three subcategories: statistical, machine learning, or deep mining.

2.2.1. Statistical methods

To detect unusual patterns or outliers in data sets, statistical methods provide powerful tools. The basis of these methods lies in the underlying distribution of the data to detect instances that deviate from expected patterns.

One of the simplest methods is that of the z-score, also known as the standard score, which measures the number of standard deviations that a point is away from the standard deviation. It is assumed that elements x are normally distributed, which describes a bell curve, as can be seen in Figure 2.1. When z scores are compared, points can be identified that deviate significantly from the expected behaviour. To do so confidence intervals can be used, where the amount of standard deviations that an element differs from the mean is calculated. The z score is calculated as described in Equation 2.6 [20].

$$z = \frac{x - \mu}{\sigma} \tag{2.6}$$

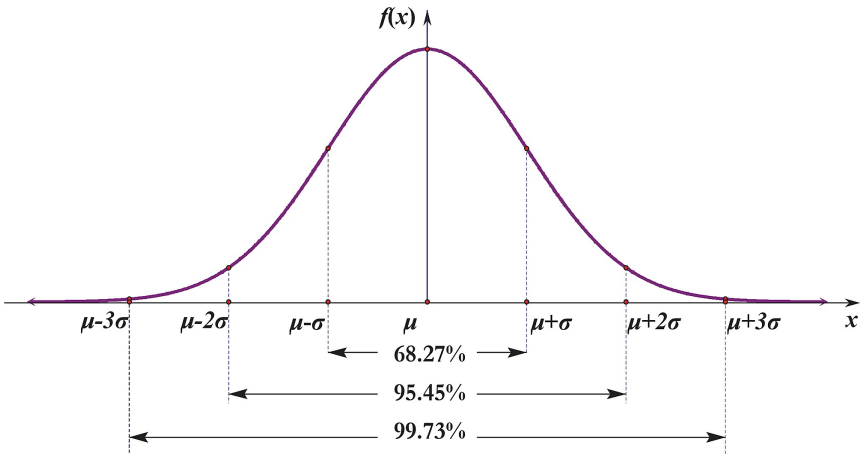


Figure 2.1: Confidence intervals of a typical bell curve [21]

A second group within statistical methods is that of the percentile methods. The data set is divided into equal intervals, each of which contains a certain percentage of data. The most well-known method is that of the InterQuartile Range (IQR), which is the range between the 25th (Q1) and 75th (Q3) percentiles. It is a robust measure of variability and is often visualised in the form of boxplots. Anomalies can be detected by setting a lower limit (L) and an upper limit (U), defined by Equation 2.7, where k is defined by the user and determines the sensitivity of the outliers [20].

$$L = Q1 - k \cdot IQR \tag{2.7}$$
$$U = Q3 + k \cdot IQR \tag{2.8}$$

A second method within the percentile methods is the Mean Absolute Deviation (MAD), which measures the median of the absolute deviations from the median of the data. Anomalies can be detected by comparing absolute deviations with a MAD-based threshold.

In conclusion statistical methods provide a relatively easy yet useful approach to detect anomalies. It is important to note that the effectiveness of these methods often depends on assumptions made about the data distribution. Additionally, an appropriate selection of threshold values should be applied to avoid detecting too many outliers.

2.2.2. Machine Learning

An overview of different existing literature is provided by Chandola et al. [22]. From their work it becomes evident that there is a wide variety of approaches, and it has been applied in several research areas. Depending on the applied technique, assumptions are made between normal and anomalous data. In a more recent survey by Karim et al. [23] a focus is placed on machine learning algorithms. In machine learning, a division can be made between supervised, unsupervised, and reinforced learning. Supervised learning makes use of labelled data points that can be normal or anomalous. The goal of these algorithms is to create a relation between the input and output variables that enables the model to make predictions regarding unseen data points. In contrast, unsupervised learning does not require any specific or labelled input. The goal is therefore to find patterns, relations, or structures within the dataset. Therefore, the use of either of these learning methods is heavily dependent on the dataset at hand. Finally, reinforced learning requires feedback from the environment and is therefore separated from unsupervised learning. The system can be stopped at any desired point in time to observe its behaviour. It is therefore often compared to human learning [24].

Supervised

In terms of supervised algorithms, Support Vector Machines (SVM) have been widely applied in the literature. The algorithm aims to divide the dataset into two distinct categories, where normal behaviour can be considered to be the majority group and anomalies the minority group. The two separate groups are also called support vectors; hence the name. The main idea is to find a hyperplane that separates

data points with maximum margin. One of the advantages of SVMs is their ability to handle both linear and nonlinear data by the use of kernel functions. Depending on the kernel function, the data points are mapped into higher-dimensional feature spaces [24].

A second supervised approach can be grouped as tree based methods. Decision trees are heuristic structures that are built upon choices or comparisons. The advantages of these approaches are the following. First of all it can work directly with non-numeric data which can be beneficial for categorisation problems. Secondly, it handles directly missing data as there is no prediction step required. Thirdly, they can easily scale from linear to non-linear data. Finally, they can be presented as if-then rules, which makes them more interpretable for users[24]. An overview of different tree based methods within anomaly detection is made by Bariol et al. [25].

In the field of power curve anomalies, Moreno et al. [26] compare four different tree based methods to a SVM method and a clustering approach. In order to apply a supervised learning method, they made use of knowledge from the maintenance crew to label abnormal and normal modes. The dataset was divided into four distinct classes: spread, offset, derated, and normal. Firstly the four tree based methods are the following: Boosted Trees (BT), RUSBoosted tree (RBT), RF, and Rotation Forrest (RotF). Secondly, SVM uses a Gaussian kernel function to map the higher-dimensional data. Thirdly, a weighted KNN (w-KNN) was applied with Euclidean distance, which differs from a regular KNN since an added weight is given to data points close to new observations [27]. The inputs for the models are the power and the wind speed. To validate the model performance k-fold cross-validation is applied. The performance of the classifiers is based on Matthews Correlation Coefficient (MCC). Comparing the MCC of all methods the authors show that SVMs can classify anomalies in power production but are less effective compared to other methods. w-KNN was found to be the most effective method with the lowest error. The main issue is the fact that maintenance knowledge has to be applied in order to detect the anomalies limits the appliance of these algorithms. The authors suggest, therefore, that in future work this should be automated.

Unsupervised
As mentioned above, unsupervised learning is based solely on the underlying statistical properties of the data. Therefore, the algorithms aim to identify anomalies based on the characteristics and structures of the data, without relying on predefined class labels. These types of algorithms are described on the basis of a highly influential paper, described in detail below.

Morrison et al. [28] compare four unsupervised methods. Furthermore, they aim to show that prior filtering improves the accuracy of the AD techniques. As a filter, the missing data is divided into three categories defined as: completely at random, at random, and not at random. The difference between the first two categories is that for completely at random there is no correlation with the ”missingness” and any variables, while for at random there is a correlation with a variable. Not at random means that there is a correlation with a variable, but the cause cannot be measured. Proceeding, obvious anomalies are filtered on a rule base which is compared with an unfiltered dataset. The first algorithm applied is that of an isolation forest [29], which is a binary decision tree that focusses on isolating anomalies by measuring the path length of data points. Secondly, Gaussian Mixture Modelling (GMM) was applied, which is a clustering method that assumes that each cluster can be described by a Gaussian distribution. Anomalies are scored based on the distance, usually the Euclidean distance, from the data point to the mean of that cluster. Thirdly, Local Outlier Factor (LOF) [30] calculates the relative density deviation between neighbouring data points. It gives a score for each data point based on the ratio of the local density and the average local density of the neighbouring points. Points with a significantly lower density are considered to be outliers. The downside of this method is that multidimensional data should first be scaled before applying the algorithm. Finally, the authors applied KNN to a labelled dataset in order to compare with other methods. It is concluded that the choice of AD algorithm is highly dependent on what the user wants to achieve and what SCADA data the user has at hand. Furthermore, filtering before applying the first three methods leads to a lower relative error and, therefore, is beneficial.

2.3. Power Performance

This section provides an overview of the available literature on wind turbine and wind farm power performance. Two particular articles will be highlighted due to overlap with the previous sections.

Papatheou et al. [31] explore the potential to use SCADA data to monitor how well the power curves of

wind turbines can predict the power production of other wind turbines. The analysis is based on an MLP regressor neural network compared with Gaussian process regression for each of the turbines in the wind farm. Eventually the predicted power is evaluated based on the wind speed measurements of the other turbines. Standard control charts are presented as the residuals for a wind turbine with a 99% confidence interval. It is concluded that the MLP regressor and GP regression produce similar results in terms of MSE error but that GP regression has the advantage of being able to produce confidence intervals. Future work of their approach is to have multiple features as an input to their models.

Lyons and Göçmen [32] demonstrate a regression-based normal behaviour approach to model the performance of wind turbine power. The SCADA dataset, used in the paper, is 1 Hz, which, due to the high temporal resolution, makes it possible to identify abnormal events on a fine temporal scale. It was found that an ANN regressor possessing both an LSTM and a feedforward layer outperformed all other algorithms. Furthermore, it is shown that a local wind farm scale, where only SCADA features of wind turbines with high correlation with the investigated wind turbine are used, provided superior results compared to a turbine and full wind farm scale. Further improvements can be made by connecting the abnormality-filtration sequence and the residual output of the regression model. By doing so the uncertainties of the model can be minimised. Additionally, the range of focus can be adapted to avoid contamination of the residuals in the rated power range. Furthermore, the neural networks were only trained on the wind speed measurements only; future work could incorporate more channels to create multivariate networks.

2.4. Conclusion

In conclusion, this chapter examined the domain of evaluation of power performance in wind turbines and wind farms by evaluating the existing literature. The chapter was structured into three distinct sections, each contributing to an overall understanding of power curve modelling, anomaly detection methods, and their integration for performance monitoring. First, the importance of power curve modelling was highlighted due to its role as a Key Performance Indicator (KPI) for wind turbine operations and maintenance. Three techniques were discussed, including discrete, deterministic, and machine learning methods. Looking at different models, it is clear that the incorporation of multivariate inputs beyond wind speed greatly improves predictive accuracy. From the comparison of parametric and nonparametric methods, these studies collectively emphasise the complexity in modelling. The choice of which model to use is therefore heavily dependent on the dataset at hand and the application for which it will be used. In the following section, anomaly detection methods are described. These methods aim to identify deviations from the expected behaviour within large datasets, ultimately helping to detect underperformance or anomalies in wind turbines or wind farms. A distinction was made between statistical methods and machine learning models, both supervised and unsupervised. Finally, the findings of the previous sections are combined in power performance assessment. Two notable studies were reviewed that showed the use of advanced algorithms for power performance modeling and anomaly detection. The significance of temporal resolution, feature selection, and integration of abnormality filters is underscored as areas for potential improvement.

3

Data Set Description

In this chapter, the data set used for further analysis is described. First, an overview of the wind farm will be given. Second, the distribution of the features are described. Then, the features

3.1. Wind Farm Description

The data set at hand consists of an onshore windfarm, which is called wind farm A and consists of 61 turbines. The layout of wind farm A can be seen in Figure 3.1. From the layout, it can be seen that the wind turbines are placed on an elevated plateau and close the ridges.

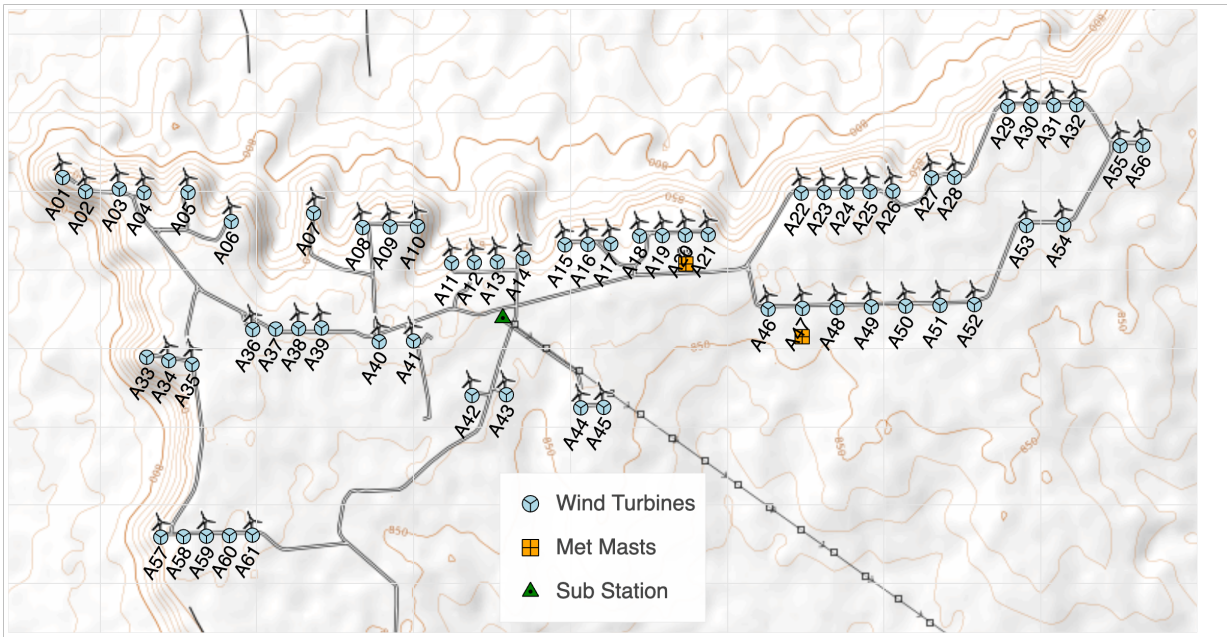


Figure 3.1: Lay-out wind farm A on an orographic chart

The SCADA data for each wind turbine span a period of approximately 2.5 years. For each wind turbine, the information of about 60 channels has been monitored. A summary of the characteristics of wind turbines can be found in Table 3.1.

Table 3.1: General characteristics wind turbine

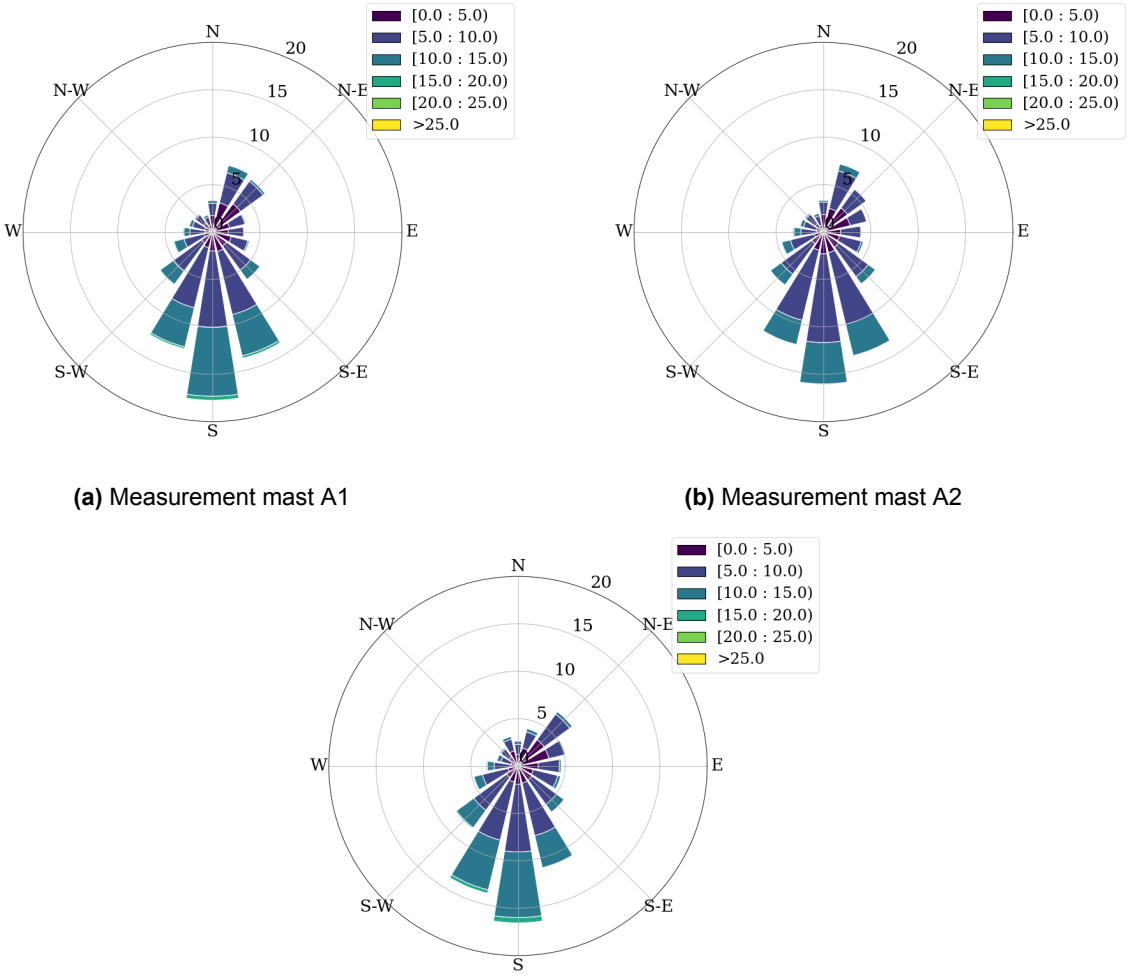
Description	Value	Unit
Rated power	1	[MW]
Control concept	Constant speed power control	[-]
Diameter	61.4	[m]
Hub height	60	[m]
Rotational speed	21	[RPM]
Location	On-shore, rough terrain	[-]

Since the architecture of the SCADA systems differ per wind farm, it is useful to give an overview of the different features or channels that can be used. The features provided in Table 3.2, are often given as a minimum, maximum, mean, and standard deviation over a 10-minute period. To give a concise overview only the main channels are given, but channels with their 10-minute statistics recorded can be found by an asterisk $*$.

Table 3.2: General characteristics wind turbine

Channel	Description	Unit
TStampOfMidPoint	Midpoint of 10-min	Date
DownTimeID	Down time = 0 & No down time = 1	[-]
RefWSpeed	Reference wind speed	[m/s]
RefPressure	Reference pressure	[Pa]
RefWDir	Reference wind direction	[deg]
MaxGust	Maximum gust speed	[m/s]
kW $*$	Active power output	[W]
NacWSpeed $*$	Nacelle wind speed	[m/s]
NacOrientation $*$	Nacelle orientation	[deg]
RotorRPM $*$	Rotational speed	[rpm]
Pitch $*$	Pitch angle	[deg]
SecsOnline	Number of seconds the turbine is available	[s]
PowerFactor $*$	Power factor	[-]
WindDirDiff $*$	Yaw error	[deg]
AmbientDeg $*$	Ambient temperature	[$^\circ$C]

In addition to the wind turbine data, site A provides environmental information from two meteorological masts and one mast in site B. The masts provide information regarding wind direction and wind speeds at hub height. The wind rose for these three measurement masts can be seen in Figure 3.2. From the wind roses it can be seen that for both farms there is a dominant southerly wind. Looking at the wind farm lay-outs this means that there are very little wake effects for this wind direction. The average wind speed on farm A is found to be around 7 m/s and for farm B around 7.4 m/s.



(a) Measurement mast A1

(b) Measurement mast A2

(c) Measurement mast B1

Figure 3.2: Wind roses plots for all measurement masts

3.2. Feature Distributions

The understanding of the underlying structure and distribution features in the SCADA data gives a deep understanding of what the data are that are examined. In this section, a comprehensive view is given through a scatter matrix, shown in Figure 3.3. It visualises the distribution of the features and shows the pairwise relation. The plot gives a useful tool to uncover patterns and gain insight into the overall characteristics.

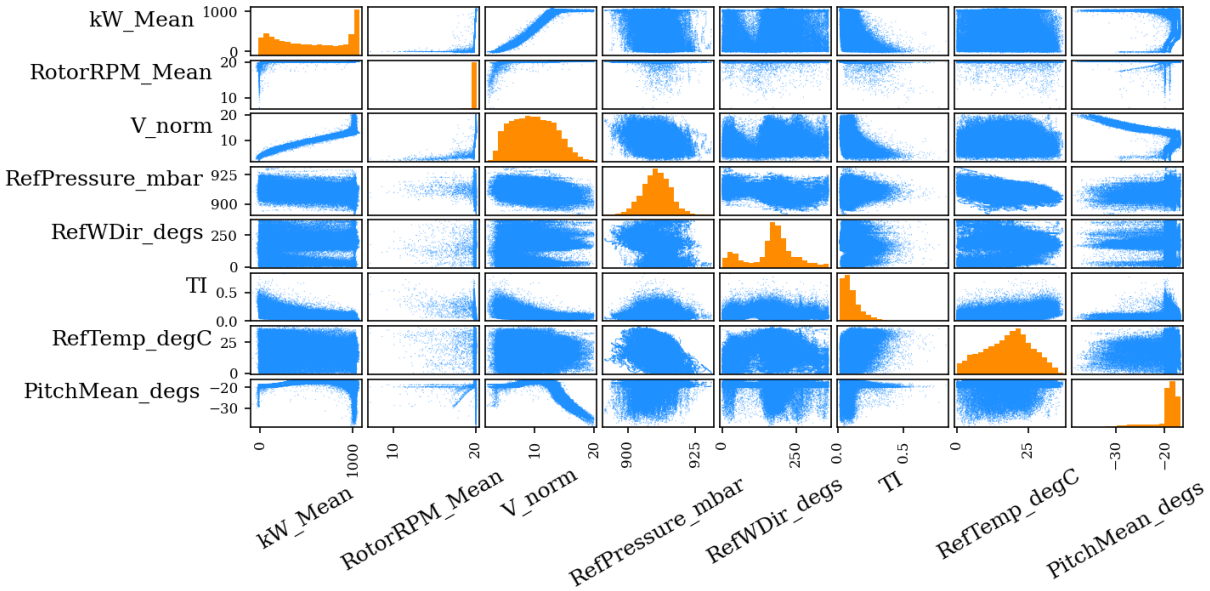


Figure 3.3: Scatter matrix of selected features in the SCADA data sets

In Figure 3.3, the operational curves can be seen. First, the power curve is the relation between the wind speed, V_norm, and the active power output, kW_Mean. Second, the relation between pitch and wind speed, where it can be seen that the pitch angle remains constant up to the rated wind speed. Afterward, the pitch angle gradually increases in magnitude to keep the rotational speed constant. Looking at the distribution of the wind direction, a similar trend can be seen as described in Figure 3.2. At wind turbines, a dominant wind direction from the south is found. As expected the atmospheric measurements, pressure, and temperature, are described by a Gaussian distribution.

3.3. Feature Visualisation

Due to the terrain characteristics of these particular wind farms, a closer look is taken into account of the characteristics and the effect that the location of the turbine has on the features.

3.3.1. Power Output

First, the average power production per turbine is compared to the average for the whole wind farm, which is computed by Equation 3.1. This is done to have an overview of the turbines that might perform better than others.

$$\% \text{ Difference} = \frac{\overline{P_i}}{\overline{P_{WF}}} - 1 \tag{3.1}$$

Where $\overline{P_i}$ is the average power of turbine i and $\overline{P_{WF}}$ the average power of the wind farm. The sign of the percentage difference is an indication of whether a turbine has a higher or a lower energy output. For wind farm A this is shown in Figure 3.4 and a similar visualisation for wind farm B can be found in Appendix A.

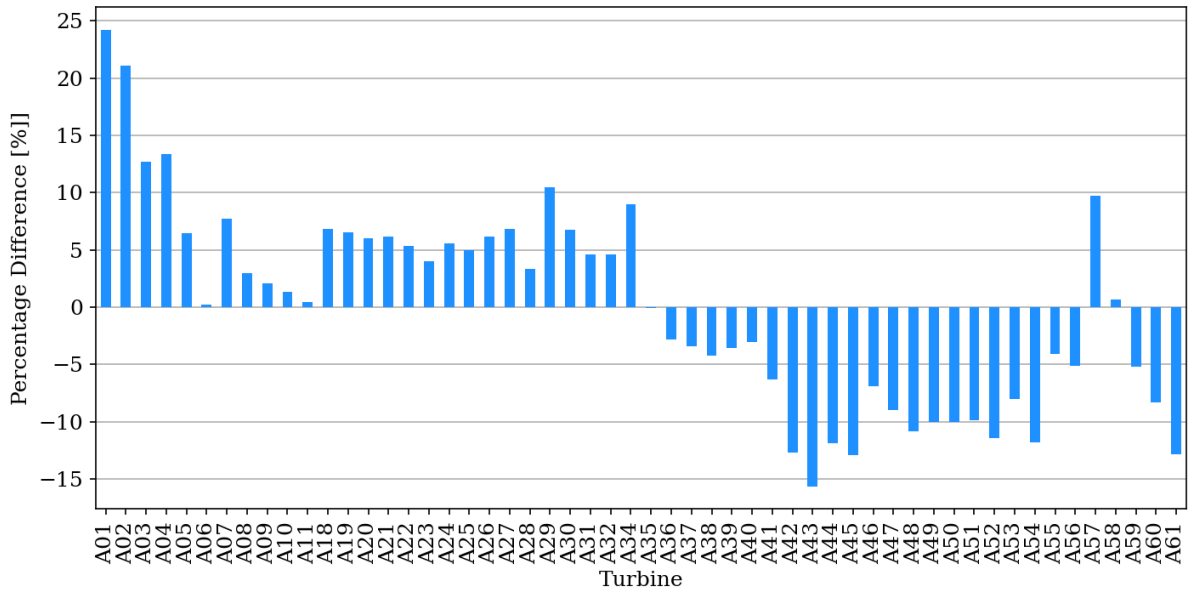


Figure 3.4: Percentage difference of the power output of turbines in wind farm A

From Figure 3.4 it becomes clear that the average power differs largely over the whole farm while being of the same wind turbine type. A high percentage difference in the positive direction can be found for turbine A01 and in the negative direction for turbine A43. Putting this in relation to Figure 3.1, it can be seen that turbine A01 is located near a mountain ridge, while turbine A43 is built in the middle of the ridge. The difference in roughness of the terrain makes it difficult to compare these two turbines. For this reason, the turbines on the wind farm should be clustered with other turbines that have similar conditions. Looking at other turbines that have a negative percentage difference, they are all located on the plateau further away from the mountain ridges.

3.3.2. Wind Speed

Knowing that the roughness of the terrain has an influence on the wind speed measurements, a nacelle wind speed transfer function might have to be applied [33]. It is unknown, however, whether a nacelle transfer function has already been applied in the control system. To evaluate whether this is the case, the normalised nacelle wind speed is plotted with respect to the reference wind speed from the meteorological measurement mast. From Figure 3.5 it becomes clear that a transfer function has indeed been applied to the control system, since a linear relation between both variables is found, indicated by the red line. To know how well the regression line describes this relation, the coefficient of determination is found to be approximately 0.99.

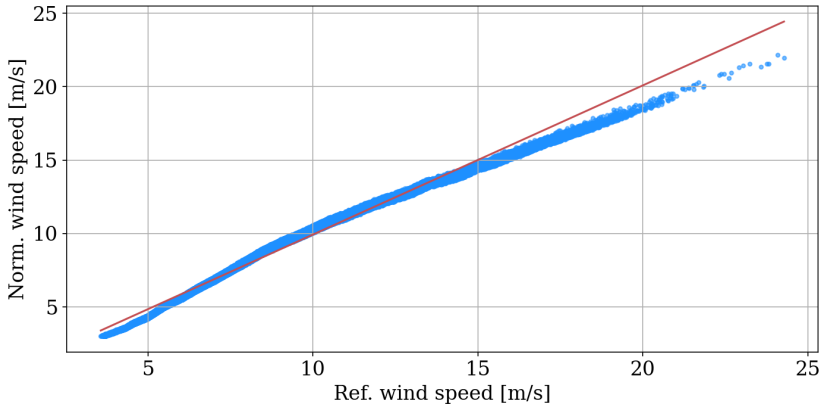


Figure 3.5: Normalised wind speed versus reference wind speed

3.4. Downtime Evaluation

Downtime is referred to as the period that wind turbines are not operating, which has an impact on the energy production of the wind farm. This section aims to provide an overview of turbine downtime, as depicted in Figure 3.6. The factors of downtime have been divided into seven categories as described in the legend of the plot. For all turbines, the main downtime description is credited to the manufacturer. In essence, this contribution is awarded to all component failures, which is known as corrective maintenance. On the other hand, preventive maintenance can be put into the category of maintenance credit, which is the reason why the percentage is in the same range for all turbines. Finally, an important aspect to point out is that of the downtime due to ”owner”, this is related lightning damage that some of the turbines occurred. Knowing that there is a very high percentage of downtime in turbines A24 and A55, special attention should be paid to these. The same evaluation for wind farm B is performed and can be found in Appendix A.

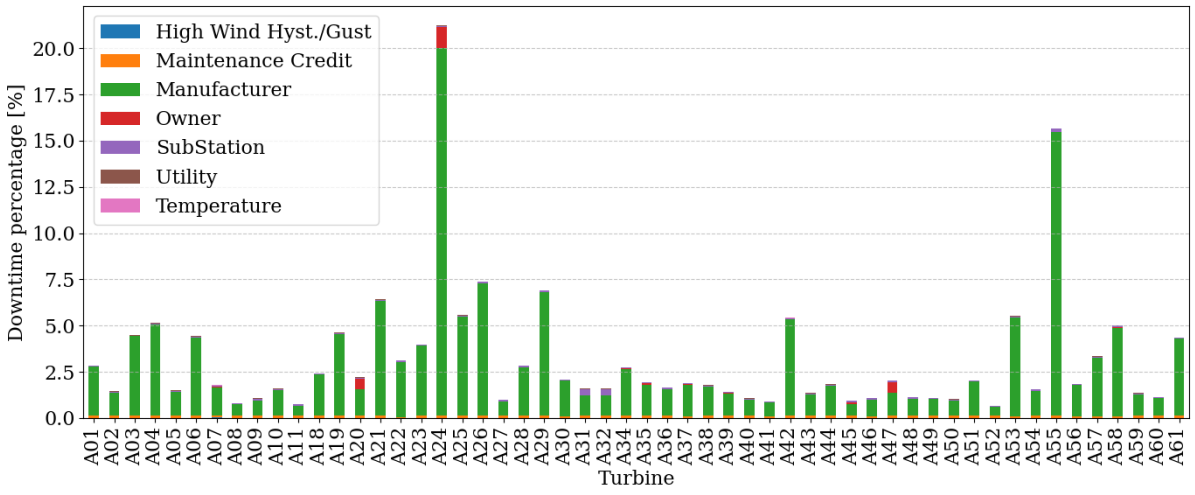


Figure 3.6: Downtime description of wind farm A

4

Methodology

This chapter describes the methodology for detecting underperformance in wind turbines. It consists of six sections. The first K-means clustering is described, followed by feature extraction. Third, the data filtering process is described. Fourth, a detailed explanation of artificial neural networks is given. Fifth, the model evaluation is described as performance metrics. Finally, the Shapley values are explained and their application in this analysis. An overview of the methodology work flow can be found in Figure 4.1.

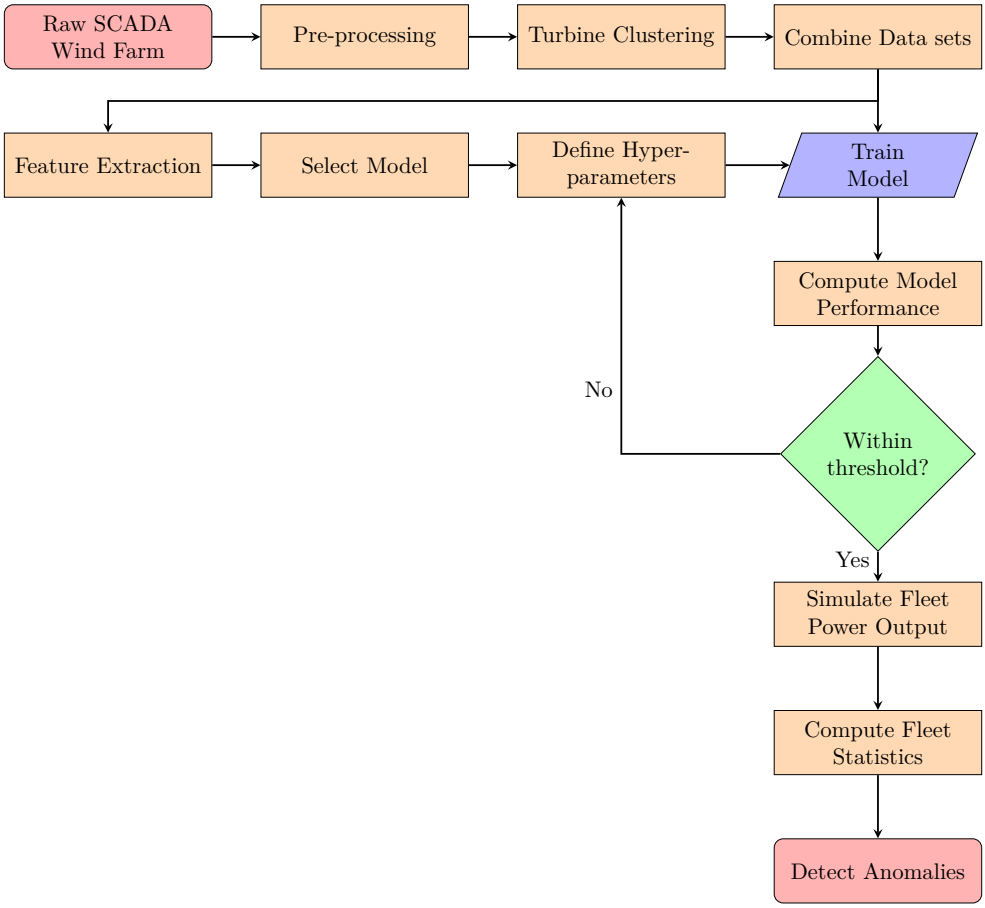


Figure 4.1: Work flow of the applied algorithm

4.1. Clustering

The first step in the approach is to cluster the turbines in the wind farm according to the measured wind speed at the nacelle. Clustering is achieved by making use of K-means clustering, an unsupervised machine learning technique. The technique aims to divide a given data set into K distinct clusters. The

inputs of the data set are the time series of each turbine on the farm. The K-means clustering algorithm works as follows [24]:

1. Start with a value of K, where K is the amount of clusters to be found in the cluster.
2. Randomly initialise the clusters so that there are no duplicates.
3. Assign each of the training samples to one of the k cluster centres based on the Euclidean distance metric.
4. Update centres of each cluster as the mean of all samples
5. Repeat steps 2-4 until there are no changes in cluster centre location.

To evaluate the appropriate number of clusters, the elbow method is applied. The method is based on the inertia or mean squared error (MSE), shown in Equation 4.1 [34].

$$MSE = \frac{\sum_{i=0}^{n} \min_{\mu_j \in C} \left(\|x_i - \mu_j\|^2 \right)}{N} \tag{4.1}$$

Where μ_j is the mean of the samples in the cluster. The aim of the method is to find the so-called elbow point. It is the point where adding more cluster centres does not significantly improve the inertial value and therefore provides the optimal number of clusters.

The implementation of this algorithm is performed using the *Clustering* package within *scikit-learn*[34] [ADD VERSION]. To initialise the cluster centroids, Step 2 in the work flow, *k-means++* is used. This method, proposed by Arthur and Vassilvitskii [35], only initialises the centroid of the first cluster at random. The following amount (K-1) of cluster centroids is chosen based on a probability proportional to the distance from the already chosen centroids. By doing so, a more spread-out initialisation is ensured to avoid getting stuck in a suboptimal solution. The outcome of the method is described further in Section 6.1.

4.2. Filtering

To describe the normal behaviour of a wind turbine, the data should be filtered before going to the training phase. It is essential to ensure that the data is of sufficient quality [36]. In the following subsections, the data filters are described.

4.2.1. Primary Filters

An important step before training the model is pre-processing. This includes removal of Not-a-Number (NaN) and primary filtering. To avoid removing too much data by removing all NaNs, the channels containing the largest amount are monitored. Second, primary filtering has been applied based on four channels: seconds online, temperature, wind speed, and standard deviation (SD) of wind speed. The reason for creating these filters is based on the conclusions of Morisson et al. [28]. Here, it is shown that the application of adequate filters can lead to an increasing performance of the model. Filter values and their relative impact on the total length of the data set are listed in Table 4.1. Note that the impacts of the filters are shown as a range over the entire wind farm.

Table 4.1: Filter approaches and their percentage impact on the full dataset

Channel	Filter	Impact
NaNs	Built-in	1-2%
Downtime	Labeled	1-10%
Seconds online	< 570 [s]	15-24%
Temperature	$\geq$ 2 [$^{\circ}C$]	$\pm$ 5 %
Wind speed	$\leq$ 25 [m/s]	$\pm$ 0.02%
Wind speed SD	$\leq$ 1E-4 [m/s]	$\pm$ 1%

As a first step a pre-processing is performed on the 10-min average data to eliminate invalid points. To evaluate whether the data can be considered to be useful, the percentage of NaN values is monitored for

each wind turbine. To avoid removing too much data by blindly discarding NaN values, the columns that contain the largest percentage of NaNs are evaluated. If the column is considered to be useful for the problem, the whole row at that timestamp is discarded. The three channels leading to NaN values are the reference wind direction, the exported energy, and the reference pressure. Second, the downtime is excluded, since the turbine does not produce any energy while performing maintenance.

Further data filter approaches are listed in Table 4.1. First, the online status of a wind turbine during operation is defined as any measurement between cut-in and cut-out wind speeds [6]. However, to avoid taking into account 10 minute statistics during which the turbine was not online during the entirety of this period, a filter has been applied at 95% of this time, thus 570 seconds. Second, a filter is applied to exclude temperatures below 2 $^{\circ}C$. This is done because blade icing can reduce power production up to 17 % [37]. Furthermore, icing can lead to failure or degradation of test equipment and is therefore recommended to be rejected [33]. Third, the cut-out wind speed for this type of turbine is 25 m/s; therefore, the power output above this limit will not be taken into account. This leads to an exclusion of 0.02% of the data. Finally, data are removed where the measured standard deviation of the nacelle wind speed is below 1E-4, since this is considered to be an error in the cup measurements.

4.2.2. Local Outlier Factor

Local Outlier Factor (LOF) is an unsupervised outlier detection algorithm that was first introduced by Breunig et al. [30]. It is a relative density-based technique, which means that it uses a local approach to identify how far the points are separated in relation to their neighbouring points k, which is visualised in Figure 4.2.

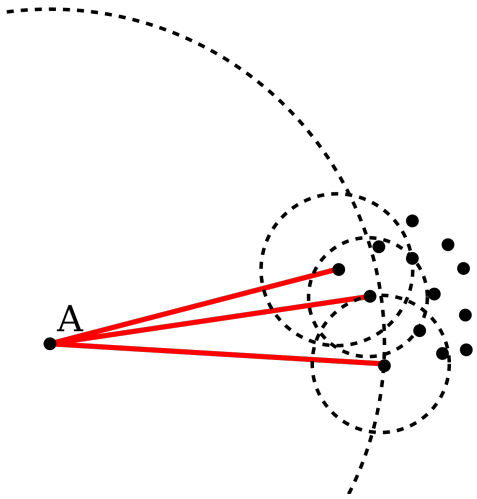


Figure 4.2: Visualisation of local outlier factor[38]

As a first step, the number of nearest neighbours, *MinPts*, has to be defined. The local reachability density (lrd) for a point p can then be defined as the ratio of the sum of distances from the point to its neighbours and the number of neighbours, which can be seen in Equation 4.2 [30]. The distance, reach-dist, between points can be calculated in various ways, but the Euclidean distance is most commonly used.

$$\text{lrd}_{MinPts}(p) = 1/\left(\frac{\sum_{o \in N_{MinPts}(p)} \text{reach-dist}_{MinPts}(p,o)}{\|N_{MinPts}(p)\|}\right) \tag{4.2}$$

To capture the degree of which object p can be called an outlier, the average ratio between the lrd of point p and the lrd of its neighbouring points has to be calculated. This can be seen in Equation 4.3 [30].

$$\text{LOF}_{MinPts}(p) = \frac{\sum_{o \in N_{MinPts}(p)} \frac{\text{lrd}_{MinPts}(o)}{\text{lrd}_{MinPts}(p)}}{|N_{MinPts}(p)|} \tag{4.3}$$

According to Breunig et al. [30] most instances should have a value of one. If the value is larger than one, it is likely to be an outlier. For values smaller than one, the data points are most likely to be in a more

densely populated region. It is important to note that for multidimensional features, they have to be scaled before running the algorithm to account for the different dimensions of features in distance calculations.

In the *sklearn.neighbors.LocalOutlierFactor* package the *MinPts* can be defined as the number of neighbours. By default, this is set to twenty but has been optimised manually; however, it is recommended to take a value that is larger than ten [30]. Second, the contamination can be set, which is the proportion of outliers in the data set. To avoid removing too much data from the original dataset, this was set to 0.005. The parameters used are summarised in Table 4.2

Table 4.2: Parameters used in LOF algorithm

Parameter	Value
Neighbours	50
Contamination	0.005
Distance	Euclidean

Applying this to one of the power curves in the wind farm provides a good indication of which data points are removed from the data set. Figure 4.3 shows the power curve of turbine A03 where the outliers are represented by the red dots.

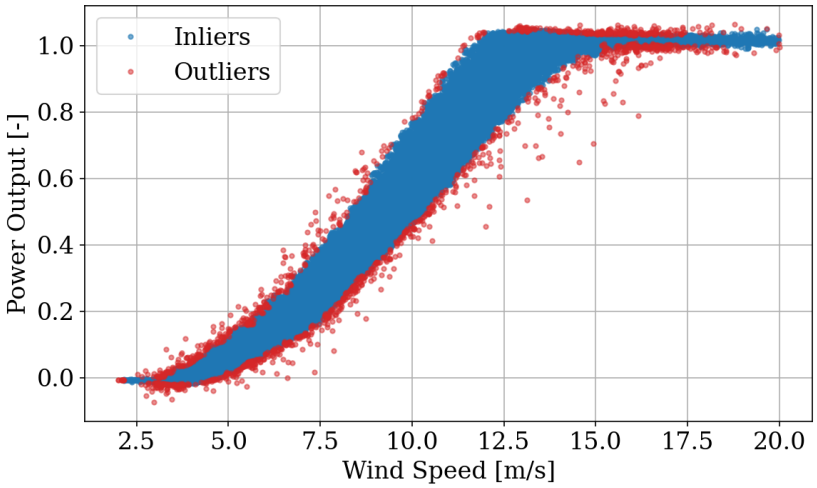


Figure 4.3: Power curve of wind turbine A03 where the outliers are marked in red

Combining all the filters described above gives us the data set that will be used for training. Figure 4.4 shows the unfiltered power curve on the right side and the power curve after filtering on the left side. The filtered data describe our normal behaviour model.

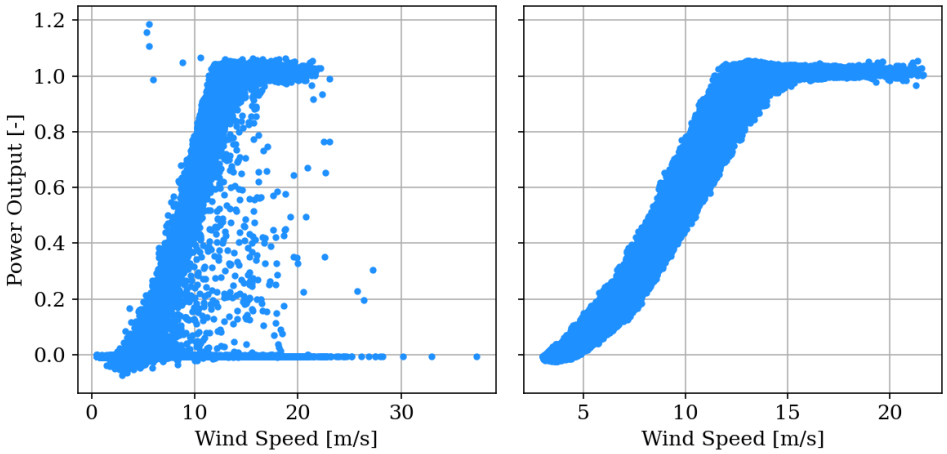


Figure 4.4: Unfiltered power curve (left) and fully filtered power curve (right)

4.3. Feature Extraction

An important step in data-driven anomaly detection is feature extraction. In this case, the Spearman correlation is applied to identify the most relevant variables. Spearman correlation is a statistical measure that quantifies the relation between variables. For two samples $\{X_1, X_2, \ldots, X_n\}$ and $\{Y_1, Y_2, \ldots, Y_n\}$ of size *n*, the rank of each element X_i is determined based on its comparative value. In other words, $R_{X_i} = 1$ for the smallest value and $R_{X_i} = n$ for the largest value. The correlation coefficient ρ_s is calculated based on the rank of the variables $R(X)$ and $R(Y)$, which is defined by Equation 4.4 [39].

$$\rho_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n\left(n^2 - 1\right)} \quad \text{where } d_i = R_{X_i} - R_{Y_i} \tag{4.4}$$

If multiple several elements have exactly the same value, an average rank is given. If this is the case for multiple instances, a correction must be made, which is described by Eqs. (4.5) to (4.7) [39].

$$\rho_s = \frac{S_X + S_Y - \sum_{i=1}^{n} d_i^2}{2\sqrt{S_X \cdot S_Y}} \tag{4.5}$$

$$S_X = \frac{n\left(n^2 - 1\right) - \sum_{i=1}^{g}\left(t_i^3 - t_i\right)}{12} \tag{4.6}$$

$$S_y = \frac{n\left(n^2 - 1\right) - \sum_{i=1}^{h}\left(t_j^3 - t_j\right)}{12} \tag{4.7}$$

The coefficients can range between a value of -1 and 1, where -1 & 1 denote perfect monotonic relation and 0 no association. The benefit of using Spearman over Pearson's correlation is that the Spearman method is more robust to outliers since it does not rely on the actual value of the variables. Furthermore, it is advantageous when dealing with nonlinear relations between variables [40]. The Spearman method is applied in the *pandas.corr* package,

4.4. Artificial Neural Network (ANN)

Based on the literature study, an ANN is implemented. To predict the power output of wind turbines, a Multilayer Perceptron (MLP) algorithm with a forward scheme and ReLU activation functions is applied. It is a supervised nonparametric machine learning algorithm that performs a linear regression. As the name indicates, the network consists of multiple layers of nodes or neurones, as can be seen in Figure 4.5.

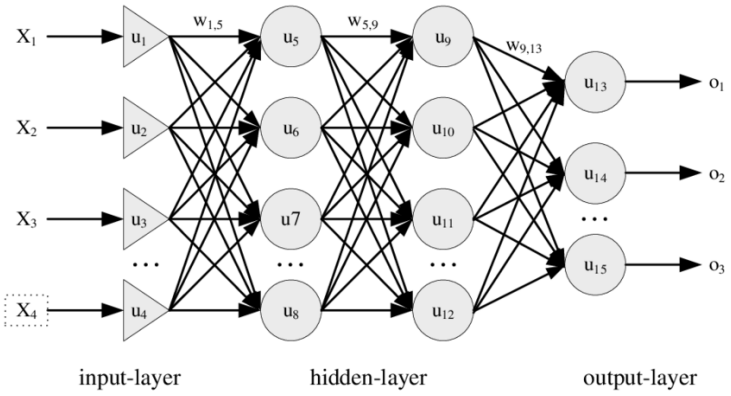


Figure 4.5: Architecture of a typical MLP algorithm [41]

4.4.1. Framework

The results of this thesis are produced mainly using *Python 3.10.9* and *PyTorch 1.12.1*. The code was run on an Apple M1 8GB during training.

4.4.2. Data Split

As a first step before training the model, the data set is split into a training, validation, and test data set, with a ratio 70%, 20% and 10%, respectively. The test set consists of the final three months of the available SCADA data. This is done to verify how well the model is able to predict for unseen data. The model is then trained over a defined number of epochs. Each epoch uses the entire training dataset to update the models parameters based on the computed gradients and the evaluated training and validation losses. Within each epoch, the training data are divided into minibatches of optimised sample size. The minibatches are then trained by computing the forward pass, backward pass, and optimisation step.

4.4.3. Activation Functions

The input layer consists of the same number of nodes as the number of input features. The input layer is connected to the hidden layers with an assigned weight, which is the impact that a value has on the subsequent layer. The information is passed through each node by applying an activation function. An activation function introduces nonlinearity into the system. Several activation functions exist and for continuous problems the hyperbolic tangent (*tanh*), sigmoid, rectified linear unit (*ReLU*), and softplus are most commonly used [24]:

- **Sigmoid:** is an S-shaped and non-linear activation function. It converts the input between negative and positive infinity to a value between 0 and 1 through Equation 4.8. Therefore, it is suitable for binary classification tasks or for shallow networks such as functions simulations. The downside is that it has a vanishing gradient problem towards 0 and 1. This means that the network learns either slowly or does not learn at all. Secondly, due to its function, it has nonzero centrality, thus it always leads to a positive result. Therefore, the training process takes more steps to converge and increases the run-time [42].

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \tag{4.8}$$

- **Hyperbolic tangent:** is commonly denoted by tanh. It is defined as the ratio of the hyperbolic sine function to the hyperbolic cosine function, which can be found in Equation 4.9. Unlike the sigmoid function, the output of the function always ranges between -1 and 1. Due to the shape of the function, it also suffers from a vanishing gradient towards -1 and 1.

$$f(x) = \tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4.9}$$

- **Rectified linear unit:** is a segmented linear function, meaning it is composed of straight-line segments, which can be seen from Equation 4.10. It is also known as the ramp function because

of its shape. It solves the vanishing gradient problem present in functions like sigmoid and tanh, allowing for faster and more efficient training and is the most popular activation function currently used [42]. One of the downsides of ReLU is that it is prone to the dying ReLU problem. This means that some neurones get stuck in a zero or inactive state and do not recover during training. To solve this issue, other forms of ReLU have been suggested, such as Leaky ReLU [43] and parametric ReLU.

$$f(x) = \max(0, x) \tag{4.10}$$

- **Softplus:** is an alternative to the ReLU activation function to avoid the dying ReLU problem. The function, described by Equation 4.11, produces a smooth curve that asymptotically approaches zero for negative inputs and increases linearly for positive inputs. The downside of this function is the higher computational time.

$$f(x) = \text{softplus}(x) = \ln(1 + e^x) \tag{4.11}$$

The shape of each activation function can be seen in Figure 4.6. ReLU is chosen based on the fact that it produces outputs with a relatively high accuracy and a low computational time [44].

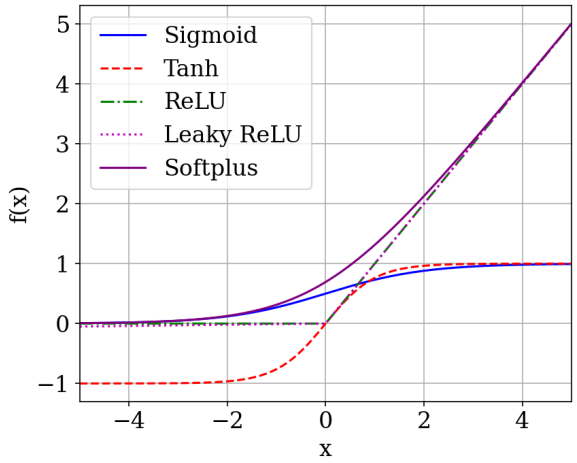


Figure 4.6: Activation functions commonly used

4.4.4. Optimizers

The purpose of an optimiser is to find the optimum weight of each node that minimises the error of the output. The choice of optimiser has an effect on both the accuracy and the training time of the model. The optimisers most commonly used are the following:

- **Stochastic Gradient Descent (SGD):** is a form of gradient descent that updates the weights of each individual point after an iteration. As each individual weight has to be updated, it has a high variance in the loss function, however, it converges in a lower amount of time compared to a regular gradient descent. A variation on SGD is mini-batch gradient descent, where instead of updating each individual point, each mini-batch is updated after an iteration. The use of SGD is beneficial when the expected learning path contains multiple local minima or when the training size is too large to use it all in one iteration [24].
- **Root-mean square propagation (RMSprop):** is an adaptive learning rate algorithm specifically designed to improve the limitations of SGD [45]. It adapts the learning rate of each parameter by dividing it by the running average of the magnitude of previous gradient observed in mini-batches. By doing so, it effectively normalises the updates, making it more stable and robust.
- **Adaptive moment estimation (Adam):** is an adaptive learning rate optimisation algorithm proposed by Kingma and Ba [46]. It combines two other optimisation techniques, momentum and RMSprop. The Adam solver maintains the moving average of the past gradients by means of a first-order

momentum and the second-order moment of the squared gradients. By doing so, the learning rate can be adaptively scaled for each parameter. The method is robust in handling sparse gradients and different learning rate requirements for various model parameters. Furthermore, it has a low memory requirement and a fast convergence time compared to other solvers.

The Adam solver is chosen because of the fast convergence and low memory requirement as described before. To make a comparison between RMSprop and Adam a neural network with the same hyperparameters is trained, which can be seen in Figure 4.7. As described earlier, the cost function of the Adam optimiser clearly shows a smoother and faster convergence to a stable solution.

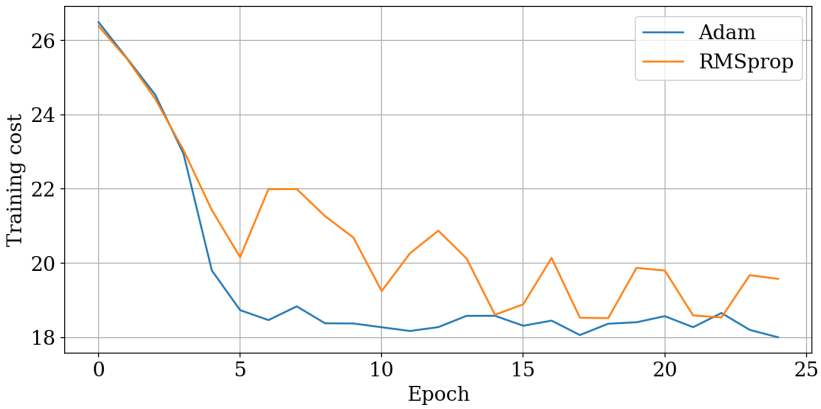


Figure 4.7: Comparison between RMSprop and Adam for the same hyperparameter inputs

4.4.5. Loss Functions

In regression task, the loss or regularisation function determines how the discrepancy between the predicted and actual target values is calculated. The choice of an adequate loss function is essential because it impacts the model's ability to learn and make accurate predictions. The following three loss functions are most commonly used for regression purposes:

- **Mean Squared Error (MSE):** is the loss function that is used the most. It penalises large errors more by the squaring operation as can be seen in Equation 4.1, in this case the error is defined as $\|y_i - \hat{y}_i\|^2$ however, which changes the equation slightly. This makes it more sensitive to outliers and encourages the model to focus on reducing errors across the entire range of predictions.
- **Mean Absolute Error (MAE:** computes the average absolute error between the target and the predicted values, as can be seen in Equation 4.12. Since this error is not squared, as for the MSE, it is less sensitive for outliers. The downside of this, however, is that it may lack the ability to detect patterns in the data because all errors are weighted equally.

$$MAE = \frac{\sum_{i=1}^{N} \|y_i - \hat{y}_i\|}{N_{\text{samples}}} \tag{4.12}$$

- **Huber loss:** is hybrid loss function that combines both the characteristics of the MSE and MAE. The unreduced loss for a batch size N can be described as $L = \{l_1, l_2, ..., l_N\}^T$. The loss l_n for each element in the batch is defined by the following equation [47].

$$l_N = \begin{cases} \frac{(y_i - \hat{y}_i)^2}{2} & \text{for } \|y_i - \hat{y}_i\| \leq \delta, \\ \delta \cdot \left(\|y_i - \hat{y}_i\| - \frac{\delta}{2}\right) & \text{otherwise} \end{cases} \tag{4.13}$$

It makes use of the MSE for small errors and the MAE for large errors, as can be seen in Equation 4.13. In this way, it is less sensitive to outliers, while retaining a smooth loss surface for efficient optimisation. Choosing the right value of the additional hyperparameter δ depends on the specific characteristics of the data and, therefore, can be challenging.

The optimisation function used in this regressor is the MAE. The presence of potential outliers in the data sets could lead to a vanishing gradient when using the MSE. The additional hyperparameter tuning of δ requires more information about the data and can be a time consuming operation. It is implemented with the *pytorch* library as $nn.L1loss$.

4.4.6. Hyperparameter Optimisation

The optimisation of the hyperparameters is a crucial step in making an accurate and reliable ANN. The performance of the network depends heavily on the chosen parameters, and therefore, an optimisation should be carried out. The following optimisation techniques are commonly used:

- **Grid Search:** in this technique an exhaustive search is performed to find the optimum combination of parameters. The user defines the values of the hyperparameters that can be tuned from which a grid is formed. All combinations in the grid are explored, and the performance is stored in a matrix. From this matrix, the best combination can be chosen.
- **Random Search:** contrary to the grid search, the exact values of the hyperparameters are not defined by the user. Instead, a statistical distribution for the parameters is defined. This allows a broader exploration of the hyperparameter space which is advantageous when there is a large search space.
- **Bayesian Optimization:** is a technique that works by creating a probabilistic model with the objective of optimising its performance based on the combinations of hyperparameters that have been used before. The performance of the algorithm is based on minimising the loss function used in the training phase. Then the probabilistic model is used to make a prediction of the next set of parameters. This process is repeated until the optimal set of parameters is found.

The Bayesian optimisation algorithm has been chosen since it requires less intuition from the user. To implement the algorithm, the *skopt.optimize.gp_minimize* package has been used. It is a Gaussian process to minimise for an objective function, in this case L1-loss of the validation phase of the model.

4.5. Performance Metrics

To add credibility and insight into the residuals of the machine learning model, performance metrics are calculated. Looking at regressors, three metrics can be employed:

- **Root-mean squared error:** measures the average errors between the model's predictions and the target values. It takes into account both bias and variance, through Equation 4.14

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}} \tag{4.14}$$

- **Coefficient of determiniation** also know as the R2-score, assesses the proportion of variance in the target values that the model's prediction try to explain. It gives us an indication of how well the model's predictions match to the actual data variation. This can be computed using Equation 4.15.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2} \tag{4.15}$$

Besides the metrics described above the MAE, described in Equation 4.12, is also monitored as the loss function. By combining these metrics a deeper understanding is gained into the prediction accuracy, robustness to errors and its ability to explain variance.

4.6. Shapley Values

Knowing the performance of the model gives an indication how well the predictive power is. However, it is still uncertain how these predictions are made because the learning method acts as a black box, making it hard to understand the rational behind the predictions. This unknown factor challenges the thrustworthiness of the model and limits it to future applications. To address this problem, Lundberg and Lee [48] introduced SHAP values (SHapley additive exPlanations). It is a powerful concept that finds its

roots in cooperative game theory, which bridges the gap between accuracy and model interpretability. Throughout this section, the main idea behind SHAP values is explained and how they are applied to explain the model's predictions.

4.6.1. Background

The purpose of SHAP is to explain the prediction of an instance, called x, by computing the contribution of each of the features used to make the prediction. The SHAP explanation computes the SHAP values based on the theory of cooperative game theory, where the feature values act as so-called players in a coalition. To see the contribution of the players, a simpler explanation model, called g, must be used, which is defined as an interpretable approximation of the original model, called f. The explanation model g can be defined as a function described in Equation 4.16 [48].

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{4.16}$$

Where $z' \in \{0,1\}^M$ are the simplified features and M is the maximum coalition. $\phi_i \in \mathbb{R}$ signifies the feature attribution for a feature i. The collective z' can be explained as follows. When there is an entry of one to the feature space, it means that it is part of the coalition. If, on the other hand, a zero is entered, it means that the feature is not a part. To calculate the shapley values for the instance of interest x, a linear model can be made for only the features that are present. This simplifies Equation 4.16 to Equation 4.17.

$$g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i \tag{4.17}$$

SHAP can describe the following three desirable properties:

- **Local accuracy:** ensure that the sum of the SHAP values for a specific instance x equals the difference between the model's output and the expected output, which the average output of the model over all instances. This property can be described by the following equation:

  $$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{4.18}$$

  In Equation 4.18 $f(x)$ represents the model's output, ϕ_0 the base value and $\sum_{i=1}^{M} \phi_i$ sums up the contribution of the individual features.
- **Missingness:** means that the features that are not part of the coallition have an attribution of zero. In essence a missing features can have any shapley value without effecting the local accuracy because it is simply not taken into account. To follow the notation of Lundberg and Lee this is described by Equation 4.19 [48].

  $$x' = 0 \implies \phi_i = 0 \tag{4.19}$$

- **Consistency:** means that if model changes so that the contribution of a feature value increases or stays the same regardless of the other features, the shapley value also describes a similar behavior. This is described by Equation 4.20.

  $$f'_x(z') - f'_x(z'_{-j}) \geq f_x(z') - f_x(z'_{-j}), \qquad \text{for all inputs} z' \in \{0,1\}^M \tag{4.20}$$
  $$\implies \phi_i(f'_x, x) \geq \phi_i(f_x, x)$$

To satisfy the properties described above shapley values, ϕ_i, are calculated. Given M features, the shapley value of feature i is calculated as the average contribution of i accross all coalitions of features. This can be described by Equation 4.21.

$$\phi_i = \sum_{S \subseteq M \setminus \{i\}} \frac{\|S\|!\,(M - \|S\| - 1)!}{M!} \cdot (f(S \cup \{i\}) - f(S)) \tag{4.21}$$

In Equation 4.21 S is a subset of the features excluding feature i, $f(S \cup \{i\})$ is the model's prediction including i and $f(S)$ is the model's prediction of the subset. The calculation of shapley values can capture

the fairness of distribution in attributing the difference in predictions to each feature. Shapley Values' calculation aligns with a principled and interpretable method for attributing feature contributions in machine learning models. By capturing the contributions across different feature coalitions, Shapley Values provide an understanding of feature importance and their influence on predictions.

4.6.2. Kernel Explainer

To apply shapley values to explain the contribution of the features used in the training data, a kernel explainer is used. The Kernel Explainer reduces the complexity of calculating Shapley Values, making it feasible for a wide range of models. The KernelSHAP estimates can be summarised in four steps:

1. Take a sample coalition $z'_k \in \{0,1\}^M$, where $k \in \{1, ..., K\}$.
2. For each instance i, randomly permutate the features in k with those from a reference instance. This forms a coalition where the only difference is the feature you want to attribute. This can be seen by the kernel function in Equation 4.22.

$$\pi_{x'}(z') = \frac{M-1}{\binom{M}{\|z'\|}\|z'\|\|M-z'\|} \tag{4.22}$$

3. Calculate the difference in the model's prediction when the feature of interest is added to the coalition. This is then compared to the prediction without the feature.
4. Compute the weight for each coalition to estimate the Shapley value for the feature.

4.7. Conclusion

This chapter presented a systematic methodology for detecting anomalies in power production of wind turbines. By a combination of data preprocessing, machine learning based power prediction, and shapley values, it provides a solution to address the complexity of wind turbine data analysis. The K-means clustering algorithm allows for effective grouping of turbines based on their wind speed measurements. The data filtering process provides the basis for normal behaviour modelling. The ANN algorithm applied is an MLP regressor consisting of two hidden layers, each connected with relu-activation functions. The model is trained over an optimised number of epochs by computing a forward pass, and the weights of the nodes are updated through a backward pass with the aim of minimising the loss function, in this case the MAE. To evaluate the performance of the model, key metrics such as RMSE, MAE, and R2-score are used. These metrics provide insight into accuracy, ability to explain variance, and robustness to errors. Finally, the integration of Shapley values increases the interpretability of the algorithm. By assigning feature contributions to the predictions, the values give insight into the reasons behind the model's outputs.

5

Cluster Analysis

In this chapter, the results of the methodology section are discussed for a single cluster, namely turbine A01, A02, and A03. The outline of the chapter follows the workflow described in Figure 4.1. Firstly in Section 5.1, features are extracted that are used during the training phase. Secondly, the hyperparameters in the neural network are optimized in Section 5.2. Subsequently, the training process is described in Section 5.4. After the training, the residuals are examined in Section 5.5.

5.1. Feature Extraction

The first step before training an ANN is selecting the features that have a significant influence on the target column. In this extraction process, it is important to know which parameters are directly influenced by the target column. Features such as the exported energy, voltage measurements, power factor, the statistical measurements of the power production, and the power coefficient are therefore excluded from Figure 5.1. Secondly some properties can be included by a combination of the parameters, such as the turbulence intensity (TI) which is a ratio between the wind speed standard deviation (σ_u) and the mean wind speed (μ_u), described by Equation 5.1.

$$TI = \frac{\sigma_u}{\mu_u} \tag{5.1}$$

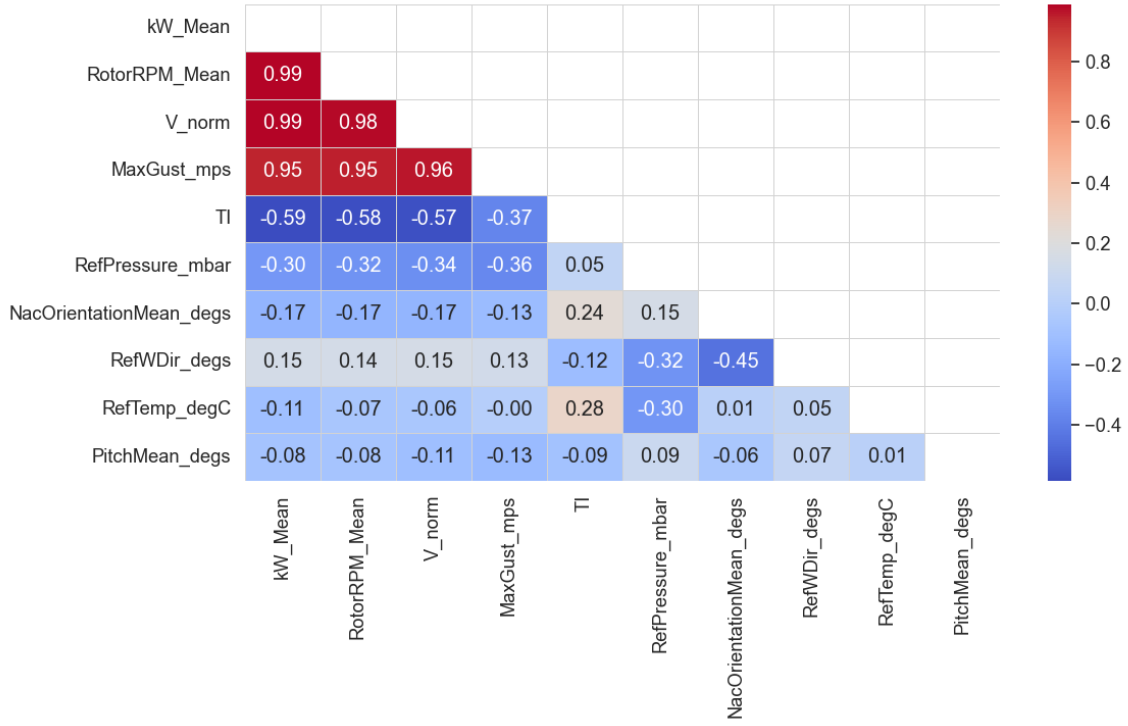


Figure 5.1: Correlation matrix for A01

After analysing the correlation coefficients in Figure 5.1, the features used in the training are summarised in Table 5.1. Since there is a direct relation between the wind speed and the rotational speed of the rotor, the choice is made to exclude the later in the training process. Furthermore, the maximum gust speed has not been included since this merely the ratio between the maximum and the average wind speed during the time interval. To capture the atmospherical influence the air pressure and temperature have been included, since they describe the air density through Equation 2.1. The full correlation matrix including all other features can be found in Figure A.1. The impact of the features on the simulated output is further explained by the means of shapley values in Section 5.6.

Table 5.1: Features used in training

Feature	Channel Name
Normalized wind speed	V_norm
Turbulence intensity	TI
Pitch angle	PitchMean_degs
Wind direction	RefWDir_degs
Temperature	RefTemp_degs
Air pressure	RefPressure_mbar

5.2. Hyperparameter Optimisation

As explained in Section 4.4.6, hyperparameter optimisation is an important step that must be taken before starting the training phase. To perform the Bayesian optimisation technique, the *Skikit-optimise* package in Python was used. The search space for the hyperparameters is the following:

- **Learning rate:** ranging from 1e-4 to 1e-1 in a log-uniform spacing in real numbers.
- **Hidden size 1:** ranging from 100 to 256 in integers.

- **Hidden size 2:** ranging from 100 to 256 in integers.
- **Number of epochs:** ranging from 5 to 50 in integers.

The learning rate defines the rate at which an optimisation algorithm updates the model's parameters during the training process. The hidden size is defined in Figure 4.5 as the number of nodes in a hidden layer. Finally, the number of epochs is the number of times that the entire dataset is exposed to the model. The complete Bayesian optimisation process can be found in Table A.1. In addition to the parameters listed above, the batch size is chosen. The batch size defines the number of elements that the model iterates before the model's gradient is updated. The choice of the batch size is done empirically so that the training curve nicely matches the validation curve. The choice of hyperparameters is summarised in Table 5.2.

Table 5.2: Hyperparameter summary MLP

	Hyperparameter	**Value/Choice**
Hidden layer 1	Type	Fully-connected
	Size	256
	Activation	relu
Hidden layer 2	Type	Fully-connected
	Size	128
	Activation	relu
	Optimizer	Adam
	Epochs	40
	Loss	MAE
	Batch	256

Figure 5.2 represents the contour plots of the combinations of hyperparameters that have been explored during the optimisation process. The black dots in the contour plot represent the values of each combination that has been used. Finally, the red star indicates the best performing combination with respect to the MAE, which is called partial dependence in this case. It can be seen that the learning rate has the largest impact on the MAE. Decreasing the learning rate further than 9e-2 results in an increase of the MAE. Secondly, it can be seen that for both hidden layer 1 and 2, the number of nodes do not have a noticable influence on the MAE. This indicates that the choice of the number of nodes can be made based on other parameters such as the runtime of the algorithm. Finally, as expected, the MAE reduces with an increasing number of epochs. Looking at the contour plot of the number of epochs versus the learning rate it can be seen that the combination of the two is most important. The lowest MAE can be found at a learning rate of approximately 0.02 at 40 epochs.

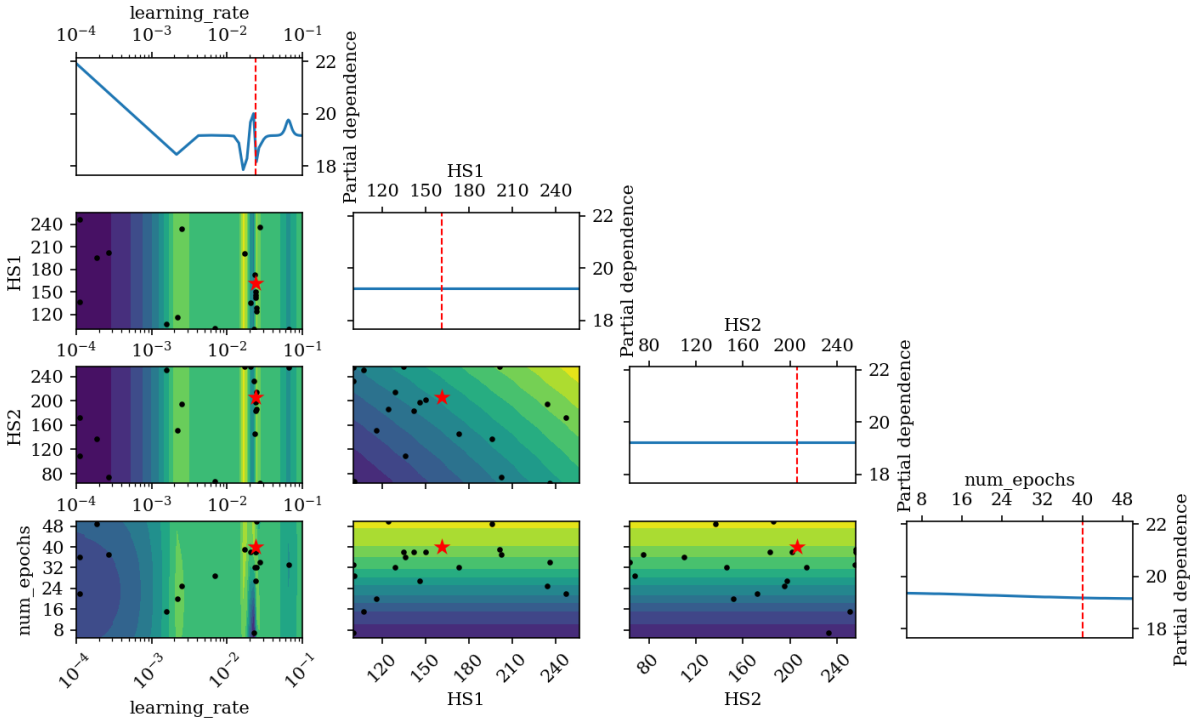


Figure 5.2: Visualisation of the Baysian Optimization steps

5.3. Feature Standardisation

After choosing the features used in the training process, they should be standardised to ensure optimal performance and convergence. MLPs are by nature sensitive to the scale of the input parameters, and when they have widely different ranges, they negatively affect the training process. Standardisation transforms the features to have zero mean and unit variance, which puts all features on a comparable scale. This avoids dominating effects of features with a larger magnitude. Without performing this step, the gradients during backpropagation may lead to slow convergence, as some features could have a disproportional large impact on the weight updates, hindering the model's ability to learn effectively.

The dataset is standardized by the use of the *preprocessing.StandardScaler* package in *sklearn*. Each element in the feature space $\boldsymbol{X}$ is assigned a standard z-score to obtain the standardised feature space $\tilde{\boldsymbol{X}}$, as can be seen in Equation 5.2.

$$\tilde{\boldsymbol{X}} = \frac{\boldsymbol{X} - \mu_{\boldsymbol{X}}}{\sigma_{\boldsymbol{X}}} \tag{5.2}$$

5.4. Training Performance

In this section the performance of the training phase is described. First the loss curve is explained. Second the performance is shown by the means of the predicted versus actual power curve and the performance metrics, described in Section 4.5. Through this analysis a deeper understanding is gained about the underlying patterns in the training process.

5.4.1. Training Process

As described in Chapter 4, the training process takes place over multiple epochs. To gain insight into this training process, the losses, in this case the MAE, for both the training and the validation phase are monitored. In case of overfitting, which means that the model has learned too much from the data, the model has become too specialised to the training data and cannot predict unseen data accurately enough. From the curve an increase in the validation loss would be seen, while the training loss keeps decreasing. In case of underfitting, however, it means that the model does not learn from the training data at all. This can be identified in two ways. First, in case the validation and training loss are not converging to a stable

loss value. If the slope of the losses is still decreasing while the number of epochs increases, it means that the model has insufficiently learned from the data. Second, if the losses are not decreasing regardless of training, it indicates that the model is not suitable to learn from the data. It could also indicate that the model requires an increased complexity to capture the underlying patterns in the data. A third case is that of a good fit, which means that the training and validation loss converge to a point of stability. Furthermore, it should have a small gap between the training and validation curve and show a similar pattern. If the pattern of the losses differ, it is indication that the training or validation data set are not representative for the underlying patterns.

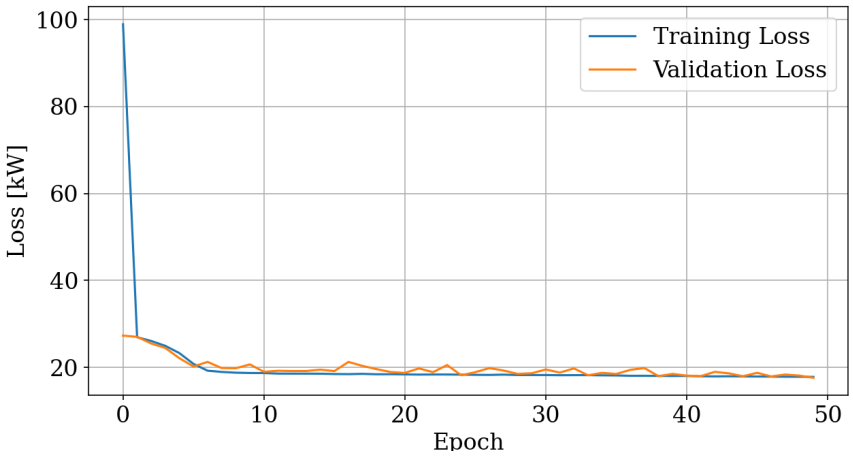
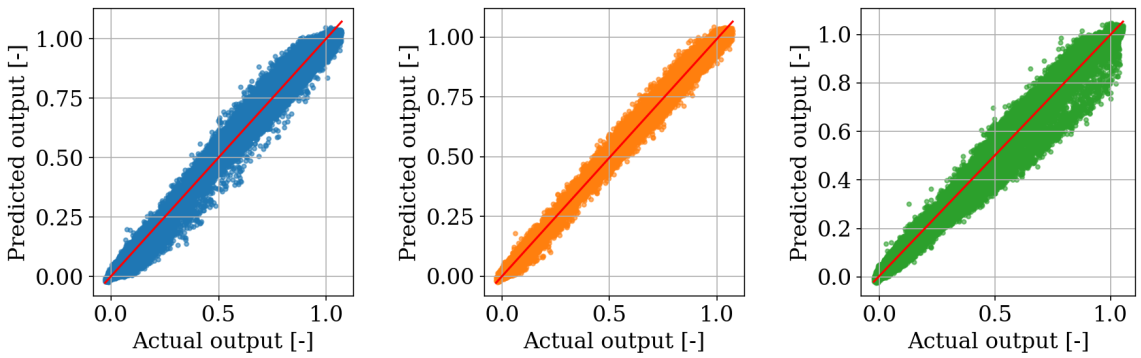


Figure 5.3: Learning curve cluster A01, A02, and A03

With this theory in mind, Figure 5.3 can be inspected. From the plot a similar trend in both loss curves can be seen, which is an indication that the training and validation data are representative for one another. Furthermore, both the training and validation loss are converging to around 17.5 kW. This is indication that the model is learning sufficiently from the training data.

5.4.2. Model Performance

In order to investigate the performance of the MLP regressor a one-to-one correlation plot is created to visualise the predicted power output versus the actual output. This plot is related to the coefficient of determination, as defined in Equation 4.15, where a good fit would mean that the regression line would go through the origin, which is indicated by the red line in Figure 5.4.



(a) Wind turbine A01 **(b)** Wind turbine A02 **(c)** Wind turbine A01

Figure 5.4: One-to-one correlation plots

Inspection of Figure 5.4 gives an indication that a closer look should be taken at Figure 5.4c. Here it can be seen that the predict power is lower than the actual power for a significant amount of the data. To further investigate this, a residual analysis is carried out in Section 5.5.

The coefficients of determination for both training and test sets are evaluated in Table 5.3. Here it can be seen that the model has the worst performance for turbine A03. The reason for this will be shown in the subsequent sections.As explained in Section 4.5, in addition to the R2-score, the RMSE and the MAE play an important role in understanding the behaviour of the model. The RMSE gives more weight to larger errors, while the MAE calculates the absolute errors and thus treats all errors equally regardless of their magnitude. Consequently, RMSE tends to be more sensitive to outliers, while MAE provides a straightforward average error measure.

Table 5.3: Performance metrics of cluster 2

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A01	26.30	17.69	0.995
	A02	19.55	13.90	0.997
	A03	31.61	21.33	0.992
Test	A01	24.79	17.89	0.995
	A02	19.88	14.63	0.997
	A03	31.54	24.97	0.992

5.4.3. Power Curve

As mentioned in Section 2.1, the power curve of a turbine is an important KPI. In Figure 5.5 the curves of turbine A01, A02, and A03 are shown on the same scale. The simulated power curve is binned in steps of 0.5 m/s in which the power is averaged. It can be seen that the curves are almost overlapping each other, confirming the behavior of this type of wind turbine.

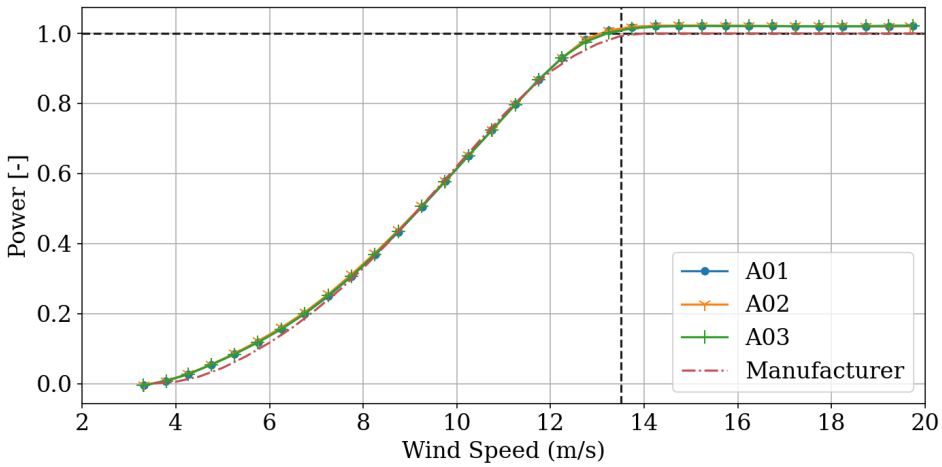


Figure 5.5: Simulated power curve of turbines A01, A02, A03 and the manufacturer

Diving deeper into the power production per turbine helps to identify the wind turbines that have a lower power output than expected. Therefore a kernel density estimation plot is constructed. The aim of this plot is to present the underlying probability density function of the data by binning and counting the observations. In Figure 5.6 it can be seen that the frequency of occurrence of the rated power of turbine A03 is lower than that of the other two turbines.

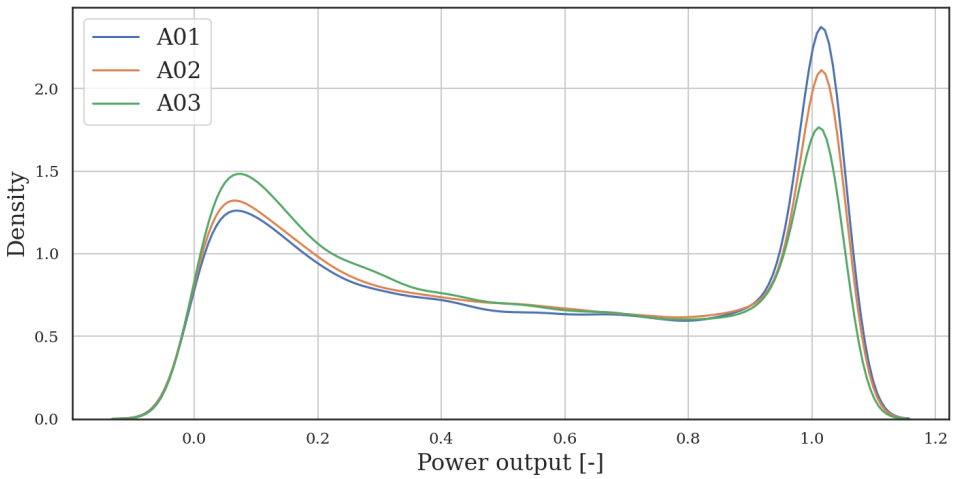


Figure 5.6: Kernel density estimation plot of turbines A01, A02, A03

5.4.4. Power Coefficient

To quantify the aerodynamic performance the power coefficient can be calculated. It is a dimensionless metric of aerodynamic efficiency, which is limited by the Betz limit of 0.59. It tells us how much kinetic energy can be converted into rotational energy. It is limited due to the thrust of the rotor, friction of the blades, flow separation at the trailing edge and other factors [9]. It can be calculated by rearranging Equation 2.4 to Equation 5.3.

$$C_p = \frac{2P}{\rho A U^3} \tag{5.3}$$

In Equation 5.3 P is ten minute averaged power output, U the wind speed, A the area of the disk, and ρ is the density as described in Equation 2.1.

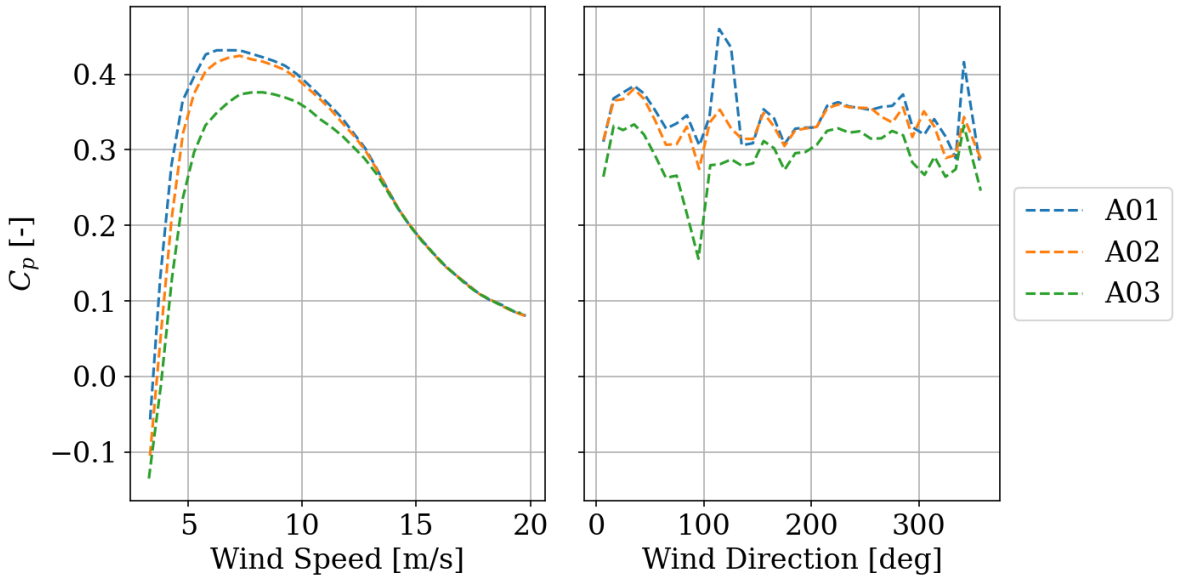


Figure 5.7: Power coefficient for the test set of A01, A02, A03 with respect to windspeed (left) and wind direction (right)

In Figure 5.7 it can be seen that the Betz limit is respected for all three turbines. Furthermore, it becomes clear that the power coefficient of A03 is lower than for the other turbines. When looking at the

relation between the wind direction and the power coefficient, it can be seen that the biggest difference with the other turbines occurs at approximately 100 degrees. Looking at the location of A03, in Figure 3.1, it can be seen that this direction of the wind corresponds to the location of the turbine A04. The spacing between these turbines is around 2.1 diameters, which is approximately 130 metres. This could be an indication that the turbine is experiencing a lower power output due to wake effects. However, it should be noted that a similar trend cannot be found in both turbines A01 and A02 when they are in the wake of other turbines.

5.4.5. AEP Predictions

AEP predictions are an essential tool for estimating the total energy output that a turbine is projected to generate. By quantifying expected energy production, AEP predictions offer a benchmark for evaluating turbine efficiency and output potential. When the predicted AEP aligns closely with the actual output, it indicates that the model accurately captures the underlying dynamics and behaviours of the turbine's power generation. Such validation strengthens confidence in the model's predictive capabilities. Differences between the predicted AEP and actual energy production can indicate potential underperformance issues. Consistently lower energy production than predicted AEP may indicate problems affecting the turbine's efficiency. Figure 5.8 provides such a comparison, by calculating the percentage difference between the actual and predicted AEP of the test test. From the figure, it is clear that turbine A03 is underperforming compared to its predicted energy output, while the other two turbines are actually overperforming.

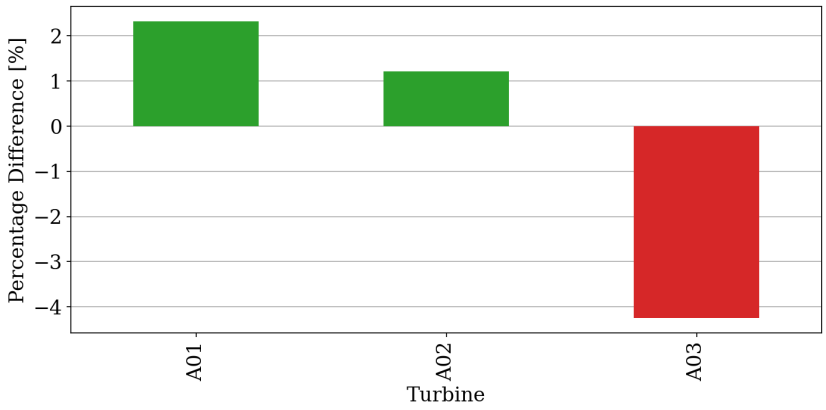


Figure 5.8: Bar plot of the percentage difference in AEP of cluster A01, A02, and A03

5.5. Residual Analysis

Now that the performance of the MLP regressor is known, it is important to see what the difference is between the predicted versus the actual output which is known as residuals. Residuals are a crucial part to gain deeper insight into the model's behaviour. This section describes the analysis of these residuals and how they are used to detect underperformance.

5.5.1. Definition

The residuals in analysis of the following subsections are calculated using Equation 5.4.

$$e = y_i - \hat{y}_i \tag{5.4}$$

Where y_i represents the actual value of the target column and $\hat{y}_i$ is the predicted value of the target column. This expression means that when the sign of the residuals is negative, there is an indication of underperformance. In other words, the model has underestimated the target value, resulting in a negative discrepancy between the predicted and actual outcomes. Conversely, a positive residual value indicates an overperformance, thus the model's prediction is higher than the actual value. The difference in sign of the residuals provides a deeper understanding of the model's tendencies to either under- or overperform across the dataset.

5.5.2. Distribution

Before going deeper into the in-depth analysis of power performance, a critical step is to thoroughly examine the distribution of the residuals. The distribution provides a window into the levels of uncertainty The model's training residuals give us an indication of the levels of uncertainty and the potential bias in the training data. When residuals exhibit a larger standard deviation, it indicates a higher uncertainty in the predictions made by the model. Additionally, a bias might imply that the model is consistently under- or overestimating in certain ranges of the data. It can also be an indication of an inherent bias in the training data.

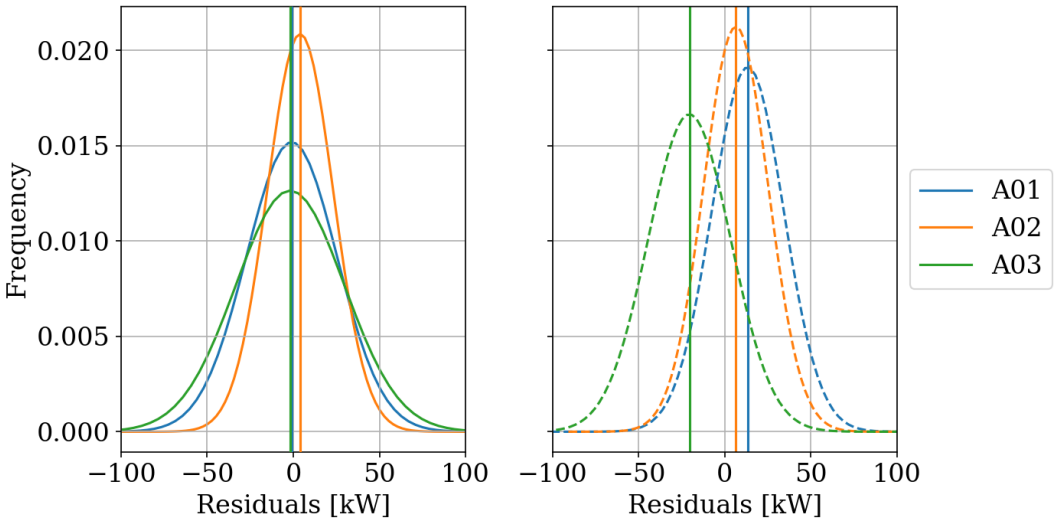


Figure 5.9: Residual distribution for the training set (left) and test set (right) wind turbines A01, A02, and A03

As can be seen in Figure 5.9, the distribution of the three turbines are described by a normal distribution, also known as a bell curve. On the left side the distribution of the training set is shown. It can be seen that the mean of the three turbines is close to zero, which is an indication that the model learns sufficiently from the data. Furthermore, a larger standard deviation for turbine A03 can be seen, indicating that there is a higher uncertainty. When looking at the right curve, where the distribution of the test residuals is shown, it can be seen that a bias is introduced. The mean and standard deviation for both the training and test set are summarized in Table 5.4.

Table 5.4: Residual distribution metrics

	Metric	A01	A02	A03
Train	μ	-0.920	4.013	-1.848
	σ	26.280	19.130	31.553
Test	μ	13.385	6.487	-20.497
	σ	20.867	18.791	23.975

With the distributions of the residuals known, a confidence interval can be constructed based on the standard deviation. It quantifies the uncertainty associated with a point estimate by specifying a range within which the true value is likely to lie. Typically, it is expressed with a certain level of confidence.

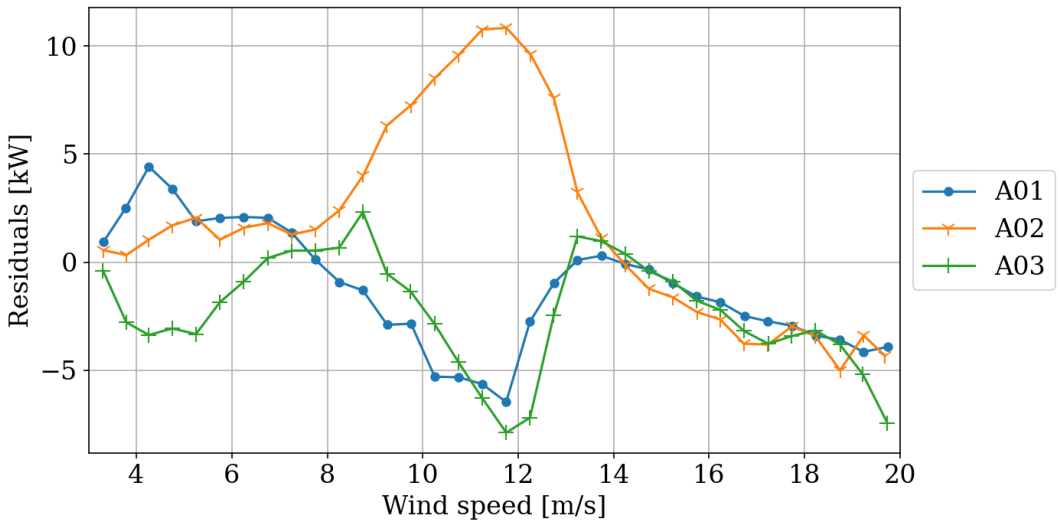


Figure 5.10: Binned residuals for turbines A01, A02, and A03

To know around which wind speed the largest residuals occur, a binned residual plot is shown in Figure 5.10. As for the power curve, the residuals are binned in wind speed steps of 0.5 m/s. It can be seen that the absolute highest values occur towards the rated wind speed, which is at 13.5 m/s. This is an indication that the model is struggling to capture the non-linear around this wind speed. Furthermore, after the rated wind speed, the model tends to be sensitive to small variations in power output, leading to the negative slope.

5.5.3. Time Series

With the predictions of the MLP regressor, it is possible to analyse the time series of the residuals. To increase the confidence in the signal, a low-pass filter is applied to the residuals, which is the running mean. By doing so, abnormalities are filtered out, and the uncertainty in the results is mitigated. The centred rolling mean can be defined by Equation 5.5 [32].

$$e_{\text{roll}}(i) = \frac{1}{2\eta + 1} \sum_{j=0}^{2\eta} e(i + \eta - j) \tag{5.5}$$

In Equation 5.5 $e_{\text{roll}}(i)$ is the rolling mean at time i. e is defined in Equation 5.4 as the residual and η is the window size, which defines the number of observations over which the running mean is calculated. A window size of two days is chosen empirically as it gives a good overview of the time series while filtering out the higher frequencies. Furthermore, a minimum amount of periods is put at 72, which equals half a day. This is done to avoid filtered data from affecting the running mean.

Train Set

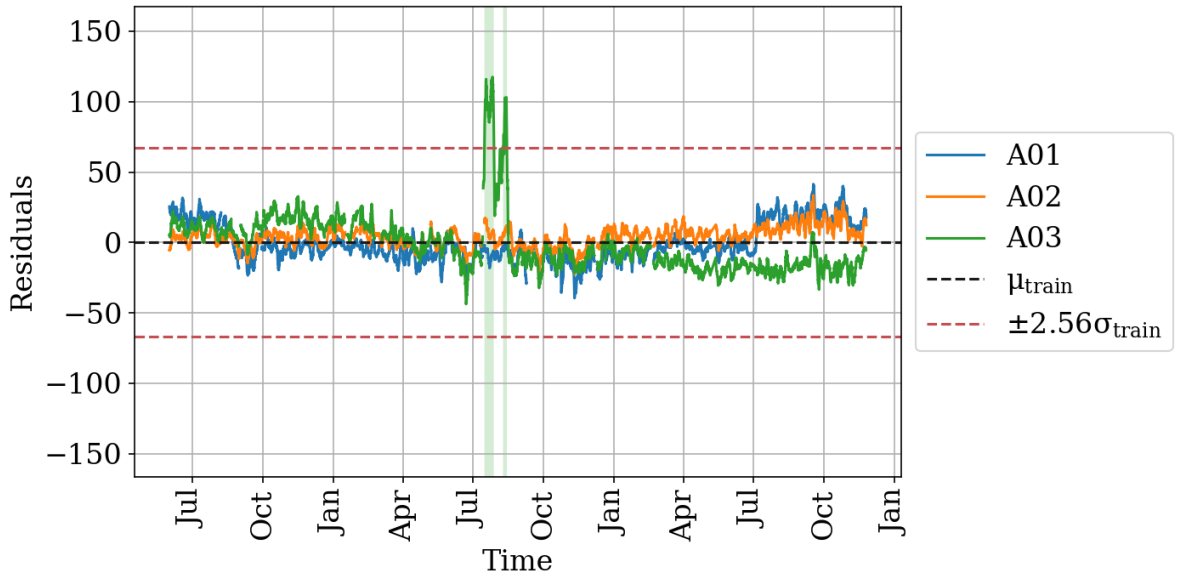


Figure 5.11: Residual time series for the training set

In Figure 5.11 the time series of the residuals in the training set are plotted. The residuals appear to behave normally for each of the three turbines, which means a fluctuating moving average around zero, up until July of the second year. This is when there is a gap in the moving average of turbine A03, after which two positive peaks are visible. Going back to Section 5.5.1, this indicates that the model has a lower prediction than the actual power output.

Test Set

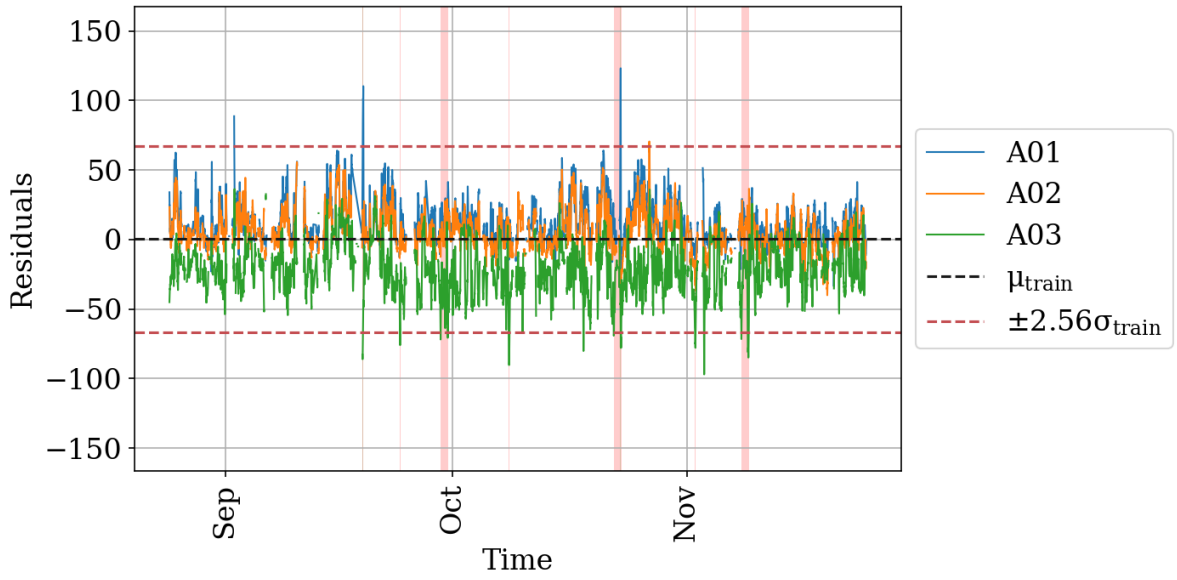


Figure 5.12: Residual time series for the test set

Figure 5.12 shows the residual time series for the test data. It should be noted that the window of the rolling mean has been changed to two hours for this visualisation. This is done because the test set time span is significantly shorter than that of the train set. Furthermore, the amount of minimum of periods has been changed to one hour. From the figure, three cases of underperformance can be seen around the end of September, end of October, and the beginning of November. To investigate these instances, a look into the maintenance reports at hand is taken. From these reports, the following points were taken:

- **September:** an operating efficiency lower than the threshold defined by the operators is noted. Although turbine A03 had more operating hours than the other two turbines, a lower energy output was measured.
- **October:** again a lower operating efficiency was measured, in a similar range for September.
- **November:** the operators diagnosed a fault code in the turbine, representing a mechanical brake problem. The mechanical brake was turned on while the wind speed was in a limit leading to a loss in power output.

5.6. Shapley Analysis

As explained in Section 4.6, shapley values are evaluated to gain a deeper understanding of the working of the model. These values provide a structured framework for attributing the distinct contributions of individual features to the predictions. In this section, the shapley values for the entire data set are evaluated. Afterward, a comparison is made with the shapely values of the overperformance case in A03. The aim is to identify whether the contributions of the features differ in the case of overperformance.

To evaluate the contributions of the feature contribution along the entire data set, a random sample of 1000 datapoints from the training set is taken. The shapley values are then calculated by the *KernelExplainer* in the *Shap* python package. The choice of the KernelExplainer is based on the fact that it does not make any assumptions about the model itself. The contribution of features to a single estimate can be visualised as a so-called waterfall plot, as can be seen in Figure 5.13.

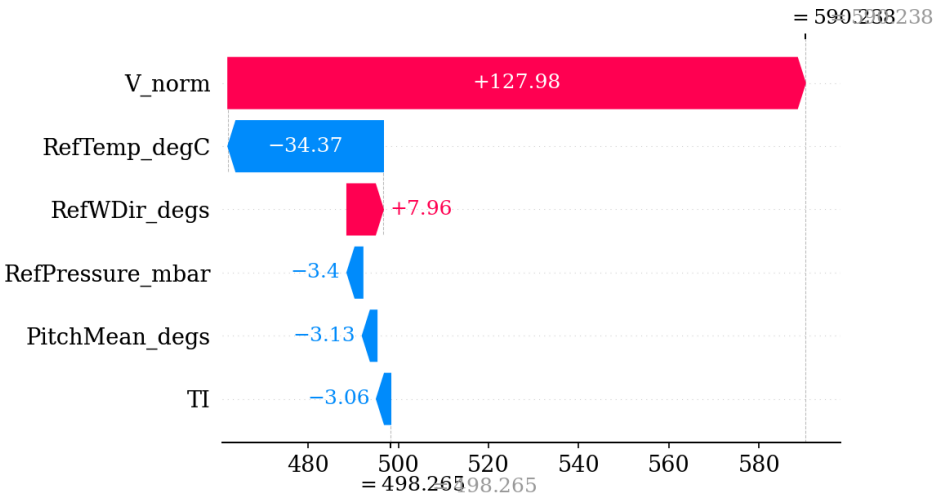


Figure 5.13: Summary plot of the shapley values

Figure 5.13 shows the local accuracy of the SHAP value. By taking the sum of each value contribution, the predicted value is reached. The features are ordered in the plot based on their respective impact on the power output. The negative impact of both the reference temperature and the reference pressure indicates that the density at this particular instance must have been lower compared to the average across the dataset. Second, the positive impact of the wind direction could be an indication that in this particular direction the turbine is exposed to little or almost no wake effects from neighbouring turbines.

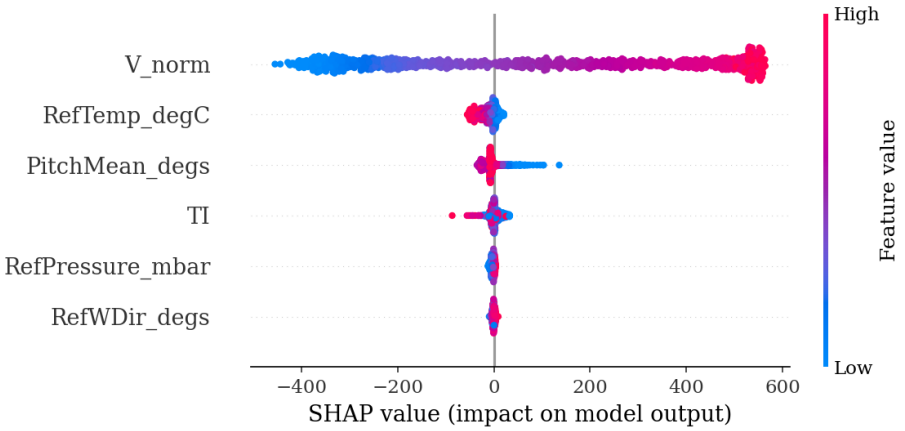


Figure 5.14: Summary plot of the shapley values

Figure 5.14 shows the summary plot of the cluster A01, A02, and A03. As expected, it can be seen that the wind speed has the highest contribution to the prediction of the target values. The pink colour in the plot represents that a higher wind speed leads to a higher prediction of the power output. The blue colour means a lower wind speed, which has a negative impact on the model output. Looking at the influence of temperature, it can be seen that a higher temperature leads to a lower prediction of power output. Going back to Equation 2.1, where temperature directly influences air density, it means that a higher temperature leads to a lower air density. In combination with Equation 2.4, it can be concluded that the model captures the power equation. If only the density is varying, a higher density leads to a higher power output, and a lower density leads to a lower power output. Third, the pitch control system is captured in the SHAP values. For high pitch angles, there is a negative effect on the SHAP value, which means that there is a lower power output. Interestingly enough, the effect of lower pitch angles seems to have a varying effect on the SHAP value. Fourth, the turbulence intensity shows that a high turbulence intensity leads to lower power output. Additionally, the model does not seem to capture that a lower turbulence intensity would lead to higher power output. Finally, both the pressure and wind direction are clustered around the zero SHAP value line, meaning that they have less influence on the model output. From this evaluation, it can be concluded that a new model can be trained where these features are not taken into account.

5.7. Root-cause Analysis

The case of A03 is of special interest, as it showed overperformance in the training set. As a first step, maintenance reports are examined around the time of so-called overperformance. Two maintenance steps have been identified within this time frame:

- **Beginning of July:** maintenance of a lubricating oil pump occured in the beginning of July which lasted for approximately 40 hours. This explains the gap in the time series of the residuals. Furthermore, it could also be an explanation for the negative slope that the residuals had before this event.
- **End of July:** The anemometer and the turbine windvane have been replaced. Due to a malfunction of the anemometer, it was found that there was an offset and incorrect slope for the controller.

To see the effect of anemometer malfunction, the power curve for the month of July is constructed in Figure 5.15. From the curve it becomes clear that the actual power output measurements have been shifted towards the right. In essence, this means that the SCADA system registered a lower wind speed than the actual wind speed, which causes the change.

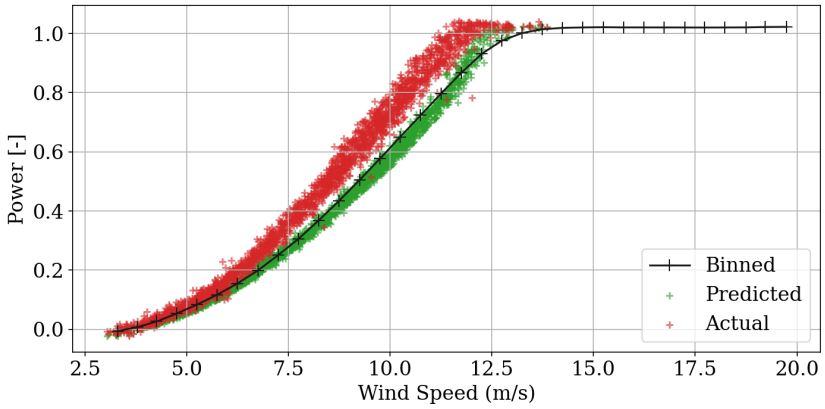


Figure 5.15: Power curve of overperformance A03

To further investigate whether the underperformance in A03 is an influence of one of the features, a waterfall and summary plot have been constructed for this time interval. First, a random sample point within this time frame is taken to see the contribution of the features to this specific prediction. Figure 5.16 shows the contributions of the features, and interestingly enough, the order of influence on this instance has changed. It is noteworthy that the pitch has a higher impact on the model output, which is also indicated in Figure 5.17.

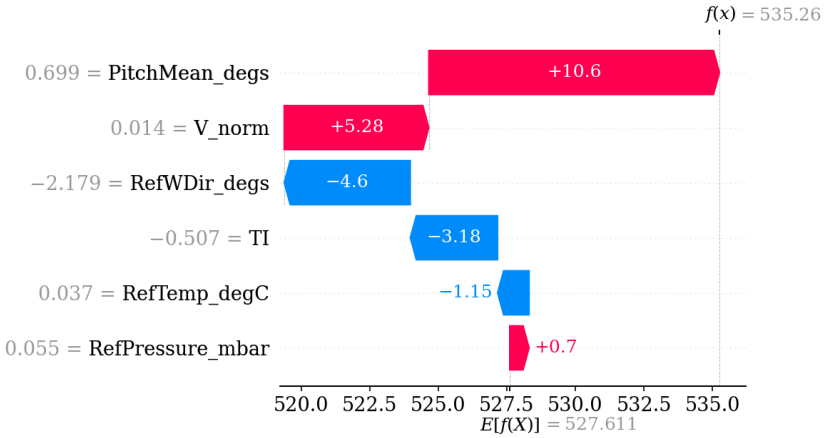


Figure 5.16: Waterfall plot for a specific instance of the data from A03

In Figure 5.14 it is found that the wind speed has the greatest impact on the model output. The influence of the wind speed has not changed in the case of turbine A03. It is merely the other features that have changed in impact order, as can be seen in Figure 5.17. As described earlier, the pitch angle exhibits to have a larger impact than for the full dataset. An explanation of this could be the effect of the wrong wind speed measurements, which means that the pitch controller lags in adjusting the angle accordingly. Since the turbine is constant speed controlled, it is expected that the

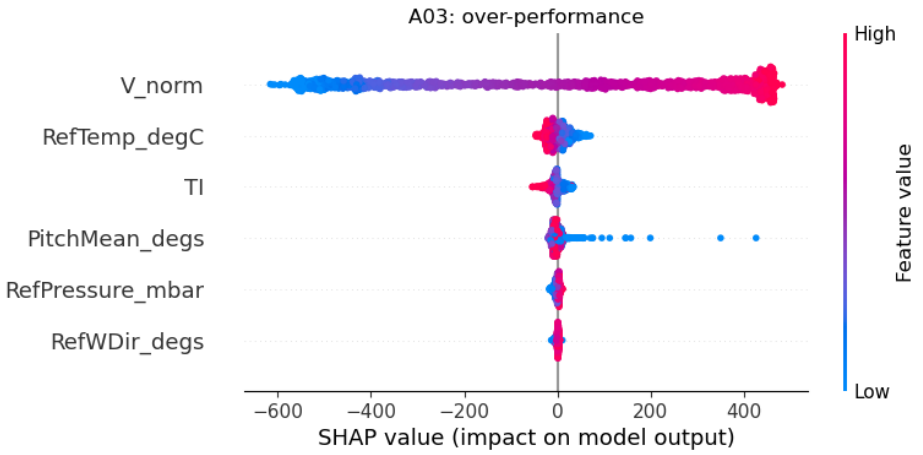


Figure 5.17: Summary plot of the shapley values for the overperformance case in A03

5.8. Conclusion

In conclusion, this chapter explores a cluster-based power performance analysis. The cluster used in this chapter is that of turbines A01, A02, A03. First, the extraction of features provided an understanding of the parameters that influence the target column, which were selected through the spearman correlation matrix. Second, the hyperparameter optimisation process not only fine-tuned the network, but also highlighted its sensitivity. Third, the training performance described the process and the impact on the power curve and the power coefficient. These analyses revealed the strengths and limitations of our trained model, establishing a foundation for understanding its predictions and identifying underperformance or overperformance. Furthermore, the study of residuals explored the uncertainties and biases inherent in the model's predictions. Distribution analysis and binned residual plots revealed how model performance varies with different wind speeds. Subsequently, the time series of residuals helped us uncover anomalies and potential issues in the turbines' operation, showing the real-world impact of the model's predictions. Additionally, the Shapley analysis provided a deeper understanding of the contribution of features, allowing us to quantify and visualise their impact on the predictions. This not only validated our model's behaviour, but also helped to identify critical features affecting power output.

6

Wind Farm Analysis

As described in Section 4.1, the wind turbines are divided into clusters. In this chapter, the wind farm clusters are described in Section 6.1. Subsequently, the residual statistics of the wind farm are presented in Section 6.2. In Section 6.4, six cases in the wind farm is discussed.

6.1. Wind Farm Clustering

In this section, the clusters in wind farm A are described. The methodology to find the clusters can be found in Section 4.1. The goal of this step is to create fleets within the wind farm that are expected to have a similar power production based solely on their nacelle wind speed measurements. Looking at the wind farm layout shown in Figure 3.1, intuitively the fleets could be selected as rows within the farm. However, the use of Kmeans is opted for to avoid using wind farm expertise. Based on the elbow curve in Figure 6.1, the optimal number of clusters is found to be 15 for farm A. The visual representation of the clusters can be found in Figure 6.2.

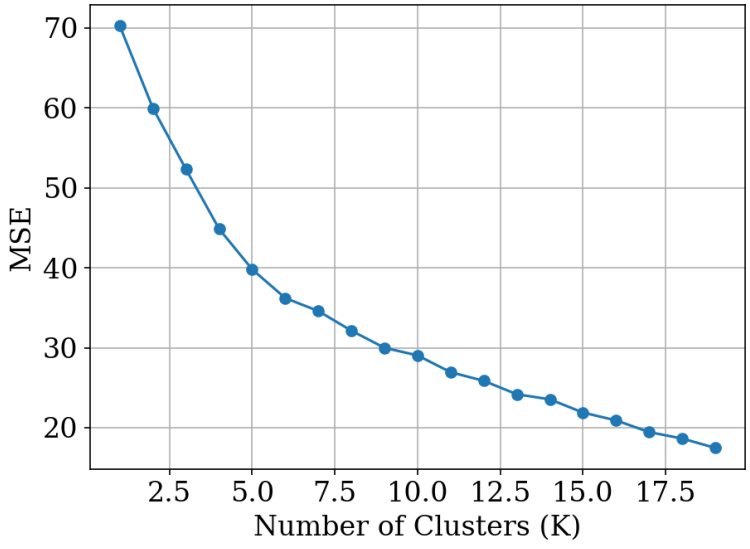


Figure 6.1: Elbow plot for wind farm A

Table 6.1: Clusters for wind farm A

No.	Turbines
0	A58-A61
1	A23-A26
2	A01-A03
3	A42-A47
4	A04
5	A05-A08, A10
6	A18-A22
7	A27-A32
8	A57
9	A34-A35
10	A09, A11, A40-A41
11	A48-A56
12	A36
13	A37-A39

From Table 6.1, it can be seen that three wind turbines have been put into single turbine clusters. The reason for this is not entirely certain, but their location could be the main reason. To visualise the location of the clusters on the orographic graph, Figure 6.2 has been constructed. To do so, the Python package *bokeh* has been used. It is important to note that on a wind farm A, the wind turbines A12-A17 and A33 have a different SCADA structure and therefore different channels and are plotted in the same colour.

Looking at turbine A04 one can see that it is situated close to the cluster of A01-A03. Unlike turbine A03, it has a plateau located in the south and a mountain ridge in the north. This terrain difference makes it more difficult to compare the performance of these two turbines. A similar terrain difference can be found for turbine A57; although only two rotor diameters divided from turbine A58, it is much closer to the ridge. For A36, the reason remains uncertain; no errors were recorded in the wind speed measurements, nor are the terrain conditions different from those of A37. A deeper analysis of the wind speed measurements would have to be taken compared to neighbouring turbines.

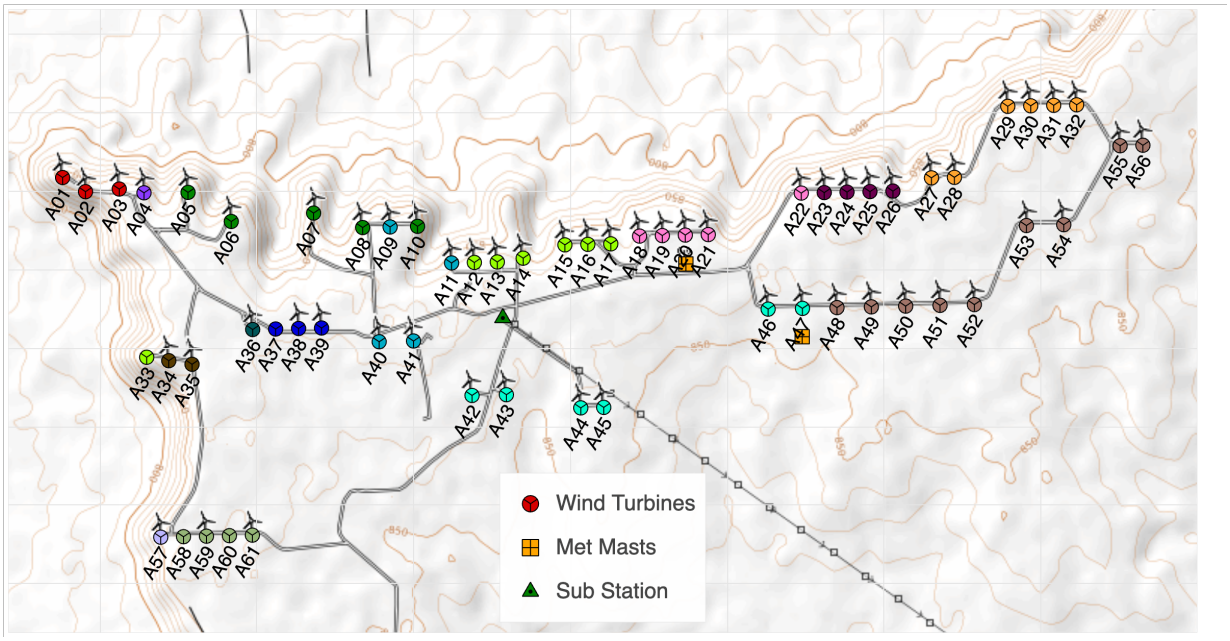


Figure 6.2: Cluster representation of wind farm A on an orographic chart

6.2. Residual Analysis

Similar to the approach taken in the residual analysis of a single cluster, see Section 5.5, the distribution of the residuals has been calculated across the wind farm. First, the bias in the residuals is explained by the residual mean. Second, the variance is explained by the standard deviation.

6.2.1. Mean

In terms of residual analysis, the mean gives us an indication of the bias present in the training and the test set. A positive bias is an indication that the particular wind turbine is consistently overperforming compared to the other turbines in its cluster. On the contrary, a negative bias is an indication that the turbine is underperforming. It should be noted that other factors also come into play when looking at the bias, such as wake from other turbines, wind shear due to environmental conditions, influences due to maintenance, and more. As explained in Section 5.5, this can be seen, for example, in the case of turbine A03. In Figure 6.3 the mean residuals for the wind farm are presented for the test and training set.

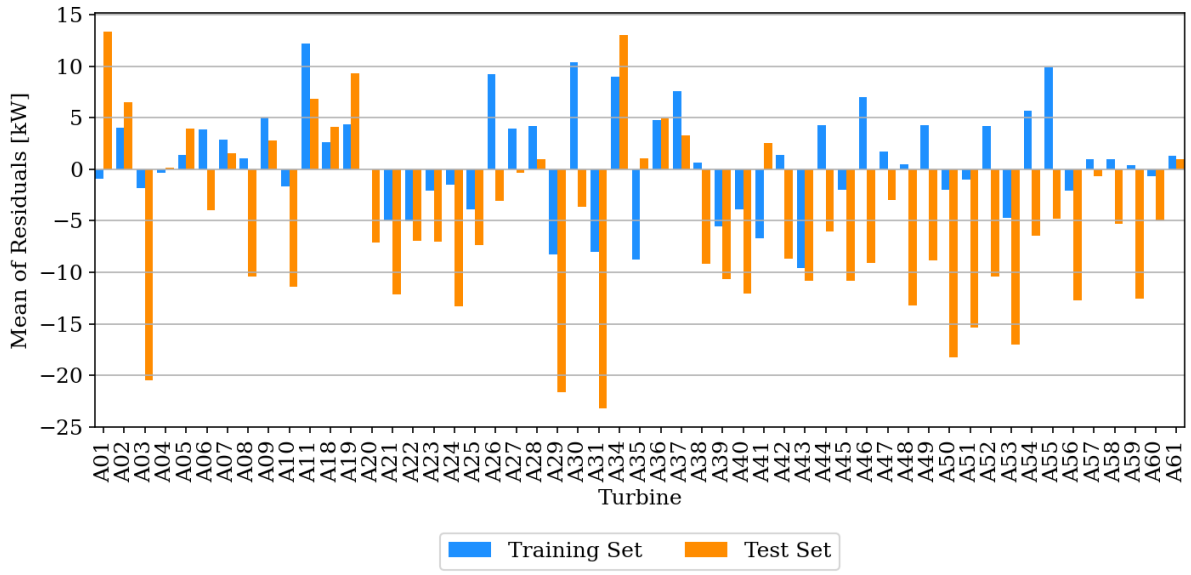


Figure 6.3: Mean residuals for wind farm A

It can be seen that three turbines have a bias larger than 20 kW for the test set, which are A03, A29, and A31. Note that A29 and A31 are part of cluster 8 and the other turbines do not exhibit the same bias. This case will be examined further in Section 6.4.

6.2.2. Standard Deviation

The standard deviation describes the variance in the residuals of the data sets. In Figure 6.4 the variance of the turbines is visualised. The median of variance is found to be approximately 23.5 kW for the train set and 22.9 kW for the test set. The variance can be explained as the model trying to train from a model with a higher level of variance, and therefore has a higher level of variance in the residuals as well.

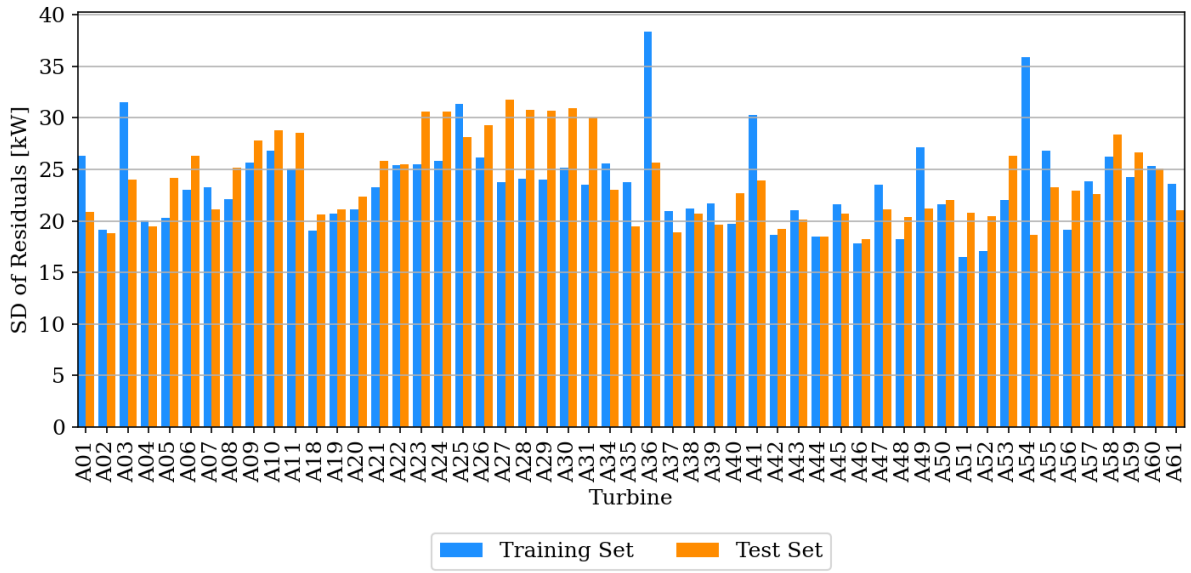


Figure 6.4: Standard deviation of residuals for wind farm A

From Figure 6.4, the main trend is that the residual variance of the training set is higher than that of the test set. This pattern is broken by the turbines in cluster 7, namely A27 up to A32, in which the standard deviation of the test set is found to be higher. Furthermore, three cases draw a special attention, those of turbines A03, A36, and A54. Here, the variance in the training set is found to be significantly higher than that of the test set. The reason for this will be examined in Section 6.4.

6.3. AEP

In this section, the AEP of the wind farm is explored, which serves as an indication of how the wind turbines perform. In Figure 6.5 the AEP of each turbine is visualised in relation to the other turbines in its cluster. As discussed in Section 3.3.1, it shows that turbines located near ridges outperform those located in the centre of the plateau. This becomes evident by comparing the best performing, namely, cluster 2 and the worst performing cluster 3. A comparison between the predicted and actual AEP will be made in Section 6.4.

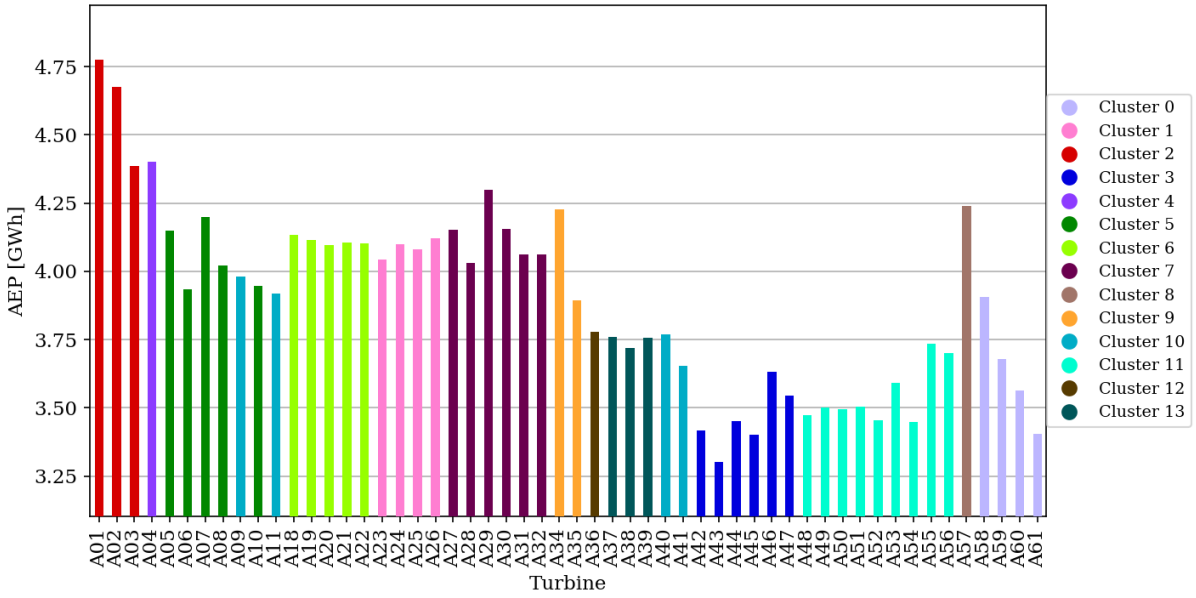


Figure 6.5: AEP calculation across wind farm A

6.4. Case Studies

In this section, the case studies of the clusters are presented. First, the location of the cluster is highlighted. Then the performance metrics of the test and training are shown. Subsequently, on the basis of the time series and the percentage difference of the AEP, a root cause analysis for deviating turbines is performed. Clusters 4, 9, and 13 have not been included in the case study since a single wind turbine forms the cluster, their time series, and AEP calculations can be found in Appendix B.

6.4.1. Cluster 0

As a recap this cluster consists of turbine A58 up to A61 and is located south-east of the wind farm. The performance metrics for this cluster are shown in Table 6.2. The metrics show that the largest outliers are present in the residuals of turbine A58 for training and in A59 for the test set.

Table 6.2: Performance metrics of cluster 0

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A58	26.28	17.00	0.993
	A59	24.27	15.62	0.994
	A60	25.35	17.41	0.993
	A61	23.63	15.96	0.993
Test	A58	28.90	19.80	0.992
	A59	29.45	21.59	0.991
	A60	25.59	18.51	0.993
	A61	21.10	15.65	0.995

To visualise the behavior of the residuals the time series for the test and train set are constructed respectively, in Figure 6.8 and Figure 6.6. The centred moving average followed the same inputs as discussed in Section 5.5.3.

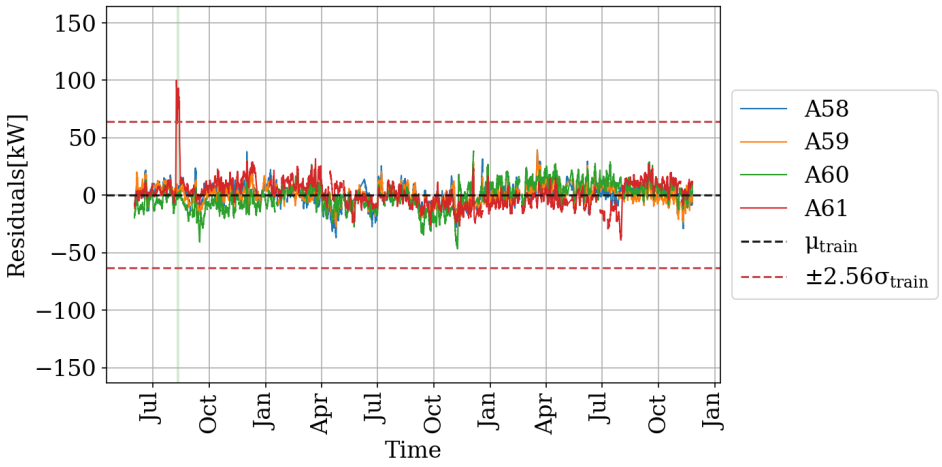
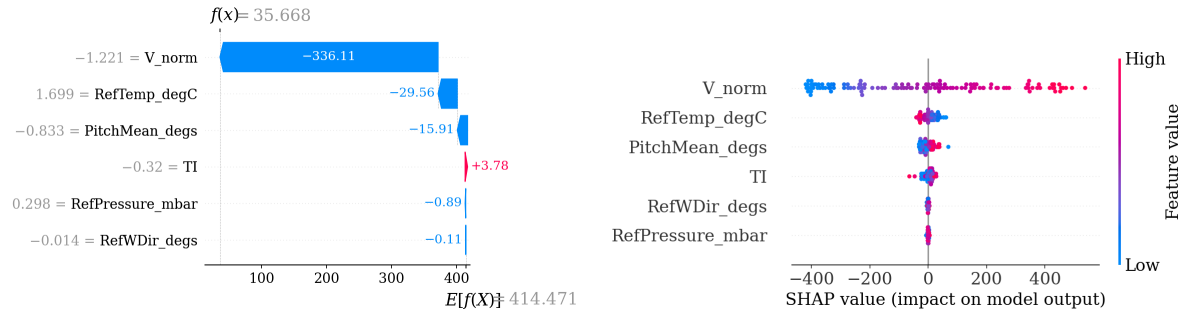


Figure 6.6: Residual time series for the training set of cluster 0

In Figure 6.6 a case of overperformance is detected for turbine A61 in August of the first year. The reason behind this overshoot is unclear, since no reports were provided for this month. A deeper understanding in the influence of the features could provide answers to this; therefore, the SHAP values are calculated in this time frame, which is shown in Figure 6.7.



(a) Waterfall plot A61 **(b)** Summary plot A61

Figure 6.7: SHAP analysis of overperformance turbine A61

From Figure 6.7 it can be seen that the contribution of the temperature affects the prediction of the power output. Furthermore, the contribution of the pitch angle differs from that presented for the entire data set, as seen in Figure 5.14. Here, relatively low pitch angles have a negative impact on the prediction. When looking at the event description of the SCADA data set, it was found that the turbine had been struck by lightning the day before the overshoot. The lightning may have affected the sensors present in the turbine.

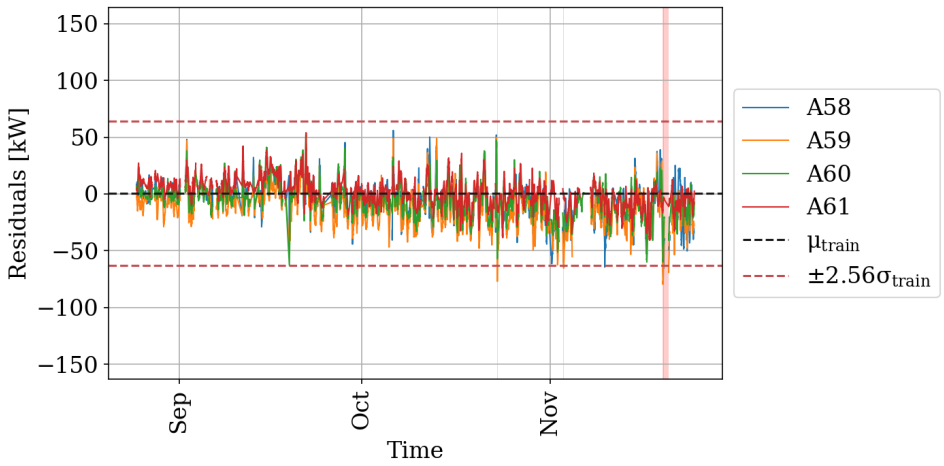


Figure 6.8: Residual time series for the test set of cluster 0

In Figure 6.8 a case of underperformance is detected for turbine A59. The reason for the detected underperformance of turbine A59 in November was found in maintenance reports. During this month, the turbine suffered from yaw break pressure faults, resulting in a larger yaw misalignment and, therefore, lower power output. Furthermore, at the beginning of the month a circuit overload was recorded, resulting in downtime in the second week of the month, which also explains the gap in the time series. To know what influence this has on AEP, a percentage difference between actual and predicted AEP is calculated, which can be seen in Figure 6.9. From this it becomes clear that the performance of turbine A59 has been affected in this time interval by the detected fault.

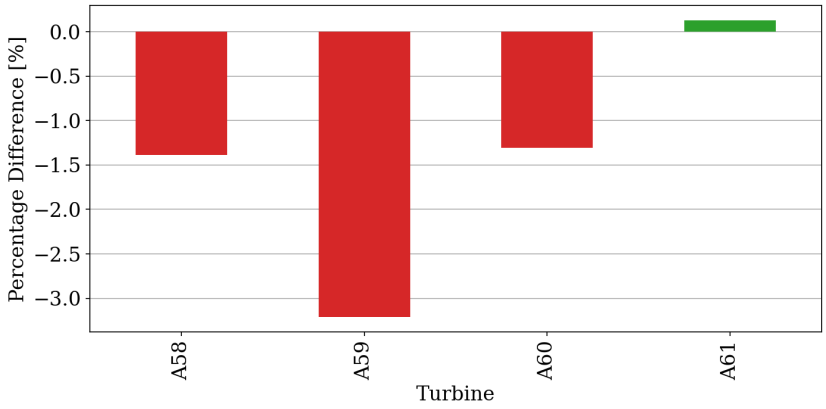


Figure 6.9: Percentage difference in AEP of cluster 0

6.4.2. Cluster 1

As a recap this cluster consists of turbine A23 up to A26 and is located North-West of the wind farm. The performance metrics are listed in Table 6.3. In the training set the largest outliers in the residuals are found to be in turbine A25 and in A24 for the test set. The performance is shown to be the best for turbine A26.

Table 6.3: Performance metrics of cluster 1

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A23	25.56	17.74	0.994
	A24	25.84	17.28	0.994
	A25	31.62	20.59	0.991
	A26	27.75	18.96	0.993
Test	A23	31.39	23.29	0.991
	A24	33.38	25.57	0.990
	A25	29.05	21.57	0.993
	A26	29.47	21.39	0.993

The time series for the test and train set are visualised, respectively, in Figure 6.11 and Figure 6.10.

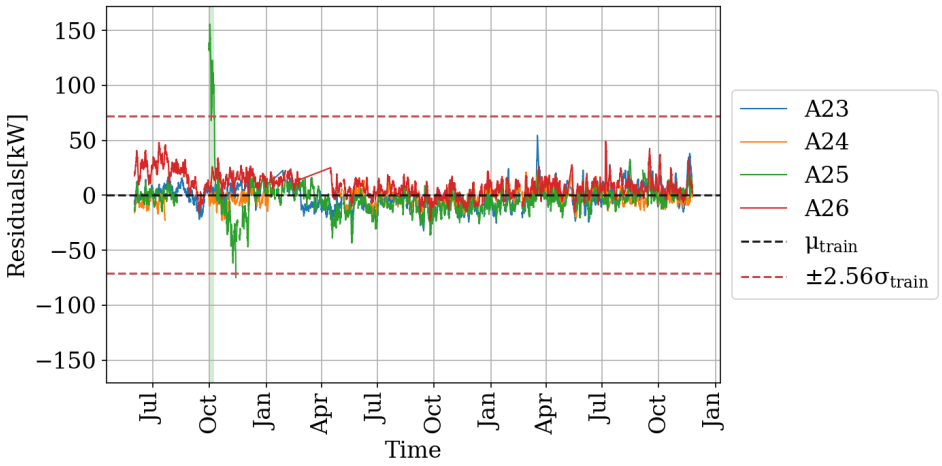


Figure 6.10: Residual time series for the training set of cluster 1

In Figure 6.10 a positive overshoot in the beginning of October is registered for turbine A25. This is considered to be a false alarm, as it can be seen that it is right after a long period of maintenance. Furthermore, after the peak, a negative trend is shown towards the end of the first year. Putting this into perspective with Figure 6.4, this could explain why the training and set variances have a different behaviour than that of the other turbines in the cluster.

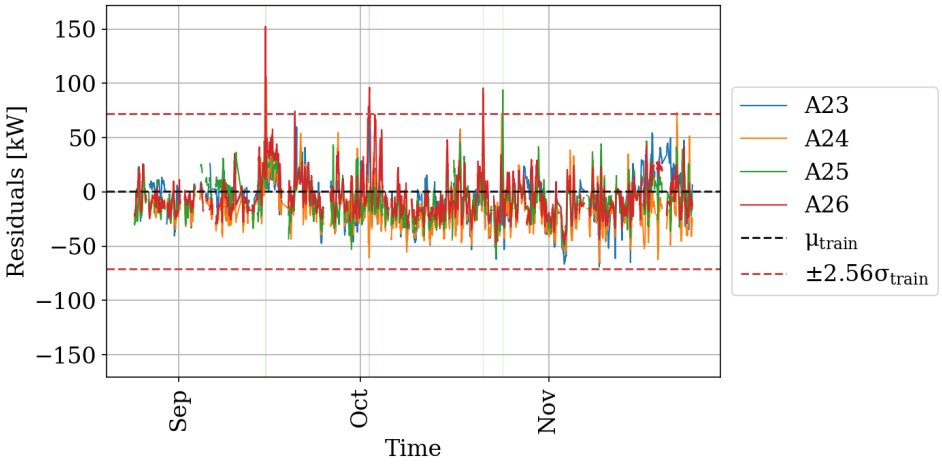


Figure 6.11: Residual time series for the test set of cluster 0

The test set for this particular shows a couple of positive overshoots in particular for turbine A26. To validate whether this is a false alarm, the predicted AEP for this cluster is created which can be seen in Figure 6.12. From the figure it can concluded that although the general trend throughout this period is to be underperforming, it performs much better than the other turbines in the cluster.

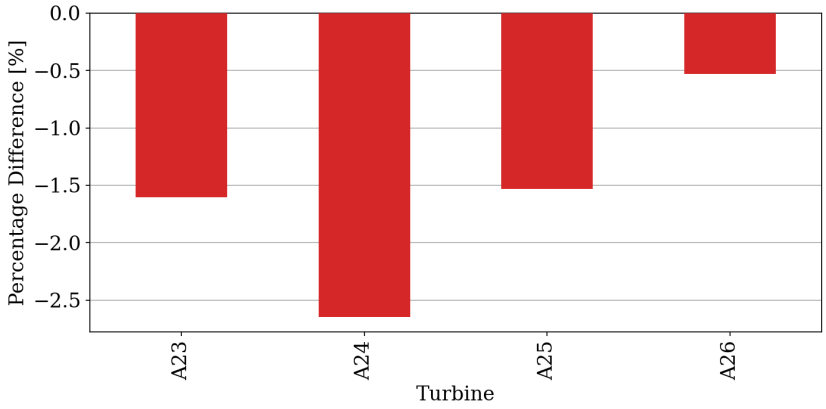


Figure 6.12: Percentage difference in AEP of cluster 1

6.4.3. Cluster 3

As a recap this cluster consists of turbine A42 up to A47 and is located in the centre of the wind farm. Additionally, it is located in the middle of the plateau, which influenced the average power output, as can be seen in Figure 6.5. The performance metrics of this cluster are shown in Table 6.4.

Table 6.4: Performance metrics of cluster 3

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A42	18.67	13.47	0.996
	A43	23.11	16.36	0.993
	A44	18.96	13.91	0.995
	A45	21.70	15.38	0.994
	A46	19.13	14.44	0.996
	A47	23.57	17.33	0.994
Test	A42	21.07	16.20	0.995
	A43	22.87	17.37	0.994
	A44	19.41	14.58	0.995
	A45	23.38	17.59	0.993
	A46	20.36	15.60	0.995
	A47	21.34	16.39	0.995

The time series for the test and train set are visualised, respectively, in Figure 6.14 and Figure 6.13.

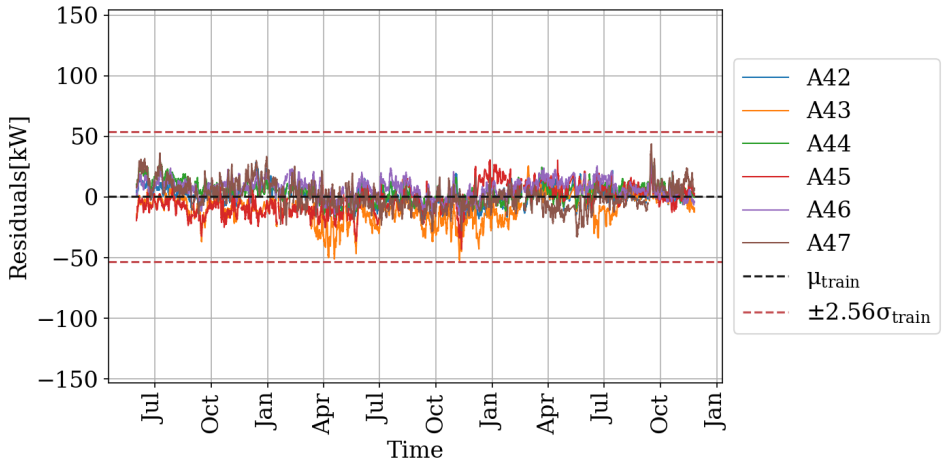


Figure 6.13: Residual time series for the training set of cluster 3

In the training set no abnormal instances have been recorded, as can be seen in Figure 6.13.

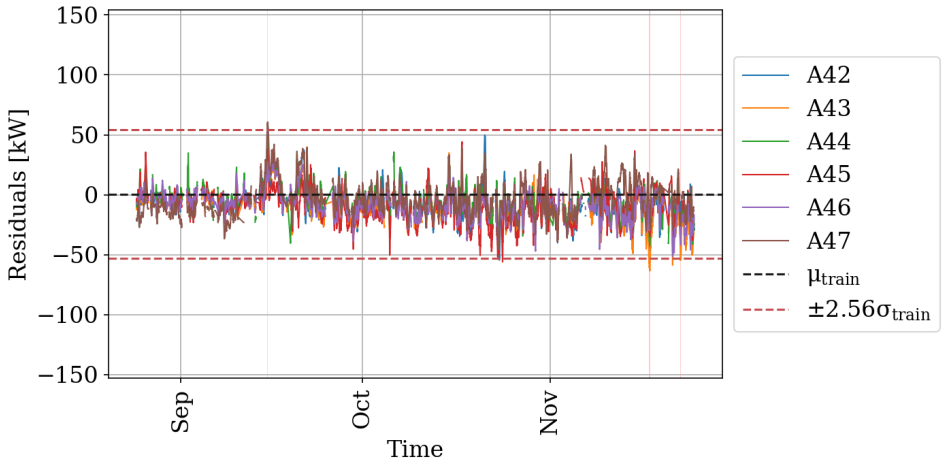


Figure 6.14: Residual time series for the test set of cluster 3

In the residuals of the test set, only two cases have been detected by the algorithm, in November for turbine A43. Looking at the maintenance reports, this is due to pitch faults which have resulted in maintenance at the end of the month. The pitch faults might have been present for a longer period, however, by looking at the difference of the actual AEP and the predicted AEP in Figure 6.15. A similar pitch fault was detected for turbine A42.

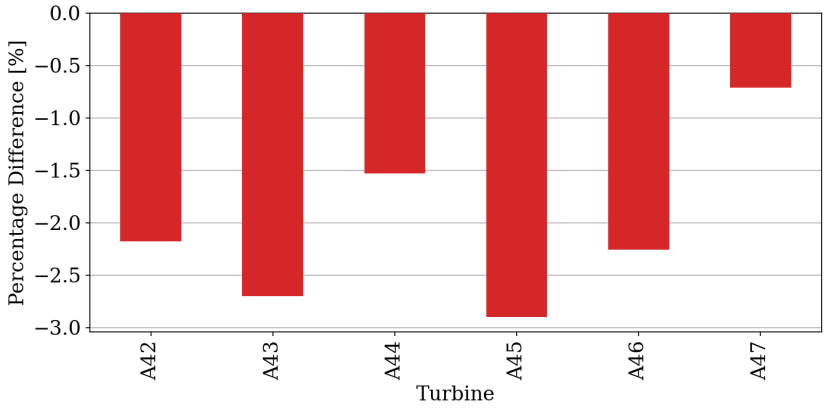


Figure 6.15: Percentage difference in AEP of cluster 3

6.4.4. Cluster 5

Cluster 5 consists of turbine A05 up to A08, and A10 and is located on the north side of the wind farm. The five turbines are located at the corner of mountain ridges and have hill flanks on various sides. The performance metrics for this cluster are shown in Table 6.5.

Table 6.5: Performance metrics of cluster 5

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A05	20.36	14.65	0.996
	A06	23.34	15.44	0.995
	A07	23.42	15.94	0.995
	A08	22.10	14.89	0.995
	A10	26.86	16.76	0.993
Test	A05	24.50	18.40	0.995
	A06	26.60	19.57	0.993
	A07	21.15	14.53	0.996
	A08	27.24	20.45	0.993
	A10	31.00	22.45	0.991

From Table 6.5 it can be seen that for the training set the accuracy is at a high level compared to the previous clusters. For the test set, the largest deviations are present in turbines A08 and A10. The time series for the test and train set are visualised, respectively, in Figure 6.17 and Figure 6.16.

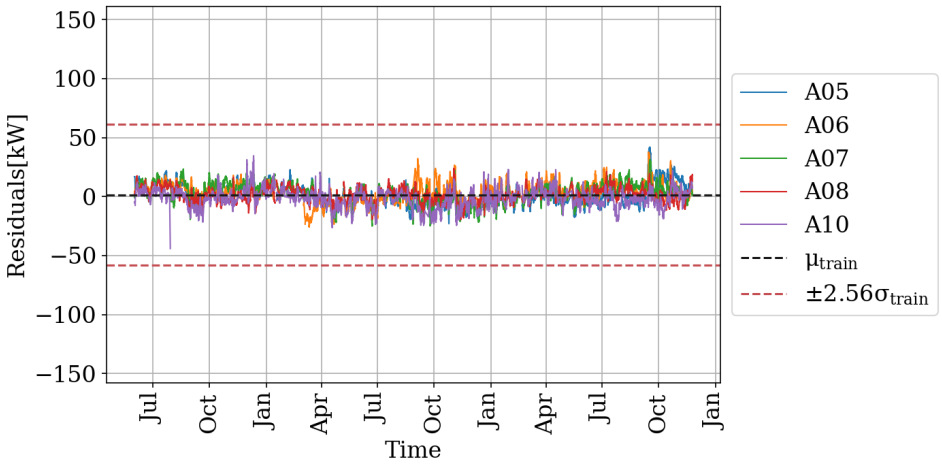


Figure 6.16: Residual time series for the training set of cluster 5

In the training set no abnormal instances have been recorded, as can be seen in Figure 6.16, which underlines that the model learns sufficiently from the data.

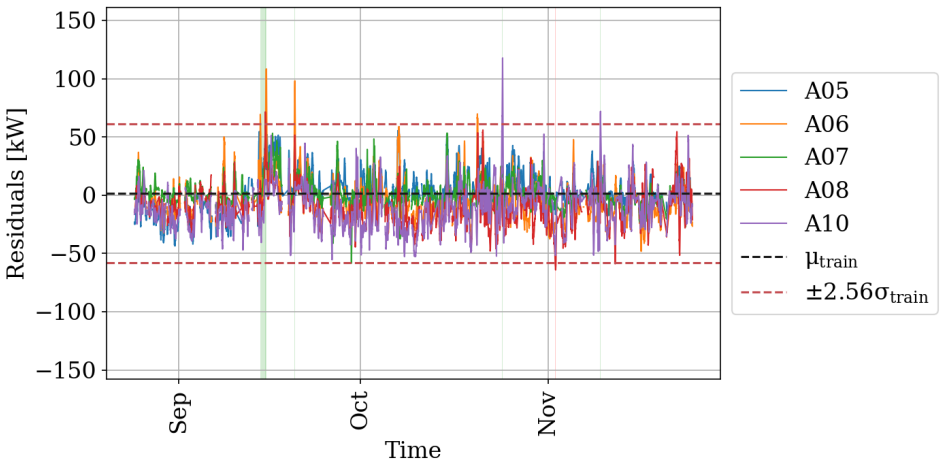


Figure 6.17: Residual time series for the test set of cluster 5

In Figure 6.17, the time series for cluster 5 are presented. The plot provides several detections of over- and underperformance. For turbine A06 two instances of overperformance were detected in September. In the maintenance reports, a similar trend is found for this month, where an operating efficiency of more than 100% was recorded. Looking at turbine A08, it can be seen that an instance of underperformance is detected in the beginning of November. In Figure 6.3, the residual mean was found to be around -10.5 kW indicating an underperformance throughout the period. In maintenance reports, it is found that the anemometer of A08 recorded erroneous measurements and was replaced in December, thus out of the investigated window. This could be an indication that the model captured faults in the control system before the anemometer was replaced.

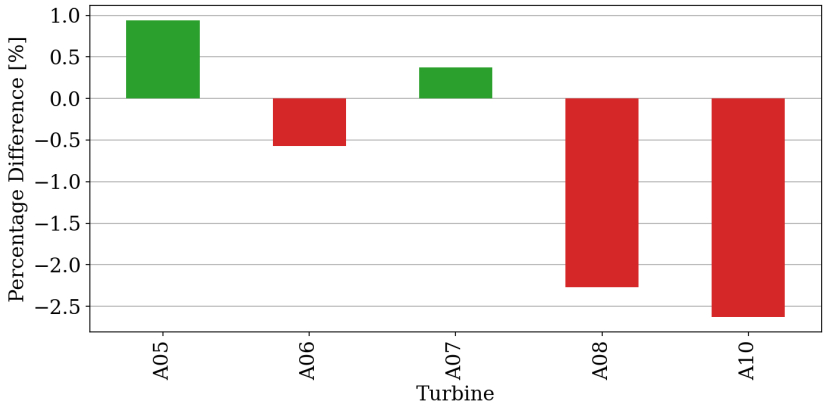


Figure 6.18: Percentage difference in AEP of cluster 5

In Figure 6.18 the underperformance in A08 is again verified by the percentage difference. Furthermore, it shows that underperformance occurs in turbine A10, although this was not detected in Figure 6.17. The underlying reason is maintenance on the pitch control system, which took place in October. In the beginning of this month, illegal pitch span data were recorded.

6.4.5. Cluster 6

Cluster 6 consists of the turbines A18 to A22 and is located in the centre north side of the wind farm. The performance metrics for this cluster are shown in Table 6.6.

Table 6.6: Performance metrics of cluster 6

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A18	19.26	13.20	0.997
	A19	21.13	14.81	0.996
	A20	21.14	14.70	0.996
	A21	23.79	17.07	0.995
	A22	25.90	17.68	0.994
Test	A18	21.06	15.05	0.996
	A19	23.13	16.53	0.995
	A20	23.45	17.80	0.995
	A21	28.52	21.04	0.993
	A22	26.42	19.04	0.994

Figure 6.19 shows the time series for the training set. It can be seen that no anomalies have been detected. This is underlined by the RMSE and MAE, as listed in Table 6.6. Furthermore, it is an indication that the model learns sufficiently from the training set to describe the behaviour of all the turbines in this cluster.

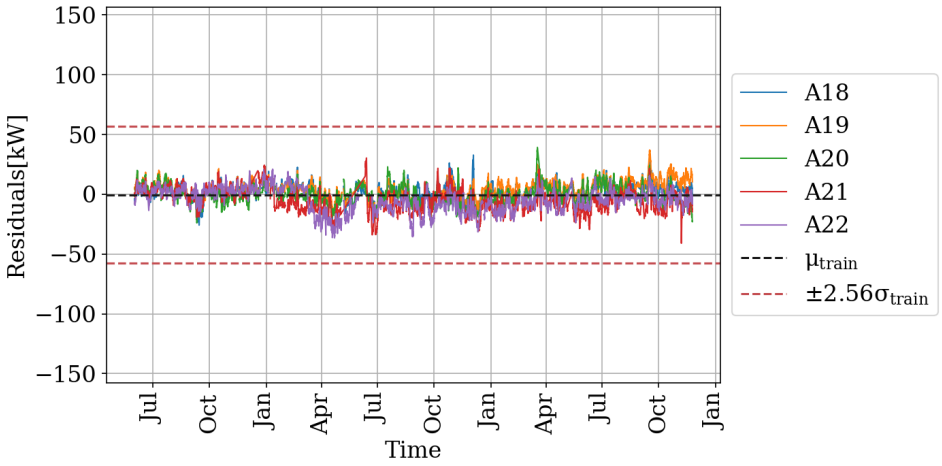


Figure 6.19: Residual time series for the training set of cluster 6

Looking at the performance metrics for the test set, it can be seen that more outliers are present in the data, which is highlighted by the higher RMSE. In Figure 6.20 this hypothesis is visualised. The model detected three instances of overperformance in turbine A19. The higher performance in A19 can also be seen in the percentage difference of AEP, which is shown in Figure 6.21. Furthermore, overperformance was confirmed by maintenance reports in which a higher operating efficiency was recorded for each of the three months. Two cases of underperformance are detected, each on a different turbine. First, at the end of October, it was detected in turbine A21. No special events were recorded in the reports, but by looking at the time series of the other turbines, the negative slope seemed to be a general effect where A21 performed the worst in comparison with the rest of the cluster. The second case is found in A20. In November, it was recorded that the turbine has a persistent error with the recorded anemometer signal during high wind speeds.

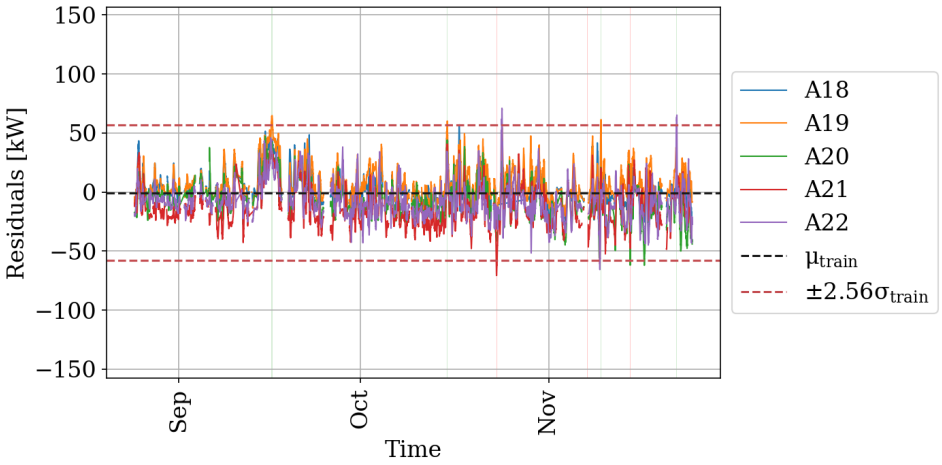


Figure 6.20: Residual time series for the test set of cluster 6

In Figure 6.21, it can be seen that the energy production of A20 is indeed affected by erroneous measurements of the wind speed. Furthermore, for turbines A21 and A22 a lower AEP is also predicted. This was found to be in line with the lower operating efficiency recorded in these months.

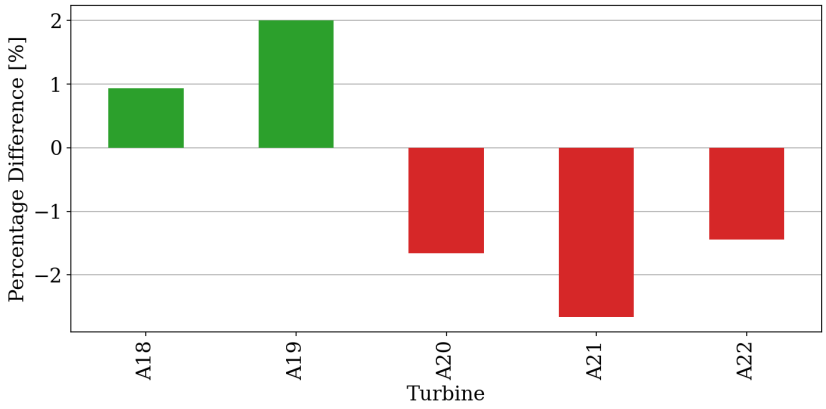


Figure 6.21: Percentage difference in AEP of cluster 6

6.4.6. Cluster 7

Cluster 7 consists of the turbines A27 to A31 and is located north-west of the wind farm. This particular cluster was mentioned in Section 6.2.2 because the standard deviation of the residuals in the test set was much higher than that of the training set. This can also be seen in the performance metrics, shown in Table 6.7.

Table 6.7: Performance metrics of cluster 7

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A27	24.09	16.29	0.995
	A28	24.44	16.32	0.995
	A29	25.43	18.06	0.995
	A30	27.22	18.48	0.994
	A31	24.82	16.92	0.995
Test	A27	31.76	24.05	0.991
	A28	30.79	22.75	0.991
	A29	37.53	29.11	0.989
	A30	31.13	23.11	0.991
	A31	37.92	30.54	0.988

Figure 6.22 shows the time series for the training set, where no anomalies have been detected. This is underlined by the RMSE and MAE, as listed in Table 6.7. Furthermore, it is an indication that the model learns sufficiently from the training set to describe the behaviour of all turbines in this cluster.

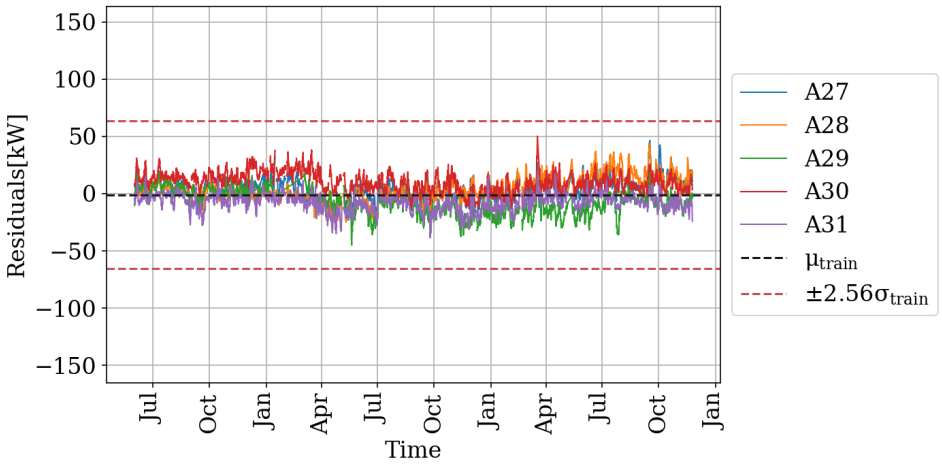


Figure 6.22: Residual time series for the training set of cluster 7

Figure 6.23 shows the residual time series, and the first thing that can be noticed is the amount of anomalies detected. In green the detected overperformance cases are highlighted, of which three were detected for A27 and three for A28. These cases of overperformance were also recorded in maintenance reports in which operating efficiencies of up to 102 % were measured. Second, it can be seen that there are several cases of underperformance in A29. In December, the turbine received a new offline filter for the gearbox. Its function is to filter and circulate lubricating oil within the gearbox to ensure the smooth operation of the gears and bearings. It could be an indication that the efficiency of the gearbox, before replacement, was already decreasing. No major fault logs were recorded for turbine A31. However, the reports did verify a lower operating efficiency.

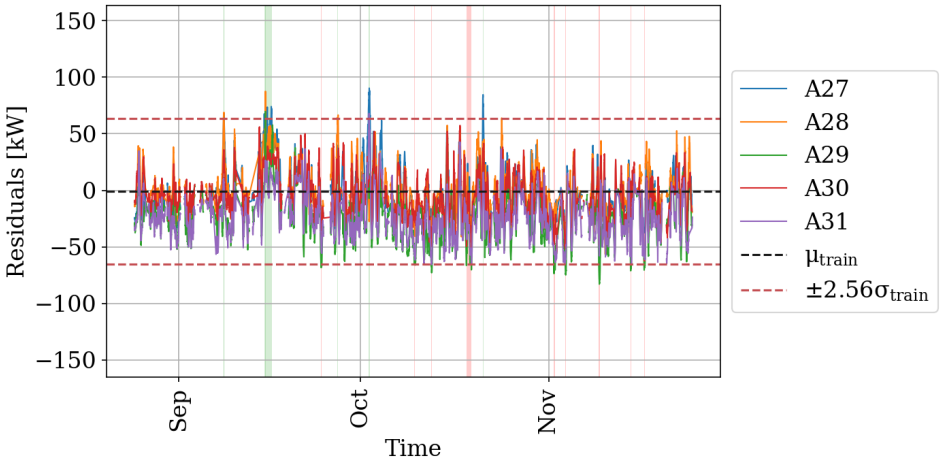


Figure 6.23: Residual time series for the test set of cluster 7

In Figure 6.24 it can be seen that the AEP prediction is heavily affected by the decrease in gearbox efficiency. Furthermore, the lower operating efficiency in turbine A31 leads to underperformance in the test set.

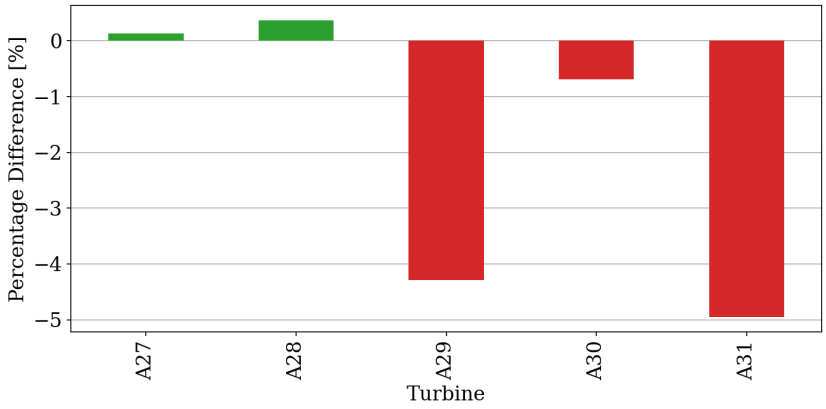


Figure 6.24: Percentage difference in AEP of cluster 7

6.5. Conclusion

In this chapter, the wind farm clusters were investigated, based on the clustering methodology discussed in Section 4.1. The main focus of this chapter was to analyse the performance and behaviour of wind turbines within clusters. The clusters were determined using K-means clustering, which led to the identification of different groups of turbines within the wind farm. Section 6.2 delved into the residual analysis of the wind farm clusters. This analysis provided insights into the performance of individual turbines within their respective clusters. The mean and standard deviation of the residuals were calculated and visualised, allowing the identification of turbines with consistent over- or underperformance, as well as variations in performance within clusters. In Section 6.3, the AEP of the wind farm was explored, serving as an indicator of turbine performance. The AEP calculations were related to turbine clusters, highlighting how the geographical location of turbines can affect their power output. This information allowed for comparisons between turbines located near ridges versus those located in other areas of the wind farm. Finally, in Section 6.4, case studies were presented for selected clusters, showcasing the analysis of individual turbines' behaviour and performance. For each cluster, the location, performance metrics, time series of residuals, and AEP comparisons were discussed. Notable cases of overperformance and underperformance were explained on the basis of maintenance records and specific events affecting the turbines.

7

Conclusion

This thesis aimed to provide a methodology for detecting anomalies in power generation within a wind farm. A literature study was conducted to evaluate the algorithms currently used, and based on the literature, a methodology was proposed. Due to the complex terrain characteristic of this particular site, a different approach had to be taken compared to the methods described in the literature. The first step was to cluster the turbines into different groups based on the wind speed measurements. Wind speed was chosen because it is the main driver for the power output. The step showed to provide rather intuitive clusters within the farm.

Subsequently, the raw data sets were filtered based on the primary filters in the data, removing obvious anomalies from the power curve. In addition to this, a local outlier factor algorithm was applied to remove the remaining anomalies, the contamination factor for this algorithm was set at 0.5 %. With the filtered data sets, the features needed for training were selected using the Spearman correlation ranks. A multivariate approach was chosen on the basis of the higher accuracy of power prediction for these methods. The following six features were selected; wind speed, turbulence intensity, pitch angle, wind direction, temperature, and air pressure. The channels for these features formed the basis for the training set needed to predict the power output.

To predict the power output, a feed-forward multilayer perceptron neural network was implemented. The architecture of the model consisted of two fully connected hidden layers of 256 and 128 nodes, respectively. The information passing through each node was passed through by applying a ReLu function. To optimise the weight of the nodes, the Adam optimiser was applied, based on fast convergence and low memory requirements. The aim of the optimiser is to minimise the loss function over a set number of epochs, which was set to 50. The loss function applied was the L1 loss also known as the mean absolute error. This decision was based on the fact that potential anomalies might still be present in the data, which could cause a vanishing gradient for other loss functions. To optimise the hyperparameters described here, an optimisation technique was applied, which is known as Bayesian Optimisation. This is a Gaussian process to minimise the loss function with the optimal set of parameters. After training, performance metrics were calculated by means of the root-mean squared error and the coefficient of determination.

The process of predicting the power output using an ANN is a black-box, making it difficult to know which features contribute to a certain prediction. To visualise the prediction, SHAP values were calculated for a random sample of the data set. SHAP values find their origin in cooperative game theory and tell us what each feature contributes to the predicted outcome. To explain the contribution, a kernel explainer function was implemented in the model.

With the tuned model, the training process began by splitting the data into a test, validation, and test set. The test set is a subset of the total SCADA data set that has not yet been seen by the model, and it is needed to see how it reacts to new data. The performance of the model was found to have a coefficient of determination of around 0.995 and an MAE of 17.2 kW. The model's predictions were used to construct a power curve that was compared to the manufacturer power curve. Furthermore, the AEP of each of the turbines was calculated for the test set and the percentage difference with the actual measured AEP was plotted to have an insight into the model's performance. The negative percentage difference in AEP gave an indication of which turbines were underperforming compared to the predicted production. The residual analysis provided a deeper understanding of the performance. By computing the bias and standard

deviation of the residuals, outliers could be detected in the time series with a confidence interval of 99%. To increase the confidence level in the signal, a low-pass filter was applied, which was the centred rolling mean. An outlier detection based on the rolling mean was applied to identify the window of time in which a turbine was underperforming and a root cause analysis for this time window was performed. By showing the power curve in combination with the SHAP values, patterns in the data were discovered that were confirmed by the maintenance reports at hand.

This methodology was systematically applied throughout the wind farm, resulting in the development of fourteen models trained separately, each trained on its respective cluster. These models were applied to detect deviations in turbine performance. Case studies were presented for selected clusters, showcasing the analysis of individual turbines' behaviour and performance. For each group, location, performance metrics, time series of residuals, and AEP comparisons were discussed. The methodology was validated by the use of maintenance reports, which were shown to give an explanation for most of the detected cases.

Future work

Throughout the literature study it became evident that the landscape of power curve and performance modelling is widely studied with many different algorithms applied. It is the fact that so many approaches have been studied that underlines that it is very dependent on the data you have at hand. Due to the different terrain characteristics of this particular site, not every approach showed to have the desired outcome, it is simply hard to compare apples with pears. In future work, the detected underperformance could be coupled with filtering prior to training. The detected instances in the training set influenced the predictions for the test set and could potentially have lead to a bias on the residuals. Furthermore, in case of SCADA data with a higher temporal resolution the use of the Long Short-Term Memory(LSTM) algorithm shows to gain popularity in power performance analysis. The LSTM algorithm can be applied in a similar methodology as described in this thesis.

References

[1] Directorate-General for Energy. “Member States agree new ambition for expanding offshore renewable energy”. In: *European Commission* (2023).

[2] Tim McPhie et al. *Boosting Offshore Renewable Energy for a Climate Neutral Europe*. 2020. URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_20_2096.

[3] Philip Totaro. *Denmark’s under-performing and curtailed wind turbines have lost DKK 2.1bn in asset revenue*. Oct. 2020. URL: https://renewablesnow.com/news/denmarks-under-performing-and-curtailed-wind-turbines-have-lost-dkk-21bn-in-asset-revenue-800172/ (visited on 06/20/2023).

[4] Ivan Komusanac et al. “Wind Energy in Europe: 2021 statistics and the outlook for 2022 and 2026”. In: *Wind Europe* (Feb. 2022).

[5] Andrew Kusiak et al. “On-line monitoring of power curves”. In: *Renewable Energy* 34.6 (June 2009), pp. 1487–1493. DOI: 10.1016/j.renene.2008.10.022.

[6] IEC Central Secretary. *Power performance measurements of electricity producing wind turbines*. Standard. Geneva, CH: International Electrotechnical Commission, 2020.

[7] Klaus Kaiser et al. “Turbulence correction for power curves”. In: *Wind Energy: Proceedings of the Euromech Colloquium*. Springer. 2007, pp. 159–162.

[8] Axel Albers. “Turbulence and shear normalisation of wind turbine power curve”. In: *European Wind Energy Conference and Exhibition 2010, EWEC 2010* 6 (Jan. 2010), pp. 4116–4123.

[9] Martin O. L. Hansen. *Aerodynamics of wind turbines*. New York, NY: Routledge, 2015. 173 pp.

[10] Daniel Villanueva et al. “A Review on Wind Turbine Deterministic Power Curve Models”. In: *Applied Sciences* 10.12 (June 2020), p. 4186. DOI: 10.3390/app10124186.

[11] M. Richmond et al. “Evaluation of an offshore wind farm computational fluid dynamics model against operational site data”. In: *Ocean Engineering* 193 (Dec. 2019), p. 106579. DOI: 10.1016/j.oceaneng.2019.106579.

[12] Yen-Cheng Chiang et al. “Power Prediction of Wind Farms via a Simplified Actuator Disk Model”. In: *JMSE* 8.8 (Aug. 2020), p. 610. DOI: 10.3390/jmse8080610.

[13] Olivier Janssens et al. “Data-driven multivariate power curve modeling of offshore wind turbines”. In: *Engineering Applications of Artificial Intelligence* 55 (Oct. 2016), pp. 331–338.

[14] M. Lydia et al. “Advanced Algorithms for Wind Turbine Power Curve Modeling”. In: *IEEE Trans. Sustain. Energy* 4.3 (July 2013), pp. 827–835. DOI: 10.1109/TSTE.2013.2247641.

[15] E Gonzalez et al. “On the use of high-frequency SCADA data for improved wind turbine performance monitoring”. In: *J. Phys.: Conf. Ser.* 926 (Nov. 2017), p. 012009. DOI: 10.1088/1742-6596/926/1/012009.

[16] Giusepinna Ciulla et al. “Modelling and analysis of real-world wind turbine power curves: Assessing deviations from nominal curve by neural networks”. In: *Renewable Energy* 140 (Sept. 2019), pp. 477–492. DOI: 10.1016/j.renene.2019.03.075.

[17] Francis Pelletier et al. “Wind turbine power curve modelling using artificial neural network”. In: *Renewable Energy* 89 (Apr. 2016), pp. 207–214. DOI: 10.1016/j.renene.2015.11.065.

[18] Bartolomé Manobel et al. “Wind turbine power curve modeling based on Gaussian Processes and Artificial Neural Networks”. In: *Renewable Energy* 125 (Sept. 2018), pp. 1015–1020. DOI: 10.1016/j.renene.2018.02.081.

[19] S. Barber et al. “Improving site-dependent power curve prediction accuracy using regression trees”. In: *J. Phys.: Conf. Ser.* 1618.6 (2020). DOI: 10.1088/1742-6596/1618/6/062003.

[20] Barbara Illowsky et al. *Introductory Statistics*. en. Houston: OpenStax, 2018.

[21] Bin Wang et al. “The confidence intervals correspond to 3-sigma rule of the normal distribution.” In: *Plos One* (Mar. 2015). DOI: 10.1371/journal.pone.0118537.g003.

[22] Varun Chandola et al. “Anomaly detection: A survey”. en. In: *ACM Comput. Surv.* 41.3 (July 2009), pp. 1–58. DOI: 10.1145/1541880.1541882.

[23] Rezaul Karim et al. “A Survey on Anomaly Detection Strategies”. In: *Second International Conference on Image Processing and Capsule Networks: ICIPCN 2021*. Ed. by Joy Iong-Zong Chen et al. Vol. 300. Cham: Springer International Publishing, 2022, pp. 289–297.

[24] Ameet V Joshi. *Machine Learning and Artificial Intelligence*. Cham: Springer International Publishing, 2023.

[25] Tommaso Barbariol et al. “A Review of Tree-Based Approaches for Anomaly Detection”. In: *Control Charts and Machine Learning for Anomaly Detection in Manufacturing*. Ed. by Kim Phuc Tran. Cham: Springer International Publishing, 2022, pp. 149–185.

[26] Sinvaldo R. Moreno et al. “Wind turbines anomaly detection based on power curves and ensemble learning”. In: *IET Renewable Power Generation* 14.19 (Dec. 2020), pp. 4086–4093. DOI: 10.1049/iet-rpg.2020.0224.

[27] Scott Cost et al. “A weighted nearest neighbor algorithm for learning with symbolic features”. In: *Mach Learn* 10.1 (1993), pp. 57–78. DOI: 10.1007/BF00993481.

[28] Rory Morrison et al. “Anomaly detection in wind turbine SCADA data for power curve cleaning”. In: *Renewable Energy* 184 (Jan. 2022), pp. 473–486. DOI: 10.1016/j.renene.2021.11.118.

[29] Fei Tony Liu et al. “Isolation Forest”. In: *2008 Eighth IEEE International Conference on Data Mining*. Pisa, Italy: IEEE, Dec. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[30] Markus M. Breunig et al. “LOF: Identifying Density-Based Local Outliers”. In: *SIGMOD Rec.* 29.2 (June 2000), pp. 93–104. DOI: 10.1145/335191.335388.

[31] Evangelos Papatheou et al. “A Performance Monitoring Approach for the Novel Lillgrund Offshore Wind Farm”. en. In: *IEEE Trans. Ind. Electron.* 62.10 (Oct. 2015), pp. 6636–6644. DOI: 10.1109/TIE.2015.2442212.

[32] John Thomas Lyons et al. “Applied Machine Learning Techniques for Performance Analysis in Large Wind Farms”. In: *Energies* 14.13 (June 2021), p. 3756. DOI: 10.3390/en14133756.

[33] IEC Central Secretary. *Power performance measurements of electricity producing wind turbines based on nacelle anemometry*. Standard. Geneva, CH: International Electrotechnical Commission, 2022.

[34] Lars Buitinck et al. “API design for machine learning software: experiences from the scikit-learn project”. In: *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013, pp. 108–122.

[35] David Arthur et al. “k-means++: the advantages of careful seeding”. In: *ACM-SIAM Symposium on Discrete Algorithms*. 2007.

[36] Atiq Ur Rehman et al. “Unsupervised outlier detection in multidimensional data”. In: *J Big Data* 8.1 (Dec. 2021), p. 80. DOI: 10.1186/s40537-021-00469-z.

[37] Xiangyu Yang et al. “Influence analysis of rime icing on aerodynamic performance and output power of offshore floating wind turbine”. In: *Ocean Engineering* 258 (Aug. 2022), p. 111725. DOI: 10.1016/j.oceaneng.2022.111725.

[38] Wikipedia. *Basic idea of LOF: comparing the reachability (density) of a point with the reachability (density) of its neighbors to detect Outliers.* Image. 2010. URL: https://en.wikipedia.org/wiki/Local_outlier_factor#/media/File:LOF-idea.svg.

[39] Yadolah Dodge. “Spearman Rank Correlation Coefficient”. In: *The Concise Encyclopedia of Statistics*. New York, NY: Springer New York, 2008, pp. 502–505.

[40] Patrick Schober et al. “Correlation Coefficients: Appropriate Use and Interpretation”. In: *Anesthesia & Analgesia* 126.5 (May 2018), pp. 1763–1768. DOI: 10.1213/ANE.0000000000002864.

[41] Adhistya Permanasari et al. “Utilization of Neural Network for Disease Forecasting”. In: *Proceedings 59th ISI World Statistics Congress*. Aug. 2013, pp. 549–554.

[42] Tomasz Szandala. “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks”. In: *CoRR* abs/2010.09458 (Oct. 2020). arXiv: 2010.09458. URL: https://arxiv.org/abs/2010.09458.

[43] Arun Kumar Dubey et al. “Comparative Study of Convolution Neural Network’s Relu and Leaky-Relu Activation Functions”. In: *Lecture Notes in Electrical Engineering* (2019).

[44] Tomasz Szandała. “Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks”. In: *Bio-inspired Neurocomputing*. Vol. 903. Series Title: Studies in Computational Intelligence. Singapore: Springer Singapore, 2021, pp. 203–224. DOI: 10.1007/978-981-15-5495-7_11.

[45] Tijmen Tieleman et al. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural networks for machine learning* 4.2 (2012), pp. 26–31.

[46] Diederik P. Kingma et al. *Adam: A Method for Stochastic Optimization*. en. Jan. 2017. URL: http://arxiv.org/abs/1412.6980.

[47] Peter J. Huber. “Robust Estimation of a Location Parameter”. In: *The Annals of Mathematical Statistics* 35.1 (1964). Publisher: Institute of Mathematical Statistics, pp. 73–101. DOI: 10.1214/aoms/1177703732.

[48] Scott M. Lundberg et al. “A unified approach to interpreting model predictions”. In: *CoRR* abs/1705.07874 (2017). URL: http://arxiv.org/abs/1705.07874.

A Figures

In this Appendix the figures are shown that were too large to include in the main chapters.

A.1. Correlation Matrix

Figure A.1 is a representation of the full correlation matrix. It should be noted that in this matrix also the identification numbers of the events have been filtered out. Additionally, descriptions have been removed since they are not valuable for training.

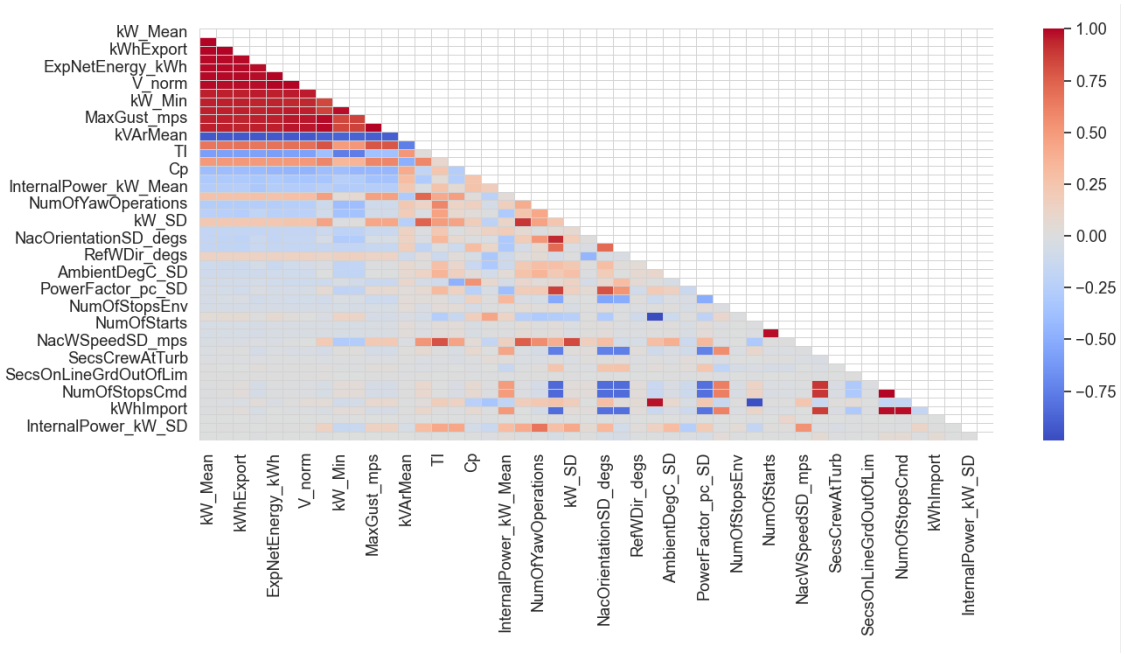


Figure A.1: Full correlation matrix for wind turbine A01

A.2. Hyperparameter Optimisation

Table A.1 represents the 20 different combinations that were searched in to find the optimal combination of hyperparameters.

Table A.1: Results of the Bayesian Optimization process, for 20 optimization iterations

Optimum L1-Loss	Learning Rate	Hidden Size 1	Hidden Size 2	Number of Epochs
17.8534	0.0245	129	214	32
18.2339	0.0022	116	152	20
18.7210	0.0003	202	75	37
18.5911	0.0654	100	255	33
18.5042	0.0068	101	68	29
18.0771	0.0016	107	251	15
18.7230	0.0002	196	137	49
18.2022	0.0025	234	195	25
23.5765	0.0001	247	172	22
23.6346	0.0001	136	110	36
17.8753	0.0253	118	192	25
26.9391	0.0237	256	64	8
18.1512	0.0174	223	227	28
18.9340	0.0946	125	244	41
19.7226	0.1000	128	256	5
17.5122	0.0014	224	243	50
17.8839	0.1000	185	243	50
17.6597	0.0066	203	255	50
17.9865	0.0036	216	256	36
18.8578	0.0001	232	256	50

B

Cluster Analyses

In this appendix, the clusters not presented in the main body, are described. Please, be aware that the analysis is much more concise.

B.1. Cluster 4

Cluster 4 is a single turbine consisting of A04. No power anomalies were detected in this turbine, indicating that the model learns sufficiently from the data. Furthermore, it should be noted that the AEP difference between the prediction and the actual, presented in Figure B.3, is within a range of 0.175%. This indicates that the model can predict the AEP with high precision.

Table B.1: Performance metrics of cluster 4

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A04	20.03	14.11	0.998
Test	A04	19.44	14.79	0.997

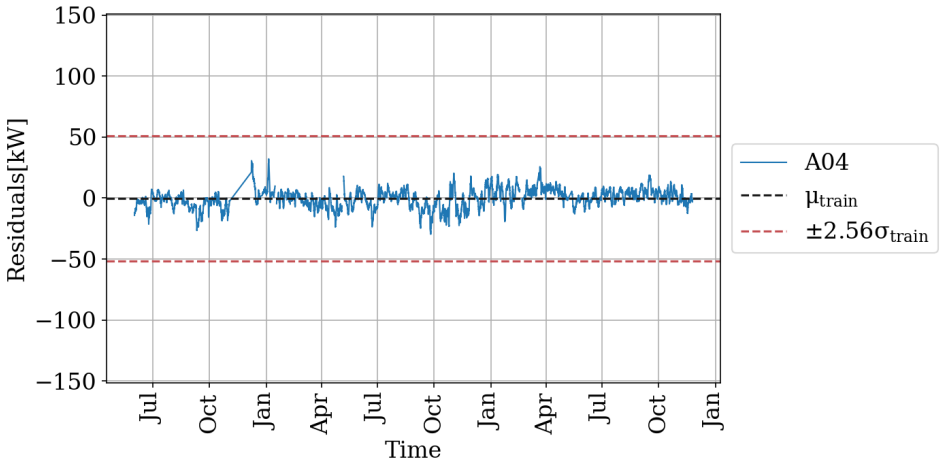


Figure B.1: Residual time series for the training set of cluster 4

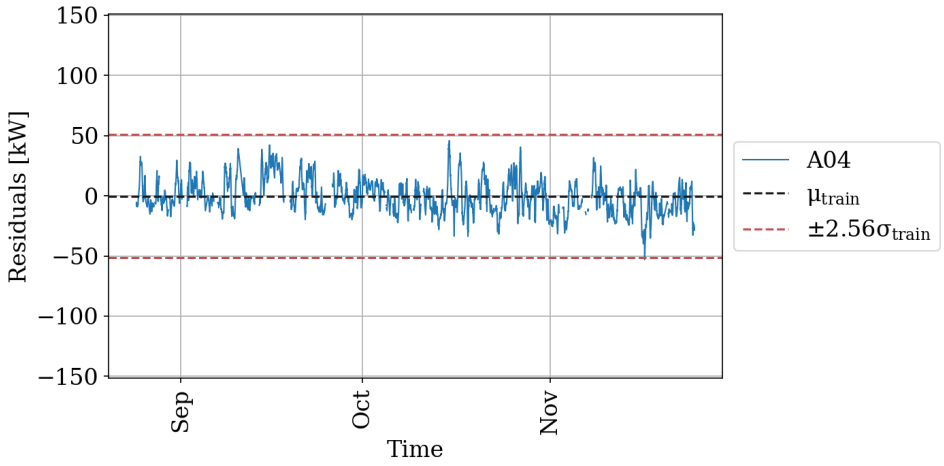


Figure B.2: Residual time series for the test set of cluster 4

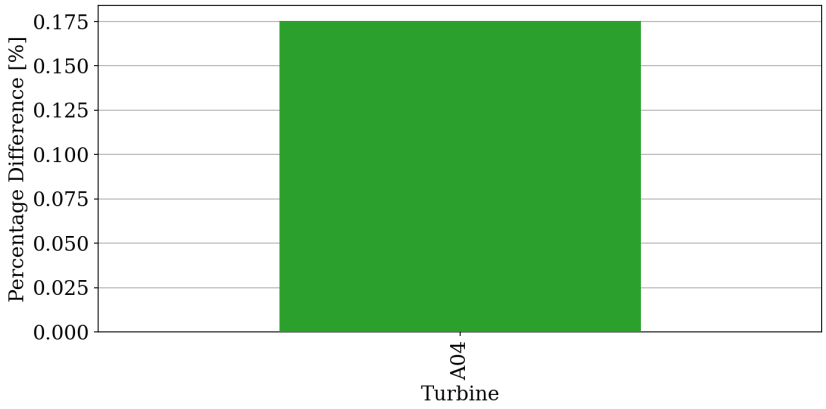


Figure B.3: Percentage difference in AEP of cluster 4

B.2. Cluster 8

Cluster 8 is a single turbine consisting of A57. In the test set, as can be seen in Figure B.4, an instance of underperformance is detected that was the result of a failing lubricating oil pump. This is translated into a lower AEP which can be seen in Figure B.6.

Table B.2: Performance metrics of cluster 8

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A08	22.10	14.89	0.995
Test	A08	27.24	20.45	0.993

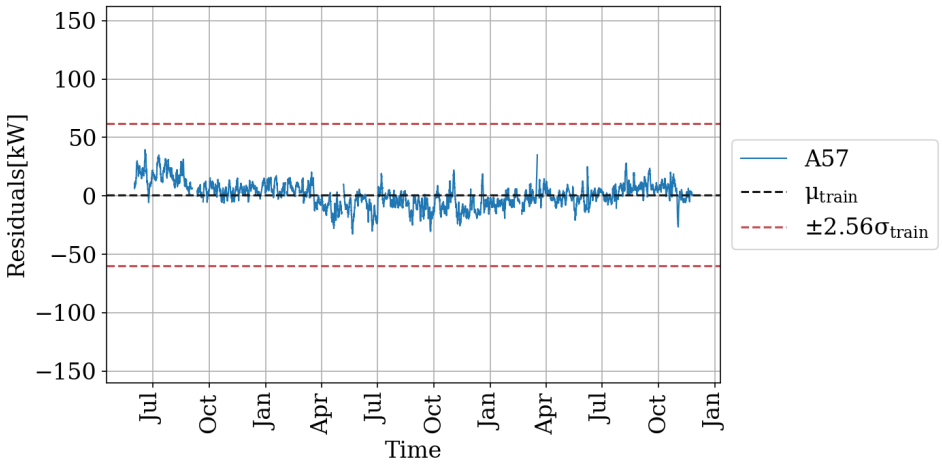


Figure B.4: Residual time series for the training set of cluster 8

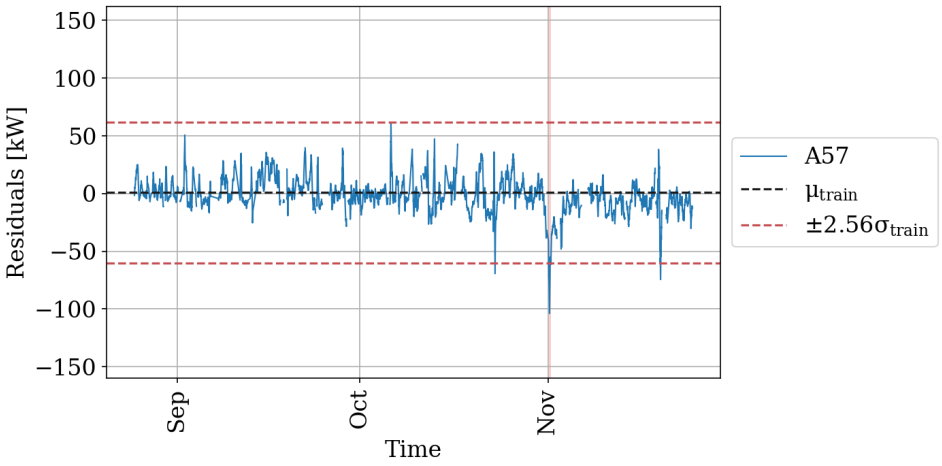


Figure B.5: Residual time series for the test set of cluster 8

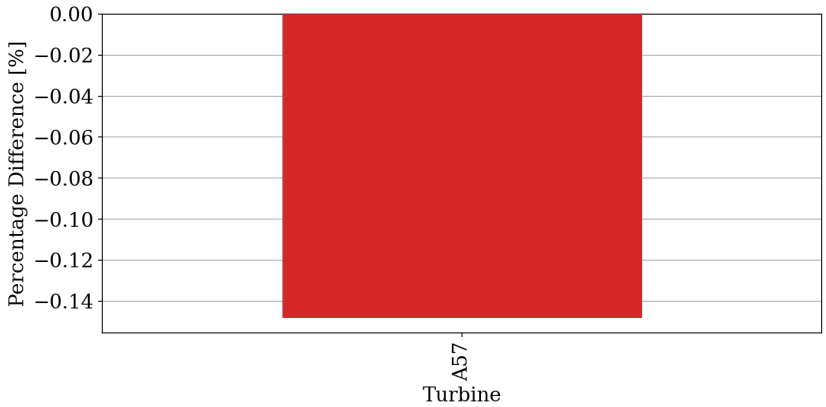


Figure B.6: Percentage difference in AEP of cluster 8

B.3. Cluster 9

Cluster 9 consists of two turbines A34 and A35. In the test set, as can be seen in Figure B.4, an instance of overperformance is detected for A34. This was verified by the maintenance reports provided for that month. This is translated into a higher AEP which can be seen in Figure B.6.

Table B.3: Performance metrics of cluster 9

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A34	27.09	18.29	0.994
	A35	25.33	17.80	0.994
Test	A34	26.50	19.29	0.994
	A35	19.46	13.90	0.996

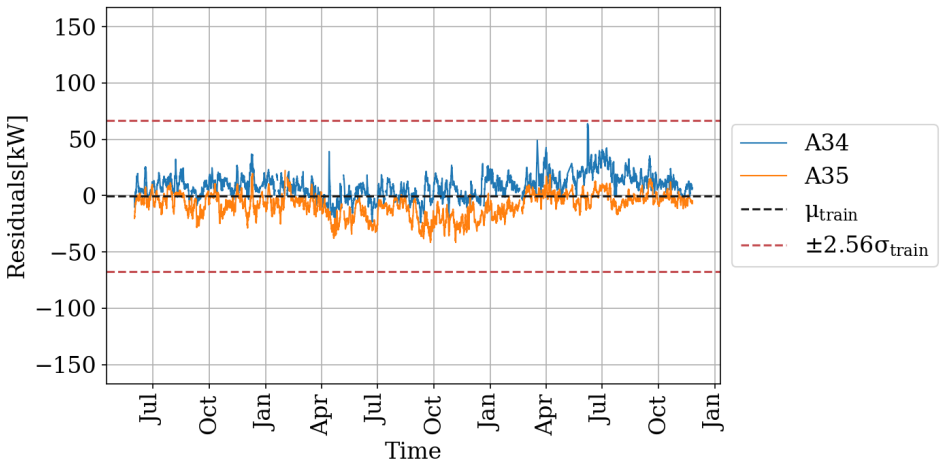


Figure B.7: Residual time series for the training set of cluster 9

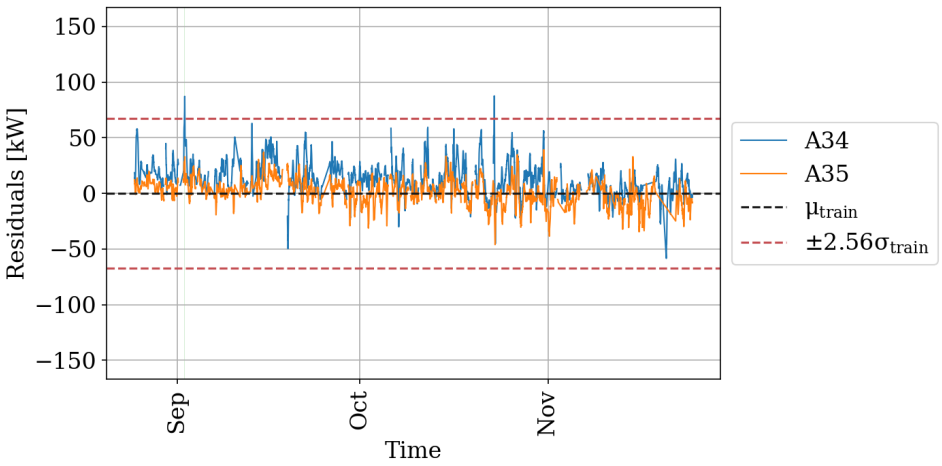


Figure B.8: Residual time series for the test set of cluster 9

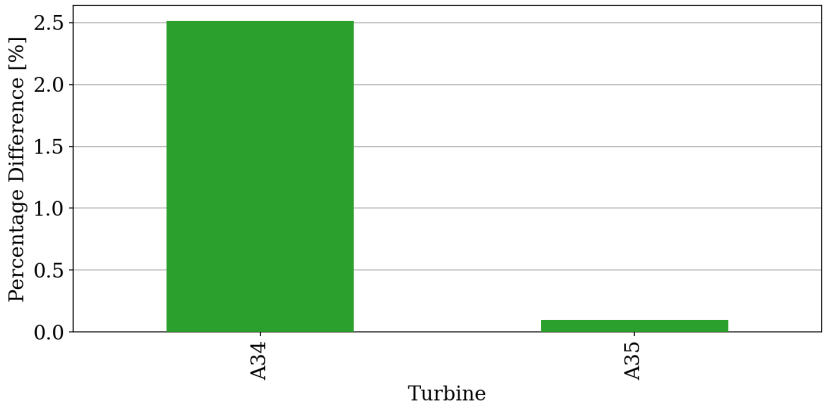


Figure B.9: Percentage difference in AEP of cluster 9

B.4. Cluster 10

Cluster 10 consists of turbine A09, A11, A40, and A41 and is located in the centre of the wind farm. Looking at the model's performance metrics in Table B.4, A41 draws the attention. In the RMSE it can be seen that the data set contains more outliers compared to the other turbines. Furthermore, in case of A41, the model performs much better for the test set.

Table B.4: Performance metrics of cluster 10

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A09	26.11	16.68	0.994
	A11	27.89	18.79	0.993
	A40	20.07	14.71	0.996
	A41	31.00	20.60	0.990
Test	A09	27.90	19.60	0.993
	A11	27.89	18.79	0.993
	A40	25.68	20.68	0.994
	A41	29.38	19.46	0.992

In Figure B.10, the bias in the residuals of A41 can clearly be seen in the period between October of the second year and August of the third year. In this period the turbine is suffering from a failing pitch control system. In an attempt to solve the error, the anemometer is replaced twice in January and March of the third year. Finally in August, the wind vane on the turbine is adjusted which appeared to have solved this issue. Knowing this, in future models this period of time should be removed from the training data to avoid introducing a bias that underpredicts the power output.

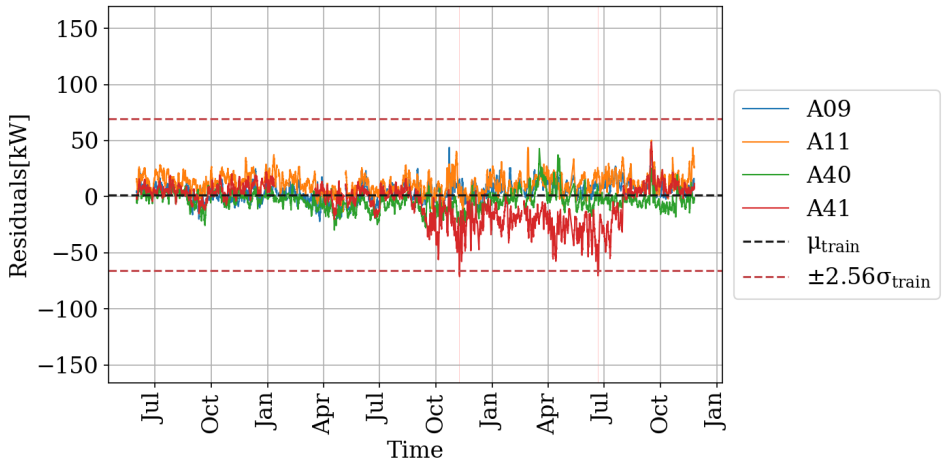


Figure B.10: Residual time series for the training set of cluster 10

Two peaks of overperformance were detected in Figure B.11 for both A09 and A11, which is also seen in the maintenance reports. For turbine A40 it can be seen that there is a negative bias of the residuals. This can be seen in Figure B.12

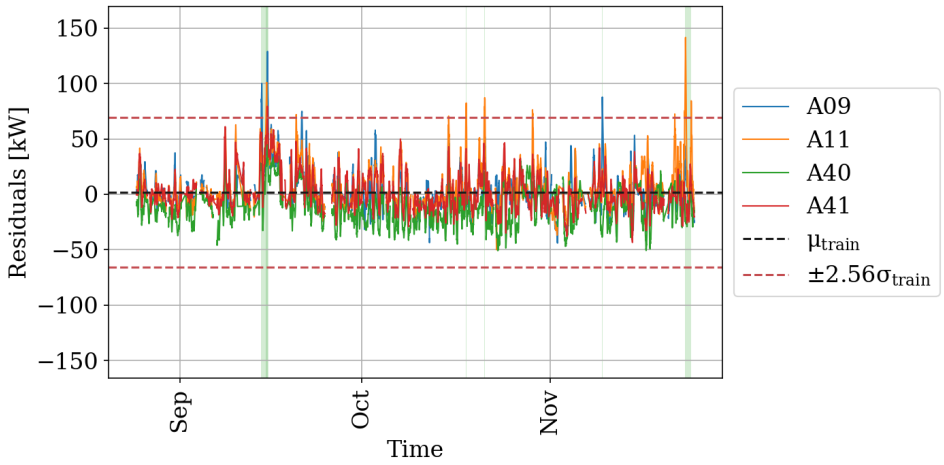


Figure B.11: Residual time series for the test set of cluster 10

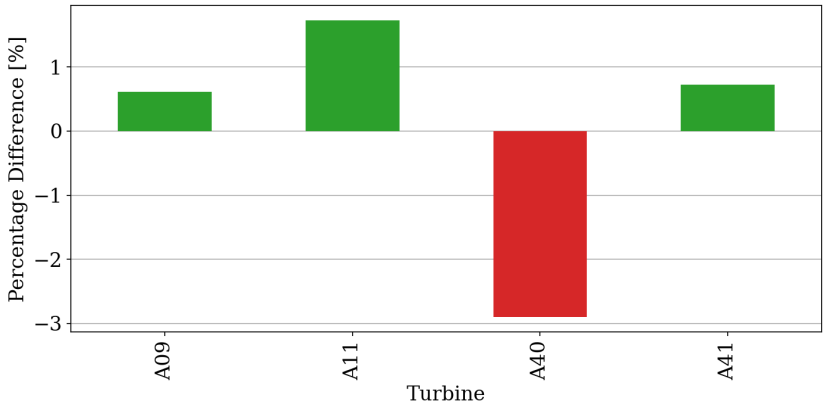


Figure B.12: Percentage difference in AEP of cluster 10

B.5. Cluster 11

Cluster 11 consists of A48 to A56 and is located to the west of the wind farm. The performance metrics in Table B.5 show that the data contain a large amount of outliers in turbine A49, A54, and A55. Certainly the RMSE of turbine A54 is an indication that additional filtering might be required for a certain timeframe.

Table B.5: Performance metrics of cluster 11

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A48	18.22	13.26	0.996
	A49	27.46	17.75	0.991
	A50	21.67	15.53	0.995
	A51	16.50	11.94	0.997
	A52	17.59	12.68	0.996
	A53	22.50	16.74	0.995
	A54	36.30	20.73	0.985
	A55	28.59	18.44	0.992
	A56	19.28	13.86	0.996
Train	A48	24.29	19.24	0.993
	A49	22.98	17.71	0.994
	A50	28.60	22.29	0.990
	A51	25.85	20.33	0.992
	A52	22.99	17.33	0.994
	A53	31.35	25.07	0.989
	A54	19.77	15.80	0.996
	A55	23.74	17.91	0.994
	A56	26.22	20.29	0.993

In Figure B.13, the reason behind the large RMSE becomes clear. In April turbine A51 was struck by lightning, which seems to have affected the sensors of the turbine. For A55 the anemometer was replaced in May, but, in the end of July, it was found that the slope and offset were incorrectly entered in the controller.

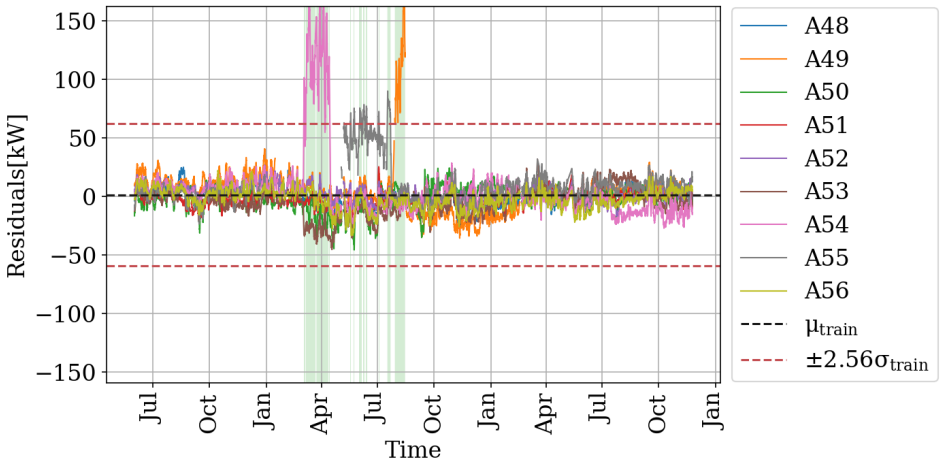


Figure B.13: Residual time series for the training set of cluster 11

In Figure B.13 it can be seen that for the test set all turbines seem to exhibit a bias in the residuals, which is confirmed by Figure 6.3.

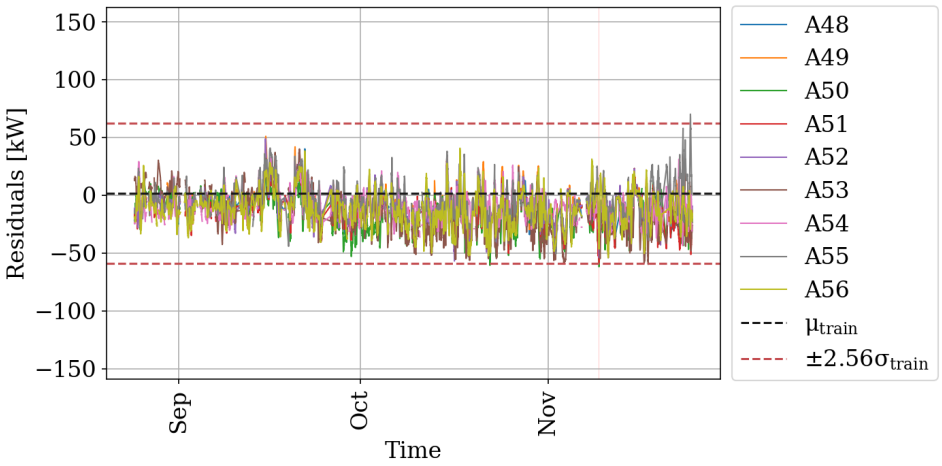


Figure B.14: Residual time series for the test set of cluster 11

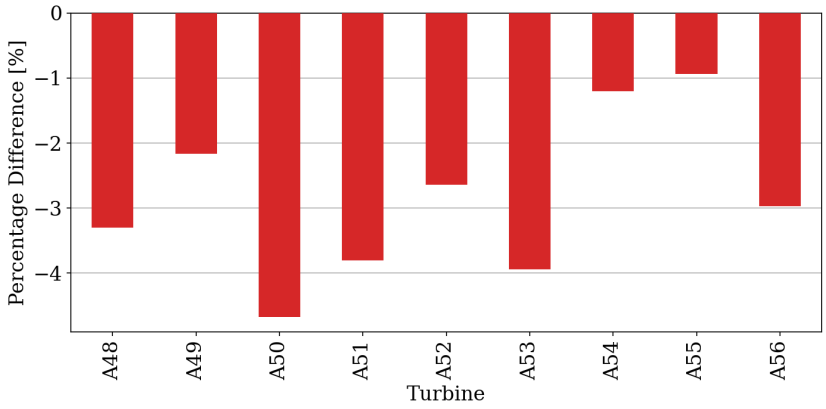


Figure B.15: Percentage difference in AEP of cluster 11

B.6. Cluster 12

Cluster 12 consists of one turbine A36. In the test set, as can be seen in Figure B.16, an instance of overperformance is detected. This is considered a false positive, as it appears to occur right after a maintenance event due to lightning impact, which also occurred in A54; see Section B.5. Knowing that this is detected suggests that the minimum number of points in the running mean should be increased.

Table B.6: Performance metrics of cluster 12

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A36	38.62	19.21	0.985
Test	A36	26.12	20.78	0.993

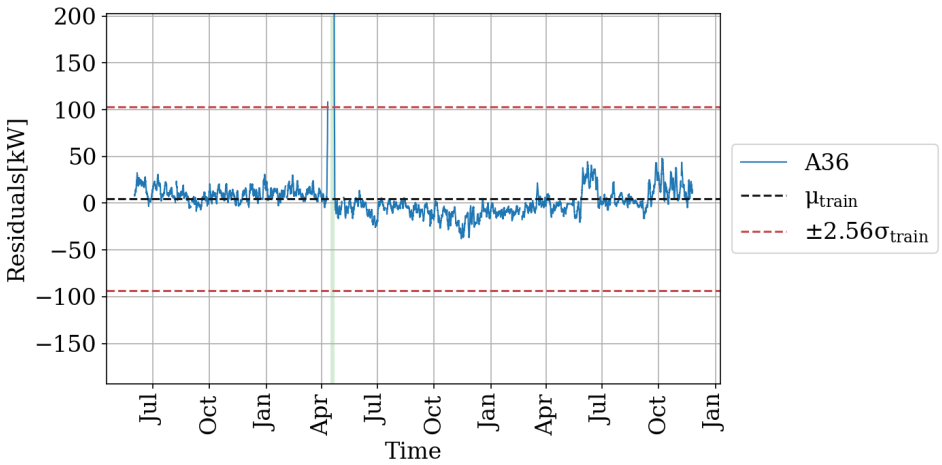


Figure B.16: Residual time series for the training set of cluster 12

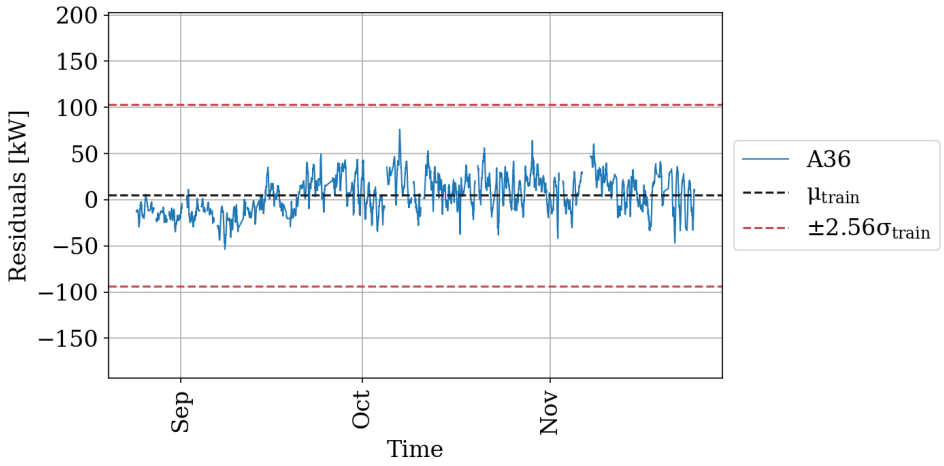


Figure B.17: Residual time series for the test set of cluster 12

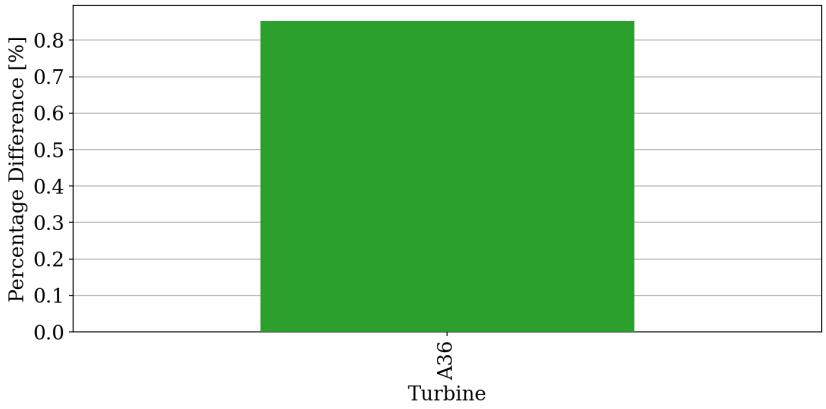


Figure B.18: Percentage difference in AEP of cluster 12

B.7. Cluster 13

Cluster 13 consists of A37 to A39 and is located in the east of the wind farm on the plateau. In the training set no anomalies were detected, but a negative trend for turbine A39 can be seen starting from February up until the end of the second year.

Table B.7: Performance metrics of cluster 13

	Turbine	RMSE [kW]	MAE [kW]	R2
Train	A37	22.29	16.17	0.995
	A38	21.18	14.79	0.996
	A39	22.40	16.07	0.995
Test	A37	19.20	14.14	0.996
	A38	22.68	17.08	0.995
	A39	22.33	17.16	0.995

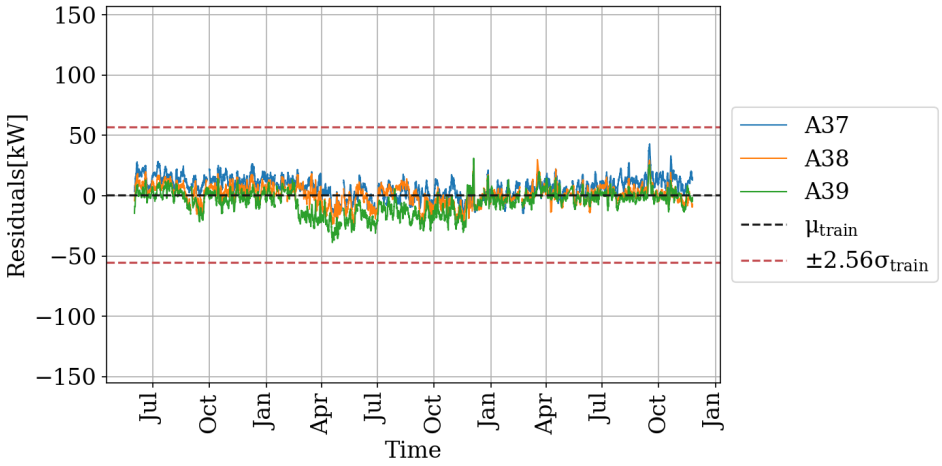


Figure B.19: Residual time series for the training set of cluster 13

In Figure B.20 it can be seen that the turbines in cluster 13 have a negative slope in residuals toward the end of November, at which the algorithm detects one peak of underperformance. The lower efficiency of turbine A39 can also be found in the maintenance reports. In October an operating efficiency below 96% was recorded, which is considered to be underperformance.

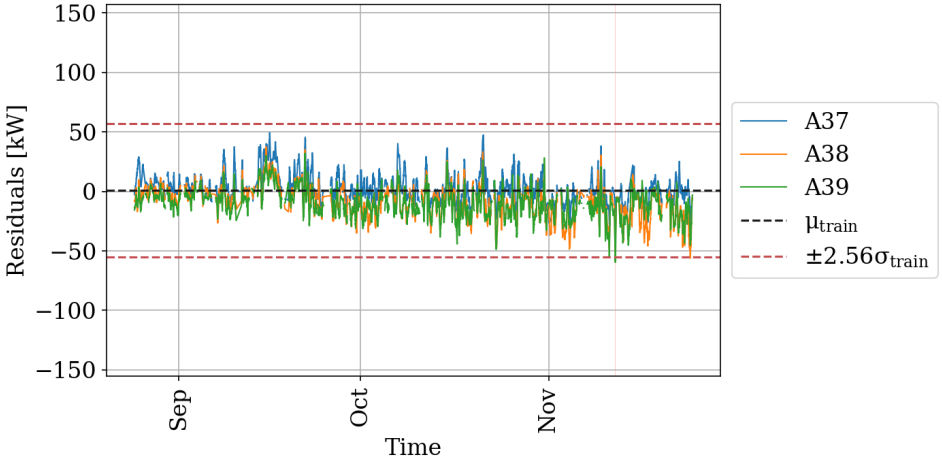


Figure B.20: Residual time series for the test set of cluster 13

Looking at Figure B.21, the underperformance in A38 can be explained by a yaw brake error in October.

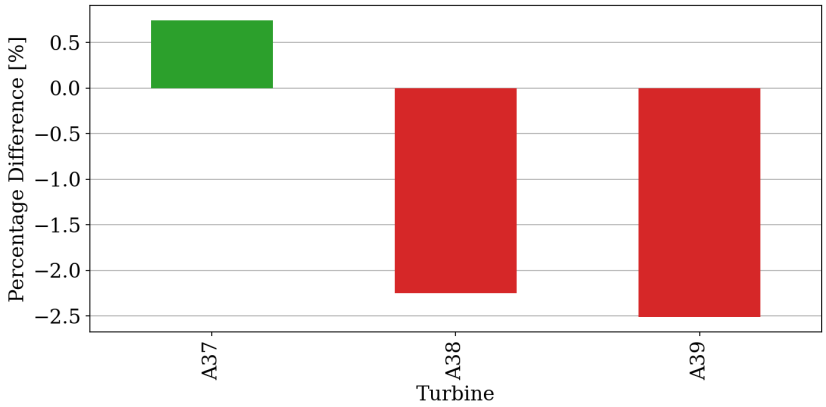


Figure B.21: Percentage difference in AEP of cluster 13