The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

School of Computer Science

20125049

Supervisor: Isaac Triguero Velazquez

Module Code: COMP3003

2022/04

Wind Turbine Fault Detection Based on Machine Learning Techniques and Knowledge Transfer

Submitted April 2022, in partial fulfilment of

the conditions for the award of the degree **Computer Science with Artificial Intelligence BSc.**

20125049

School of Computer Science

University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature ________ZC____________

Date __30___/__04___/__2022___

I hereby declare that I have all necessary rights and consents to publicly distribute this dissertation via the University of Nottingham's e-dissertation archive.*

*Only include this sentence if you do have all necessary rights and consents. For example, if you have including photographs or images from the web or from other papers or documents then you need to obtain explicit consent from the original copyright owner. If in doubt, delete this sentence. See Copyright Information for more details.

**Only include this sentence if there is some reason why your dissertation should not be accessible for some period of time, for example if it contains information which is commercially sensitive or might compromise an Intellectual Property claim. If included, fill in the date from which access should be allowed.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Isaac Triguero Velazquez for his generous and patient guidance and help in the whole project development process. I would like to thank the school and department for providing me with this opportunity to realize my project vision. I also want to thank my parents for their company and support.

Abstract

With the intensification of global warming, governments and private groups began to pay more attention to the development of clean and renewable energy. As the main new energy, the maintenance and fault prediction of the wind farms is of great significance. Among several common maintenance methods, predictive maintenance can use the machine learning method to detect and predict faults according to Supervisory Control and Data Acquisition (SCADA) data. This paper proposes machine learning models for wind turbine anomaly detection, which uses an artificial neural network and AdaBoost to accurately classify the SCADA data samples of wind turbines. Our model finally detected 93% of the anomalies in the test set. In addition, this work also transfers the learned machine learning model to another unlabeled wind turbine SCADA dataset to realize the knowledge transfer. The model can still detect 88% of the anomalies in the target domain dataset, which proves that transfer learning has great development prospects in the field of predictive maintenance of wind farms. It uses the pre-trained model without additional training. The next work will try to expand the application of the transfer learning model to more industrial fields.

Contents

List of Abbreviations

ANN	Artificial Neural Network
AUC	Area Under Curve
CMS	Condition Monitoring Systems
CM	Condition Monitoring
CNN	Convolutional Neural Network
ENN	Edited Nearest Neighbor
FNN	Feedforward Neural Network
GP	Gaussian Process
LDT	Levenmouth Demonstration Turbine
LSTM	Long and Short Term Memory
MLP	Multi-layer Perceptron
NBM	Normal Behavior Model
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SCADA	Supervisory Control and Data Acquisition
UoN	University of Nottingham
WT	Wind Turbine

Chapter 1

Introduction and Motivation

With the expansion of the earth's population and the industrialization of countries, human demand for energy has reached an unprecedented level. Among them, fossil fuels, as the main energy source, have caused great hidden dangers to the earth. There is an inseparable relationship between the combustion of coal, oil and natural gas and the rising greenhouse gas emissions in the earth's atmosphere. They are the primary cause of climate change. In order to alleviate the crisis caused by fossil energy, more and more countries begin to support the development and application of renewable energy, especially wind energy.

Through technological innovation and economic development, the scale of the global wind power market has almost quadrupled in the past decade and has become the most important alternative to traditional fossil energy. By the end of 2020, the global accumulated capacity of wind farms has reached 743 GW, with a 53% year-to-year increase [1]. Compared with other energy sources, wind power generation has little impact on the environment. As long as necessary wind power generation devices are built, unlike hydropower generation, which requires the construction of a dam for water storage and power generation, it will inevitably make some irrecoverable changes to the environment, affect the local ecological development and the original natural landscape, and sometimes even affect the lives of indigenous people. For the greenhouse gas emissions caused by power generation, coal-fired thermal power is the most serious and wind power is the least. From an economic point of view, wind power has great advantages. It only needs to build power generation equipment in the early stage and fewer maintenance costs in the later stage.

The wind turbine studied in this paper is a very common wind power generation equipment. The wind turbine is composed of the generator, bearing, gearbox, pitch, rotor, and other mechanical components [2]. Among them, the rotor is used to receive wind power and convert it into electric energy through the generator. However, the operation and maintenance of wind farms become more challenging because wind turbines usually operate for a long time and are usually located in sparsely populated areas. Damage to some key steps in wind turbines can lead to serious power loss and maintenance costs. Due to the unknown environmental conditions, the initial periodic maintenance strategy can not work well, which has aroused people's interest in using artificial intelligence to maintain wind turbines [3].

Generally speaking, there are several ways to maintain wind turbines [4]:

Corrective Maintenance: In this way, maintenance can be carried out only after the fault occurs or the faulty components have been replaced. This is the most popular maintenance method and is very effective in solving widget failures because it has the cheapest implementation cost. The main problem of this method is that when the fault occurs in some emergency situations (such as when the wind turbine is running at high speed), stopping the fan and starting to detect the cause of the fault will cause great economic losses.

Preventive Maintenance: Workers will regularly maintain the wind turbine and regularly replace components [5]. It can reduce the probability of failure and reduce the downtime caused by failure. The time interval between periodic maintenance is determined by domain experts based on the historical performance of components. This method is to replace or maintain the parts in advance before they fail, which is a waste of resources. This is also one of the purposes of introducing predictive maintenance because it can predict faults just before they occur.

Predictive Maintenance: Predictive maintenance is a technology that notifies equipment and components that may fail and replaces them at the right time [6] which includes estimating the remaining useful life of equipment, detecting anomalies in equipment, and diagnosing equipment fault types [7]. Predictive maintenance usually includes two methods.

On the one hand, the traditional condition monitoring system is developed, and sensors are added to the wind turbine components to detect factors such as temperature and oil quality. However, the price of installing these condition monitoring systems is often as high as tens of thousands of dollars, and high professional knowledge is required. In addition, the vibration signal analysis [8][9][10] is regarded as the main concept of this method, which is vulnerable to the strong influence of sensor location and surrounding environment [11].

On the other hand, the principle of data-driven modeling method is to obtain a large amount of data from the real system and use artificial intelligence technology to develop the model to help detect faults. It does not require staff to have professional knowledge of system evaluation. The real data used in this method usually adopts SCADA data, which is the abbreviation of monitoring and data acquisition. It is a real-time data monitoring system, which is used to monitor the status and operation control of various industries and projects [8]. As the monitoring system of wind turbine, SCADA can collect multiple variables related to operating characteristics (including wind speed, power, temperature, etc.) without installing additional sensors in various operating states of components. Typically, the SCADA system will use a sampling frequency of 10 minutes and provide data labels representing the operating status of the wind turbine.

In the last two decades, artificial intelligence technology has been successfully applied to the field of wind turbine fault detection and prediction using SCADA data. For example, in [12], the multiple linear regression model is used to judge the failure of the gearbox by predicting the bearing temperature of the gearbox; In [13], the author takes the four characteristics of state temperature, residual power output, nacelle temperature, and shaft speed as input values to predict the bearing temperature of gearbox using Multilayer Perceptron (MLP), and the prediction results are better than the linear regression model; In [7], the author established the characteristic behavior model of key wind turbine components through Long Short-Term Memory (LSTM) and extreme gradient boosting (XGBoost).....

This study will realize the transfer learning attempt between SCADA data of two offshore wind farms. For the first dataset, this experiment generates the operation status label through the records in the wind turbine alarm log; For the second dataset, a clustering model is established to generate abnormal data and normal data.

1.1 Research Objectives

The objectives of this thesis are:

- Analyze the obtained wind turbine dataset and determine how to use this information to develop a machine learning model for fault prediction. For the data recorded with historical fault information, label the operation dataset based on fault record; For unlabelled data sets, unsupervised learning algorithm is used to label them.
- Use the data preprocessing method to deal with imbalanced data sets and outliers. The best features are obtained by feature selection, the machine learning model is established, the cross validation is used for training, and the grid search is used to obtain the best model parameters.
- Use transfer learning, the trained model is transferred to the target domain, and the data of the target domain is used as the test set to obtain the prediction results. Retrain in the target domain, compare the training results with the results of transfer learning, and evaluate the effect of transfer learning.

The source domain dataset of this work is downloaded from EDP open data and the target domain is collected from LDT wind farm.

1.2 Thesis Outline

This thesis is structured as follows:

- Chapter 2 indicates the background and related knowledge of this experiment including the concept of fault detection, data preprocessing and transfer learning.
- Chapter 3 indicates the methods used in data preprocessing and model training.
- Chapter 4 indicates the description and preprocessing of 2 datasets which have been used in this work.
- Chapter 5 indicates the experiments and represents the results including the training on different datasets and the results of transfer learning.
- Chapter 6 indicates the conclusion and distribution of this work, and also carry out the self reflection.

Chapter 2

Related Work

This chapter will summarize some research related to the subject of this paper, focusing on the data-driven fault prediction method based on SCADA data. Data preprocessing can effectively extract the relevant information from the original dataset. Through the statistical or machine learning algorithms, researchers can clearly explore the relationship between input and output and find the complex behavior pattern of the system.

Section 2.1 introduces a simple and intuitive method to detect faults through signal trend analysis; Section 2.2 introduces the prediction method of normal behavior modeling of wind turbine using regression model; Section 2.3 introduces the method used in this experiment to detect faults by classifying normal and abnormal samples of SCADA data; Section 2.4 introduces other methods of predictive maintenance using machine learning; Section 2.5 indicates the principle of transfer learning; Section 2.6 summarizes this chapter and explains why ANN and AdaBoost models are used in this experiment.

2.1 Signal Trend Analysis

The original artificial intelligence data-driven algorithm is the signal trend analysis method. It usually involves comparing the data over a period of time with the historical data of the same turbine or the real-time data of another similar wind turbine, observing the ratio relationship between different parameters, and recommending a statistical method to judge the fault.

In [14], the authors showed an example of the signal trend method. They recorded the normalized temperature changes of the driven train system components in the test turbine and the other four other identical turbines in a time series. The average temperature of all components in the wind farm is set at 100%, and the temperature is recorded every ten minutes. Compared with the control turbines, it can be seen that the component temperature deviation of the test turbine increases significantly, which also corresponds to the increase in component damage and failure risk. In [15], the authors used the SCADA data from the Controls Advanced Research Turbine at National Wind Technology Center. The original data is split according to different motion states, which can effectively distinguish the fault characteristics from similar operational features. In order to obtain more feature data, the author reduces the sampling rate to once per minute. Then, by observing the changes in SCADA data before and after the known transmission fault, the most sensitive features to the fault are extracted.

The changing trend of parameters in SCADA can directly reflect the development of wind turbine fault, but sometimes the change of parameters can not represent the fault because the change in wind turbine operating conditions and the influence of the surrounding environment will also change the values of these parameters. In particular, the change in temperature has a high degree of uncertainty, which needs to be explained manually most of the time. If the fault is judged only by setting the threshold and simple calculation, it will lead to more false alarms and alarm omissions.

2.2 Regression Model

The normal behavior regression model was developed at a stage when wind turbine components were considered healthy. Unlike the signal trend analysis introduced in Section 2.1, which interprets faults by observing trend changes, the normal behavior regression model puts SCADA data into the machine learning regression model for training. Assuming that the components of the wind turbine are healthy, some independent features are used as input variables to predict the relevant output variables. For example, many manufacturers will test the relationship between wind speed and power generation and provide power curves under idealized conditions. However, due to the influence of other environmental factors (such as temperature, altitude, wind direction, etc.), the power curve can not represent the actual situation. In [16] the author first established a simple linear regression model with wind speed and output power as input and output, and then made the model fit better by taking ambient temperature and wind direction as input variables.

For SCADA data with few features and obvious linear relationships, linear regression can achieve the fault prediction effect under the condition of low time cost. In [12], the authors found that the abnormal bearing temperature was the main cause of WT failure, so they took the bearing temperature as the label value of the regression model. Then, by observing the data, the author finds that there is a high correlation between stator temperature and bearing temperature. Taking the stator temperature as the original input, and adding four characteristic inputs of retaining power output, engine room temperature, and shaft speed to reduce the prediction error, a multiple linear regression model is established. However, the change in the environment around the temperature sensor will lead to the fluctuation of the signal value, resulting in more noise in the data set and reducing the prediction effect. Finally, the model can predict the bearing temperature in the range of $\pm$ 5 $^\circ$ C.

When linear regression cannot fit the data well, polynomial regression can be used. In polynomial regression, increasing higher characteristic power (such as square term or cubic term) is also equivalent to increasing the degree of freedom of the model to capture the nonlinear changes in the data. For example, if $\hat{h}$ is the predicted value, x is the only one feature the polynomial regression equation is:

$$\hat{h} = \theta_0 + \theta_1 x + \theta_2 x^2 + ... + \theta_{n-1} x^{n-1} + \theta_n x^n \tag{2.1}$$

Wilkinson *et al.* [14] developed polynomial models for predicting gearbox and bearing temperatures under different SCADA inputs. The authors used SCADA data from 472 wind turbines from five different wind farms in Europe. Through the modeling of bearing or gearbox temperature, rotor speed, power output, and engine room temperature, this paper introduces the comparative experiment carried out on two different turbines in the same position and two different turbines in different positions. The purpose of the experiment is to detect the faults of the gearbox and main bearing. In the final test set, 24 of the 36 component failures were detected, and only three false alarms. It can be seen that the polynomial model can greatly reduce the false alarm rate, but there is a high probability of omission. This is because the learning ability of the model is limited and some fault information is judged to be normal. Therefore, for some datasets with complex nonlinear relationships between variables, more complex machine learning models need to be used.

Feedforward neural network is one of the most common artificial neural networks. It can determine the nonlinear relationship between the input value and output value through training. In [17], the author applies a feedforward neural network to the fault prediction of bearing temperature after driving the main shaft of the wind turbine. Based on about one year's data of two 3 MW Turbines on the farm, using output power, engine room temperature and turbine speed as inputs, the failure of the first turbine was detected three months in advance and verified under the normal operation of the second turbine. In [12], the author establishes a feedforward model to predict the bearing temperature in wind turbine gearbox. The four input characteristics are stator temperature, residual power output, nacelle temperature and shaft speed. However, the model shows large prediction error under transient conditions. When more input signals are provided to the network and more hidden units (neurons) are used, the network can better adapt to transient conditions. This means that when using feedforward

neural network for regression prediction, we will encounter the situation of increasing model parameters to reduce the prediction error, but this may cause over fitting.

Different from the traditional feedforward neural network in which all neurons are fully connected, the convolution neuron network (CNN) has unique convolution layers and pooling layers. The convolution layer is a set of parallel feature maps, which is composed of convolution kernels sliding in a certain stride on the input image and performing certain operations. The pooling layer can reduce the size of data space and use a value (such as maximum value and average value) to represent the characteristics of data in an area. The essence of CNN is multiple filters to extract spatial features hidden in data. Through the training process, the convolution layer of CNN is optimized to extract high discrimination features, and the latter layer imitates the multi-layer perceptron performing classification [13].

In traditional neural networks, all samples are independent by default, and the relationship of data in time will be ignored. For many time series data problems, such as state detection and trend prediction, the recurrent neural network (RNN) [18] can transfer the data information of the previous time to the neurons of the next layer, so that the previous historical data can be considered in the training of the model. Figure 2.1 shows the simple structure of the RNN model. It can be found that the principle of RNN is to update a hidden state h_n iteratively, which is calculated by x_n and h_{n-1}, and the output y_n is calculated by h_n through the activation function. This structure gives RNN a strong ability to deal with sequence problems.Long short-term memory (LSTM) [19] is a special recurrent neural network. By introducing a storage unit to store historical data and controlling the information flow by using input gate, forgetting gate and output gate, LSTM can remember longer sequence information than RNN, which alleviates the problem of gradient disappearance or explosion to a certain extent.

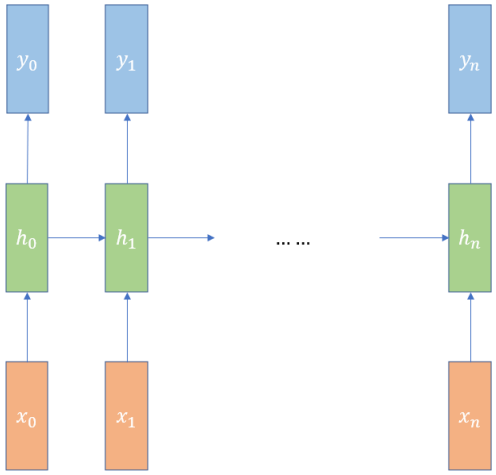

Figure 2.1: an Example Structure of RNN Model

Xiang *et al.* [13] proposed a method for fault detection of wind turbine based on cascaded deep learning network of CNN and LSTM based on AM. CNN extracts the spatial features of the original data and takes the extracted spatial features as the input of LSTM network. The temporal feature is extracted through LSTM, and the result is input into am layer. Am can simulate the attention distribution mechanism of human brain on the data results, and calculate the correlation between input and output and the feature distribution with large weight. It enhances the influence of important features and improves the accuracy of feature selection.

2.3 Classification Model

In the problem of wind turbine fault detection, researchers often set normal or abnormal labels for samples according to the description of the operation state in SCADA data. Then, the data set is divided into the training set and the test set. Through feature engineering, the features with high

correlation are selected as the input, and the label value is used as the prediction output to form the final machine learning classification model. The model can predict the operation state of the wind turbines in real-time through SCADA data. Although it can not predict the fault for a long time in advance, it can detect and repair it in time when the fault is about to occur or just begins to occur.

In [20], the author collected SCADA data of 27 wind turbines in three months to predict blade pitch failure. Data mining algorithms such as artificial neural network, k-nearest neighbor, partial decision tree and bagging were applied to evaluate the quality of blade fault prediction model. Finally, genetic algorithm achieved the best effect, reaching 66.9% specificity and 71.2% recall rate. In [21], the author proposed a new hybrid model combining LSTM and XGBoost. LSTM is mainly used for accurate classification problems, and XGBoost decision tree is used for transparent output. Finally, the model achieved 97% classification accuracy in the test set. The model also analyzed the feature importance of the causes of anomalies and accurately identified the components causing anomalies in the wind turbine. In [22], the author uses a support vector machine for fault prediction. The data set of this study is divided into operation data, status data, and warning data. The operating data includes information about the performance of the wind turbine, and the status data includes fault and warning information. Fault samples can be separated from normal operation samples through status data, and then the author classifies them according to specific fault types, including generator overheating, power failure, air cooling, etc. Finally, the model accurately predicted all generator overheating faults, but there may be data leakage, which makes the prediction too optimistic.

2.4 Other Approaches

Chen *et al.* [23] developed a machine learning diagnosis model based on Gaussian processes (GP) and Principal Component Analysis (PCA). This work inputs SCADA data into the constructed GP prognosis model and uses the PCA model to determine which components are defective. PCA uses an orthogonal transformation to linearly transform the observed values of a series of possibly related variables, so as to project them into the values of a series of linearly uncorrelated variables, which plays a good role in dimension reduction.

Zhang and Kusiak [24] collected SCADA vibration data (including drive train and tower acceleration) every 10 seconds and developed a monitoring model using a k-means clustering algorithm. K-means algorithm groups the data by the similarity of vibration data. According to the error report of the WT, mark the cluster as a normal or abnormal state of WT vibration. Abnormal data points can be detected by comparing the abnormal data points with the data points of two types of clustering. Combining vibration data with CM is an interesting idea, but detecting the root cause of wind turbine vibration needs further research and higher frequency data.

Qiu *et al.* [25] provided a study on SCADA alarm. It investigates the alarm data of two wind farms in two years and proposed an alarm analysis method based on time series and probability. These methods can rationalize and reduce the alarm data, and carry out fault detection, diagnosis and prediction according to the alarm conditions. This is an innovative state detection research method. If this method is used in the alarm system of wind turbine, it is likely to give early warning of faults in advance, reduce the shutdown time of wind turbine and improve the maintenance efficiency.

2.5 Transfer Learning

Generally speaking, transfer learning is to using existing knowledge to learn new knowledge [26]. The core is to find the similarity between existing knowledge and new knowledge. In some fields, on the one hand, the cost of learning from scratch is too high. For example, in some large SCADA data sets, such as [27], the data is often measured at an interval of one second, which will produce millions of data samples in a month, which undoubtedly requires huge space and time to build the model. On

the other hand, some newly-built or frequently shut-down wind farms, lack SCADA historical data for training, which makes it very difficult for the predictive maintenance of wind farms. Therefore, it turns to using the existing relevant knowledge to assist in learning new knowledge as soon as possible. The existing knowledge is called the source domain, and the new knowledge to be learned is called the target domain. The traditional machine learning model is not flexible enough and the results are not good enough when dealing with the tasks such as the distribution and dimension of data and the change of model output, while the transfer learning relaxes these assumptions. Under the conditions of data distribution, feature dimension, and model output change, the knowledge in the source domain is used to better model the target domain [28]. In addition, in the absence of labeled data, transfer learning can make good use of the labeled data in related fields to complete the labeling of data.

In [21], The author combines the recursive neural network method for classification with the XGBoost decision tree classifier for transparent output, realizes the knowledge transfer from offshore wind turbine to onshore wind turbine, and achieves 65% classification accuracy in the target domain. In [29], due to the difficulty in obtaining the fault gearbox signal data of wind turbine under some working conditions, it is impossible to diagnose the health condition. According to the characteristics of wind turbine vibration signal, the author realizes the knowledge transfer under different working environments through linear transformation matrix.

The transfer learning realized in this work is feature-based transfer, which realizes knowledge transfer by mapping the source domain and target domain to the same feature space. Figure 2.2 shows a sample learning process of transfer learning.

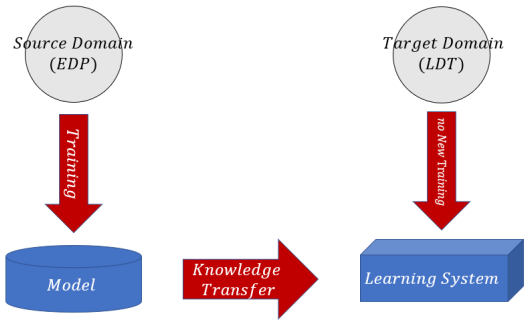

Figure 2.2: Knowledge Transfer from Source Domain to Target Domain

2.6 Summary

In some cases, the signal trend method can achieve good prediction results at a very small cost, but due to its singleness, it does not have universal applicability and variability. There is no way to change the interference factors accordingly, and it is difficult to change the false alarm problem. The basic principle of establishing the regression model according to normal behavior is to construct machine learning model training data, determine the threshold according to the residual between the predicted value and the actual value, and test the performance of the test set according to the threshold. Most fault prediction regression models take the parameters in SCADA data (such as output power, bearing temperature, etc.) as the predicted value, and judge the fault of the component by detecting the abnormality of a parameter. This method can make the fault accurate to specific components, but in some cases, the failure of some core components will lead to the shutdown of the whole wind turbine. Therefore, if you want to predict the overall failure of the wind turbine, the maintainer is required to build a model for each core component, which often requires a lot of time and resources, which shows that in some cases, the regression model can not predict the failure of the wind turbine. Compared with the regression model, building a classification model for the overall SCADA data can be applied to almost all fault detection problems. This is because building the classification model needs to mark the operating state of the whole turbine, and all training is based on the whole data

set. However, some SCADA data sets do not record the fault problems, and the samples need to be classified by a data mining algorithm, which may be different from the real situation.

The focus of this research is to use feedforward neural network and AdaBoost algorithms to detect the anomaly of SCADA data and apply the transfer learning task between to different SCADA datasets. Feedforward neural network has a strong nonlinear fitting ability when dealing with large datasets. It can map any complex nonlinear relationship, and its learning rules are simple and easy to be realized by the computer. Compared with RNN, it does not need to construct time series, and the parameters during training are far less than RNN. The model is relatively simple, which is also the reason why this experiment uses the feedforward neural network. AdaBoost has a simple structure and high classification accuracy as a binary classifier, and it is not easy to be affected by imbalanced datasets. Feedforward neural network and AdaBoost will be described in more detail in Sections 3.1 and 3.2.

The specific experimental framework is as follows:

1. The source domain dataset is from EDP Open data and the target domain dataset is collected from Levenmouth Demonstration Turbine.
2. Use the alarm logs to label the source domain dataset and use isolation forest to label the target domain dataset.
3. Apply feature selection to the source domain and find the similar features between source domain and target domain.
4. Train different machine learning models in the source domain and save the one with best testing results.
5. Transfer the knowledge to target domain and obtain the evaluation results.
6. Retrain the machine learning model on target domain and compare the results with the results in Step 5.

Chapter 3

Methodology

This chapter indicates the methodologies which have been applied in this work. Section 3.1 introduces the process of data preprocessing including data normalization, data labeling, data resampling and feature selection. Section 3.2 explains the principle of machine learning techniques in the experiments like isolation forest and artificial neural network.

3.1 Data Preprocessing

3.1.1 Data Normalization

In the field of machine learning, different features often have different dimensions and units, which will affect the results of data analysis. Data normalization refers to scaling the data to make it fall into a small specific interval. It can remove the unit limit of data and convert it into dimensionless pure value, so that indicators of different units or orders of magnitude can be compared and weighted. Mainstream normalization technologies include *Min-Max* normalization, *Z-score* normalization and *Mean* normalization. This experiment adopts *Z-score* normalization, which can scale all the data in a normal distribution with $\mu = 0$ and $\sigma = 1$. The specific formula is as follows:

$$x' = \frac{x - \mu}{\sigma} \tag{3.1}$$

where μ is the overall average value and σ represents the standard deviation of whole dataset.

3.1.2 Imbalanced Dataset

When some experiments are carried out to detect rare samples (such as anomaly detection and fraud identification), highly imbalanced datasets will appear. The imbalance of sample classification will make the model more inclined to learn the majority of samples, so that the actual prediction will focus on the majority of categories, which will seriously affect the robustness of the model. Imbalanced data sets are very common in the field of wind turbine faults, because wind turbines are in normal operation most of the time, so the number of fault samples is bound to be much less than that of normal samples.

Nowadays, the data method mainly used to deal with unbalanced datasets is resampling, including oversampling and undersampling. Oversampling will increase the number of minority samples in the training set. The common oversampling methods are SMOTE [30] and ADASYN [31]. In [21], the author uses SMOTE to expand 3518 abnormal samples in the LDT data set to 13594, making it equal to the number of normal samples. Undersampling aims to reduce the number of majority samples to balance the class distribution. The commonly used methods are random undersampling and NearMiss. On the other hand, by adjusting the class weight, the model can pay more attention to the minority class when training, which is also used to deal with the imbalanced sample distribution. In [32], the author applies class weight by modifying the objective function to make the model pay more attention to a few classes, so as to learn equally from all classes. The method of resampling can improve the training effect in many machine learning models like ANN and SVM.

SMOTE

SMOTE [30], also known as Synthetic Minority Over-sampling Technique, is a mainstream oversampling technology. Based on the k nearest neighbor sample points of each minority sample point, it randomly selects N adjacent points for to calculate the Euclidean distance, and multiplies the distance by a threshold δ in the range of 0 and 1, so as to achieve the purpose of synthesizing data. The core of this algorithm is that the features of adjacent points in feature space are similar. It does not sample in the total data space, but in a separate feature space, so its accuracy will be higher than the traditional sampling method. However, if the selected minority samples are surrounded by minority samples, the newly synthesized samples will not provide much useful information; If the selected minority samples are surrounded by majority samples, which may be noise, the newly synthesized samples will largely overlap with the surrounding majority samples, making classification difficult. Therefore, some under sampling methods are often used to eliminate overlapping data and useless information.

Random Undersampling

Random undersampling is to randomly select some samples from majority classes and eliminate them. The disadvantage of this method is that the eliminated samples may contain some important information, resulting in the poor effect of the learned model.

ENN

ENN (Edited Nearest Neighbor) [33] achieves the purpose of under sampling by cleaning the overlapping data. For a sample belonging to the majority class, if more than half of its K nearest neighbors do not belong to the majority class, the sample will be eliminated. Another variant of this method is that if all k nearest neighbors do not belong to most classes, the sample will be eliminated.

3.1.3 Feature Selection

Because the original operation data contains many features, but not all of them will have an important impact on the training of the model, some irrelevant features may lead to overfitting and increase the calculation cost. The essence of feature selection is to measure the excellence of a given feature subset through a specific evaluation standard. Through feature selection, redundant features and irrelevant features are removed, while useful features are retained. It can enhance the generalization ability of the model and reduce the consumption of computing costs. This advantage is particularly obvious when training some more complex models, such as SVM, neural networks, and so on. The three main feature selection methods are: filter, wrapper, and embedded [6]. The filter refers to preprocessing the features and filtering out some features that are not beneficial to the target in advance. The filtering method of feature selection can well extract relevant features from SCADA data and save computing cost. It only considers the task target and has nothing to do with the model. In [21], the author used principal component analysis to extract the factor weights of the first 20 features that retain 95% of the variance of the original data, and put these features into the model as inputs. In [34] the author uses univariate parameter selection technology to reduce the number of input parameters, which can be realized through the *Select-K-Best* class provided by *Sklearn* library. Finally, the author obtained 25 features with the highest mutual information score with the target variable.

In this experiment, the Pearson correlation coefficient [35] is used to filter the features which can detect the degree of linear correlation between two continuous variables. The coefficient value is between -1 and 1. Positive value indicates positive correlation, negative value indicates negative correlation, and the greater the absolute value, the higher the degree of linear correlation. The Pearson correlation coefficient between two variables is defined as the quotient of covariance and standard deviation

between two variables:

$$\rho(X,Y) = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sqrt{\sum_{i=1}^{n}(X_i-\mu_X)^2}\sqrt{\sum_{i=1}^{n}(Y_i-\mu_Y)^2}} \tag{3.2}$$

3.2 Machine Learning Models

3.2.1 Isolation Forest

Isolation forest was proposed by Liu, Ting and Zhou [36] in 2008 and is widely used in anomaly detection of structured data in industry. The basic principle of isolated forest is that abnormal samples can be isolated by less random feature segmentation than ordinary samples. The process of isolation forest is consisted of these steps:

1. Randomly select n points from the dataset as sub samples and put them into the root node of an isolated tree.
2. Randomly specify a dimension and randomly generate a cutting point p within the data range of the current node.
3. The selection of this cutting point generates a hyperplane, which divides the current node data space into two sub spaces: put the points less than p in the currently selected dimension on the left branch of the current node, and the points greater than or equal to p on the right branch of the current node.
4. Iterate steps 2 and 3 are performed on the left branch and right branch nodes of the node, and new leaf nodes are constructed continuously until there is only one data on the leaf node or the tree has grown to the set height.

3.2.2 Artificial Neural Network

Artificial neural network is a computational model inspired by human brain. It is a complex network structure formed by the interconnection of a large number of processing units (neurons). It can change the internal structure on the basis of external information and achieve the effect of learning from experience. In this work, one of the most common type of ANN - Feedforward Neural Network (FNN) is used.

Structure

Figure 3.2 shows a feedforward neural network structure, it has an input layer, a hidden layer and an output layer. Each layer consists of a different number of neurons, which are provided by all inputs or other neuronal outputs of the previous layer. In addition to input and output, FNN also includes adjustable weights w, bias terms b and activation functions g.

On each neuron of the neural network, $x_1, x_2, ..., x_n$ stands for input, $w_1, w_2, ..., w_n$ represents the weight corresponding to each input, b represents the bias on this neuron, and a is the neuron output value after processed by activation function. The output values of neurons can also been regarded as the input values of the next layer. This is the idea of forward propagation and the specific formula on one neuron is as follows:

$$z = \sum_{i=1}^{n} w_i x_i + b, a = g(z) \tag{3.3}$$

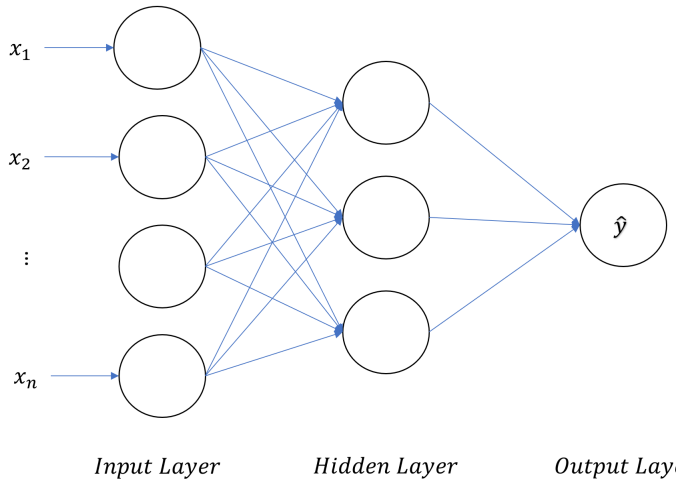

Figure 3.1: An Example of Feedforward NN

Back Propagation

Back propagation is a training method of artificial neural network. It calculates the gradient of the loss function to each parameter through the derivative chain rule, and updates the parameters according to the gradient. A common back propagation contains several steps:

1. Calculate the overall error between predicted values and actual values by using the loss function. In this binary classification work, the binary cross entropy function is used:

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i) \tag{3.4}$$

   N is the output size, $\hat{y}_i$ means the i-th value in the predicted value, y_i is the corresponding actual value.

2. Back propagate from the output layer to the hidden layer. For example, according to Figure **??**, the influence of weight w_{11} on the overall error can be obtained through chain calculation:

$$\frac{\partial E_{total}}{\partial w_{11}} = \frac{\partial E_{total}}{\partial a_{21}} \times \frac{\partial a_{21}}{\partial z_{21}} \times \frac{\partial z_{21}}{\partial w_{11}} \tag{3.5}$$

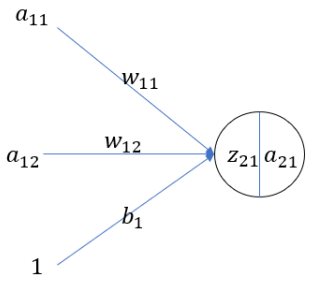

Figure 3.2: A neuron in the output layer

3. Back propagate between the hidden layers. It is different from the second step, because neurons

in the hidden layer will receive several error values from the latter hidden layer. For an instance, Figure 3.3 shows a slice of hidden layers, the partial derivative of weight w_{01} can be calculated by:

$$\frac{\partial E_{total}}{\partial w_{01}} = \frac{\partial E_{total}}{\partial a_{11}} \times \frac{\partial a_{11}}{\partial z_{11}} \times \frac{\partial z_{11}}{\partial w_{01}} \tag{3.6}$$

while:

$$\frac{\partial E_{total}}{\partial a_{11}} = \frac{\partial E_{21}}{\partial a_{11}} + \frac{\partial E_{22}}{\partial a_{11}} \tag{3.7}$$

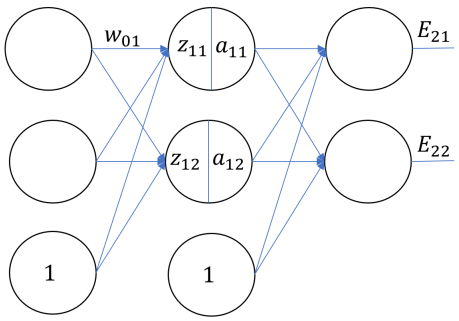

Figure 3.3: A slice of hidden layers in FNN

4. The last step is to update the weights using the gradient descent method. Similarly, select the example of w_{01} in the Figure 3.3. The updated weight w_{01}^{+} can be obtained according to the following formula:

$$w_{01}^{+} = w_{01} - \alpha \frac{\partial E_{total}}{\partial w_{01}} \tag{3.8}$$

where α is the learning rate used to control the step of gradient descent. After iterate the forward and back propagation several times, the model will stop training until the loss is sufficiently small or cannot be reduced, or the condition of early stopping is met.

Activation Function

Activation function is the nonlinear part of neuron, which is used to solve the defect of insufficient expression ability of linear model. The neural network without activation function is essentially a linear regression model, which can only solve the problem of linear separability. The activation functions selected in this experiment are Sigmoid and ReLU.

Sigmoid: The Sigmoid function can be expressed as:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

The value range of the Sigmoid function is (0,1), It is monotonic and continuous. In this work, it is used for the binary output layer. After being processed by the Sigmoid function, all outputs will be between 0 and 1. It is easy to set the threshold between 0 and 1 to classify each sample. However, it should be noted that when input x is too large or too small, the slope of Sigmoid function tends to 0, so it is easy to cause the problem of gradient disappearance in back propagation. In addition, the output value of Sigmoid function is not centered on zero, which will lead to slow convergence of the model. Therefore, this experiment does not use sigmoid function as the activation function of hidden layer.

ReLU: The ReLU function can be expressed as:

$$ReLU(x) = \begin{cases} 0, x < 0 \\ \\ x, x \geq 0 \end{cases} \tag{3.10}$$

Compared with the Sigmoid function, the calculation of ReLU is simpler and more efficient, and there is no exponential operation. In addition, ReLU is not saturated in the positive interval, which solves the problem of gradient disappearance. In this experiment, ReLU is used as the activation function of the hidden layer.

3.2.3 Decision Tree

Decision tree is a kind of tree structure, in which each internal node represents a test on an attribute, each branch represents the test output, and each leaf node represents a class. Decision tree is a very common classification method and also a supervised learning method. The core of decision tree is to find decisive features to divide the best results.

Entropy can be used to express the degree of data confusion. The more ordered or concentrated the data is, the lower the entropy is; The more chaotic or dispersed the system, the higher the entropy. For example, if the classification of event S is $(s_1, s_2, ..., s_n)$ and the probability of occurrence of each class is $(p_1, p_2, ..., p_n)$, the entropy can be calculated as follows:

$$Entropy(S) = -\sum_{k=1}^{n} p_n \log_2 p_n \tag{3.11}$$

Information gain is the difference of entropy before and after a feature is divided into data sets. It represents the reduction of information complexity under a feature division. ID3 algorithm [37] uses information gain to select features, and those with large information gain are preferred. In C4.5 algorithm[38], the information gain ratio is used to select features.

CART classification [39] tree algorithm uses Gini coefficient instead of information gain. Gini coefficient represents the purity of the model. The smaller the Gini coefficient, the lower the purity and the better the characteristics. Specifically, in the classification problem, assuming that event S has n classes and the probability of the each class is $(p_1, p_2, ..., p_n)$, the expression of Gini coefficient is:

$$Gini(S) = \sum_{k=1}^{n} p_n(1 - p_n) \tag{3.12}$$

The basic steps of decision tree construction are as follows:

1. Treat all features in the dataset as nodes.
2. Take each node as the segmentation point and find the optimal segmentation point.
3. Split the node into two nodes N_l and N_r.
4. Repeat the step 2 and 3 on N_l and N_r until stop condition is met.

In this experiment, decision tree is chosen as the weak classifier of AdaBoost.

3.2.4 AdaBoost

AdaBoost [40] is a type of ensemble model, which is based on the idea of boosting algorithm. Its core idea is to train different classifiers (weak classifiers, such as decision tree) for the same training set, and then combine these weak classifiers to form a stronger final classifier (strong classifier). AdaBoost algorithm is realized by adaptive weight in each round of boosting. According to the classification results of each training, it gives more weight to the misclassified samples, sends the new data set with modified weight to the next classifier for training, and finally integrates the classifiers obtained each time as the final decision classifier. The specific steps of AdaBoost are as follows [41]:

1. Given the training set S, set the number of iteration as M, the total number of samples as N. Initialize the weight distribution: $D_1 = (w_{1,1}, w_{1,2}, ...w_{1,i}), w_{1,i} = \frac{1}{N}, i = 1, 2, ..., N$.
2. In iteration $m = 1, 2, 3..., M$:
   (a) Train the dataset with weight distribution D_m and obtain the weak classifier $G_m(x)$.
   (b) Calculate the classification error rate of $G_m(x)$ on the training data set.
   (c) Calculate the weight of $G_m(x)$ in the strong classifier.
   (d) Update the weight distribution of the training data set.
3. Obtain the final strong classifier.

3.2.5 Evaluation Criteria

The test results on the test set will be evaluated according to the following metrics: precision, recall and F1 score. These metrics are calculated according to the confusion matrix, which is the summary of the prediction results of classification problems, as shown in Table 3.1. The abbreviations in the confusion matrix represent the following contents:

- True Positive (TP): Number of positive samples correctly classified.
- False Positive (FP): Number of negative samples misclassified as positive.
- True Negative (TN): Number of negatives samples correctly classified.
- False Negative (FN): Number of positive samples misclassified as negative.

		Predicted Class	
		Positive	**Negative**
Actual Class	**Positive**	TP	FN
	Negative	FP	TN

Table 3.1: Confusion Matrix

The precision, recall and F1 score for positive class can be calculated according to following formula:

$$recall = \frac{TP}{TP + FN} \tag{3.13}$$

$$precision = \frac{TP}{TP + FP} \tag{3.14}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{3.15}$$

The precision represents the proportion of actually positive samples among all samples with positive prediction; Recall refers to the proportion of positive samples successfully predicted by the model among all positive samples. The recall is only related to the real positive samples, not to the real negative samples; In fault detection, positive samples represent fault samples, and the cost loss of predicting fault samples as normal samples is very high. Therefore, the most important thing is to maximize the recall rate, while f1score represents the harmonic average of precision and recall.

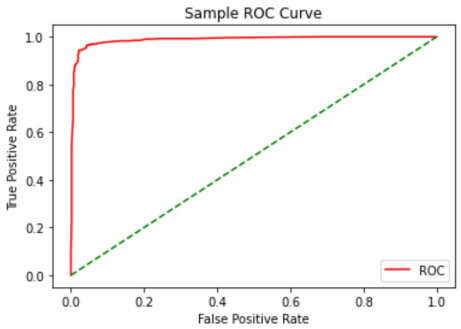

Figure 3.4: Sample ROC Curve

In this experiment, the AUC(Area Under Curve) value of the ROC curve is also used to evaluate the model. Figure 3.4 shows a sample ROC curve. The y-axis coordinate of the ROC curve, true positive rate (TPR), is numerically equal to the recall of positive class. The x-axis coordinate false positive rate (FPR) is numerically equal to 1 minus the recall of the negative class:

$$TPR = \frac{TP}{TP + FN} \tag{3.16}$$

$$FPR = \frac{FP}{FP + TN} = 1 - \frac{TN}{FP + TN} \tag{3.17}$$

Through the classification threshold, θ (between 0 and 1, the default is 0.5) take values, in turn, we can get many groups of TPR and FPR values, and draw them in the image in turn to get a ROC curve. The better the model is, the closer the ROC curve is to the upper left corner (0,1) on the image, and the larger the area (AUC value) surrounded by the horizontal axis and straight-line FPR = 1 under the ROC curve.

Chapter 4

Data Description and Preprocessing

4.1 EDP Open Data - Wind Turbine Failure Detection

The data used as source domain is downloaded from the EDP open data[42]. During 2016 and 2017, the company had obtained the SCADA data from 4 offshore wind turbines. In this experiment, the wind turbine with serial number T07 will be selected for research.

The Operational dataset contains SCADA data recorded by wind turbines at 10-mins intervals and has 82 features. The feature values recorded in this data set include environmental conditions, such as wind speed and temperature, the operation status of some components of wind turbine, such as the rotation speed of generator, and the overall performance of wind turbine, such as power generation capacity and power generation frequency.

The company also provides an alarm log dataset which records the changes during the operation.

- *TimeDetected* indicates the time when the alarm was detected.
- *TimeReset* indicates the time when the turbine was reset after detecting the alarm.
- *Remarks* indicates the text of the alarm log.

This information plays an important role in labeling the data in the operation dataset.

4.1.1 Data Labeling

The main objective of this experiment is to train machine learning models to deal with the classification problem of fault detection in wind turbines. It is important to properly label the samples. In this work, all the data has been divided into "faulty" and "normal". Based on this, a new feature *isFaulty* is created as the label. When the sample is labeled as 0, it means that it is in normal operation during this period of time; When sample is marked as 1, it means that it encounter faults in this period of time.

The marking of the faulty labels in the wind turbine operation dataset benefits from the recording of abnormal operation of the wind turbine in the alarm log dataset. Comparing the *Remarks* recorded in the log with the official documents[43] provided by Vestas, the wind turbine manufacturer, can easily obtain the event code and category corresponding to each event, so as to determine the faults. However, not all alarm logs mean the generation of errors, some artificial operation information and automatically generated output information such as "Generator 1 in" and "External power ref: 2000kW" will not be recognized as errors. Finally, the author collected 54 types of errors from the log dataset which are shown in Table 4.2. The *code* records the event code corresponding to the event, *Category* indicates the reason for the error, *Text* corresponds to the *Remarks* in the log dataset.

According to the *TimeDetected* and *TimeReset* of these errors in the log dataset, the author marks the sample in the corresponding time period in the running dataset as "faulty". For example, if error

443 is detected at 18:34 and the wind turbine is reset at 18:52 after the error is found, the operation data between 18:30 and 19:00 will be marked as “faulty” (the operation dataset is recorded every 10 minutes). For logs that do not record the reset time, the time period in which the error was detected is marked as “faulty”.

Fig 4.1 shows the relationship between power generation and wind speed of the wind turbine after labeling all the origin data. The purple points indicate the normal samples and the yellow ones indicate the samples labeled as “faulty”. In the wind turbine, there is a direct linear relationship between wind speed and power generation. In the figure, most abnormal values (such as $Power\_wh = 0$) are divided as abnormal values, which shows the correctness of the data labeling.

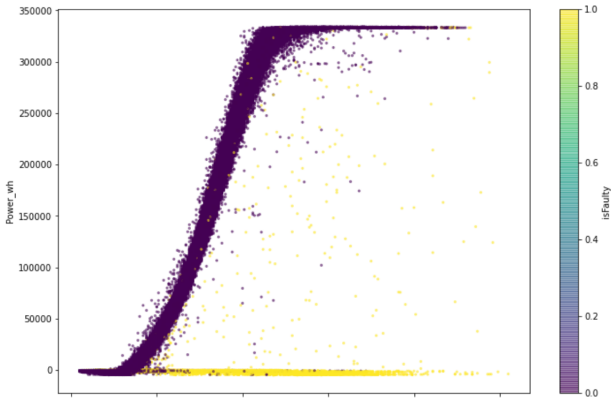

Figure 4.1: EDP Power Generation versus Wind Speed

4.1.2 Data Resampling

After completing the data labeling, a total of 100895 normal operation samples and 3843 faulty samples were obtained in this experiment. This is a reasonable distribution for a wind turbine operating for a long time, but it causes a serious imbalance in the number of the two classes in the dataset, which is not optimistic for the training of the model. Therefore, it is necessary to apply re-sampling methods, so as to balance the overall class distribution.

In this experiment, a mainstream oversampling and undersampling method: SMOTE and random undersampling, and a combination of oversampling and undersampling named ”SMOTE + ENN” are selected to compare with the original unbalanced training set data without sampling to explore the effects of different sampling methods..

The *SMOTE* function in Python *imblearn* library is used to oversample 80% of the data in the original dataset, and the remaining 20% is used as the test set of this experiment, which keeps the proportion of normal and abnormal samples unchanged. When applying SMOTE, it is necessary to use *Z-score* to normalize to data first.This is because the Euclidean distance between samples needs to be calculated when applying SMOTE. The dataset put into SMOTE model contains 3057 faulty samples and 80733 normal samples. Therefore, the training set after sampling has 80733 abnormal samples and normal samples.

Python *imblearn* library also provides the function called *RandomUnderSampler* to achieve the random undersampling. After applying this function to the training set, it selects 3057 abnormal samples and normal samples. Moreover, the combination of SMOTE and ENN can also be implemented by the *SMOTEENN* function in the *imblearn* library. And the final training set contains 76604 normal samples and 79037 abnormal samples. In addition, the testing set contains 20162 normal samples and 786 faulty samples.

4.1.3 Feature Selection

The correlation matrix is constructed to quantify variable dependencies. Pearson correlation coefficient evaluates the linear relationship between continuous variables. If the correlation coefficient between variables is greater than 0.95, it indicates that there is an obvious linear relationship between the two variables, and only the one with larger correlation coefficient is retained. Fig 4.2 shows a small part of heat map based on the correlation coefficient of feature variables. It is obviously that the average temperature in the three generator phases has a high correlation, so the experiment will only use one of them as the feature required for training. According to this principle, the number of feature variables is reduced to 52.

Then, among the 52 features filtered, 34 features with high similarity to these features in the target domain are found in this experiment, such as *Rtr_RPM_Avg* and *RotorSpeed_rpm*, *Blds_PitchAngle_Avg* and *Pitch_Deg*, and they are used as the feature variables needed for the final training model.

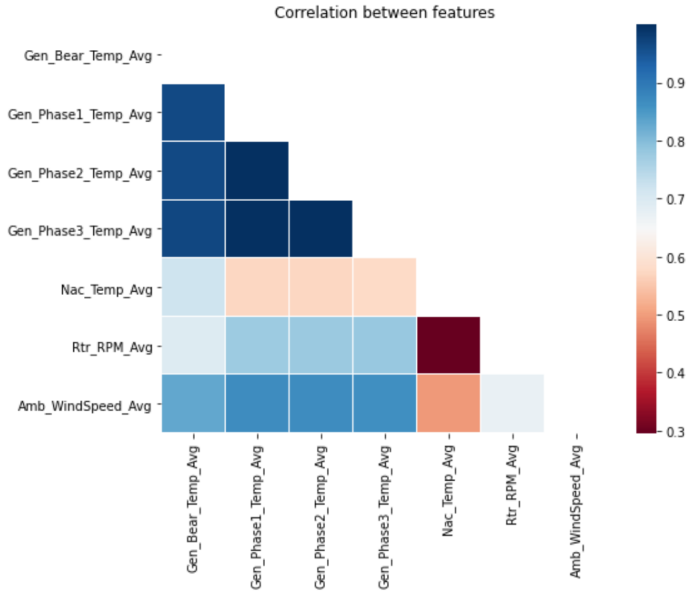

Figure 4.2: Part of the Heat Map of Correlation Matrix

4.2 Levenmouth Demonstration Turbine Dataset

The data as the target domain is selected from the SCADA data of the 7 MW offshore wind turbine Levenmouth Demonstration Turbine (LDT) [27] in Scotland in the first half of 2017. This SCADA data is measured by different types of sensors at the frequency of 1Hz, including wind speed, temperature, voltage and other variables. For transfer learning, the target domain uses the knowledge obtained from the source domain for learning. Therefore, there is no need for additional training in the target domain. The parameters of the model in target domain should be inherited from the source domain, the parameters input by the model should correspond to the source domain, and all data in the target domain should be treated as training sets.

Since the data in the target domain is measured at an interval of 1 sec, while the data in the source domain is measured at an interval of 10 mins, it is very necessary to process the data uniformly. After eliminating the outliers in the target dataset, this experiment uses the *groupby* function in *Pandas* library and calculates the maximum value, minimum value, average value, standard deviation of SCADA data in 10 mins based on the *TimeStamp* attribute in LDT dataset. For some feature variables such as *Power_kW*, *GenSpeedRelay*, unit conversion is carried out. Finally, the feature variables in the two datasets are trained and tested after normalizing by *Z-score* standardization.

4.2.1 Data Labeling

Because the LDT data set does not clearly label the normal and abnormal samples, the isolated forest is used to divide the LDT data into two clusters to represent the normal and abnormal samples. The parameters used in the construction of isolated forest model are shown in Table 4.1. All 34 features and 256 samples are extracted from the data set each time to train 100 random trees. The proportion of abnormal samples is set to 0.04, which is close to that in EDP training set.

Parameters	Values
n_estimators	50
max_samples	256
contamination	float(0.04)
max_features	float(1.0)

Table 4.1: Parameter Setting for Isolation Forest

Figure 4.3 shows a example of the labels generated after isolated forest treatment. Normal samples (majority) are represented by purple dots and abnormal samples (minority) are represented by yellow dots. It is obvious that the abnormal samples are well divided. Finally, there are 14808 normal samples and 619 abnormal samples in the total dataset.

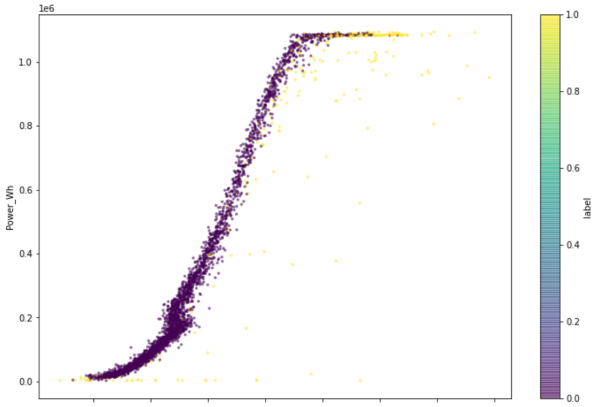

Figure 4.3: LDT Power Generation versus Wind Speed in January 2017

4.3 Summary

In this chapter, the data preprocessing method is applied to EDP and LDT, two open source SCADA datasets of wind turbines. For EDP open data set, firstly, this paper marks the samples in the operation dataset according to the text and time records in the alarm log dataset, and obtains an unbalanced data set. Then, four different resampling methods are used to resample the dataset. Finally, Pearson correlation coefficient is used for feature selection. For the LDT data set, because the original data is not given a clear label, the isolated forest is used to cluster all samples, and the discrete samples are regarded as abnormal samples.

Code	Category	Text
191	Turbine	A Ctrl: _V P.Vel:__°/s
192	Turbine	A CtrlV.STD __._V MEAN __._V
313	Ambient	Ambient temperature high: __℃
83	Turbine	B Ctrl: _._V P.Vel: _._°/s
704	Turbine	Ch High res. load C_ Load:_%
598	Turbine	Circuit breaker open
414	Turbine	EmcPitchAvel: _._°/s,Ang: _._°
102	User	Emergency circuit open
343	Turbine	Encoder signal error _:__
176	Turbine	Error on all wind sensors
315	Grid	ExEx low voltage L_:__V
127	Grid	Extr. low voltage L_:__V
336	Turbine	Ext. High cur. Grid inv. L_
335	Turbine	ExtHighIRotorInv phase:_
889	Turbine	External RPM not Reset
749	Turbine	Extra info. Err:__P:____._kW
174	Turbine	Feedback=_,:Hydraulicmotor
182	Turbine	Feedback=_, yawing CCW _
386	Turbine	Feedback=_,:VCS fan S____
71	Turbine	Feedback=_, Int.Gen.Fan S___A
72	Turbine	Feedback=_, Int.Gen.Fan S___B
73	Turbine	Feedback=_, Ext.Gen.Fan S____
172	Turbine	Feedback=_, Nacelle fan S___
173	Turbine	Feedback=_, Gearoil cool S___
189	Turbine	Feedback=_, Brake
443	Grid	Frequency error 1: ___.__ Hz
333	Turbine	High cur.rotor inv. L_:____A
147	Turbine	High gear temp:___℃
168	Turbine	High temp top ctrl.:___℃
326	Turbine	High temp. Aux. ___℃
151	Turbine	High temp. Gen bearing _:___℃
324	Turbine	High temp. VCP Board ___℃
149	Turbine	High temperature T53: ___℃ L_
144	Ambient	High windspeed: __._m/s
161	Turbine	Hydr max time: __sec
30	Turbine	Internal sublogic error
163	Turbine	Low workingpressure:___._bar
725	Turbine	No RT, High Rotor Cur L_:____A
216	Turbine	Oil leakage in Hub
341	Turbine	OVPHwErr
87	Turbine	Pitch dev. min:__._° max:__._°
352	User	Q7 breaker open
353	User	Q8 breaker open
604	Turbine	Remote Reboot
328	Turbine	Rotor inv. HW error L_
691	Turbine	SignalError. ____,__
338	Turbine	Slip:___ above limits_
312	Turbine	Thermo error, ventilators T53
169	Turbine	Thermoerr. Nac. fan F___
186	Turbine	Thermoerror yawmotor F___
100	Turbine	Too many auto-restarts:____
128	Grid	Trip Q8 L_:___V
381	Turbine	WS1 timeout err.
200	Turbine	YawSpeedFault

Table 4.2: Fault Codes

Chapter 5

Experiments and Results

5.1 Training in EDP Open Dataset

This section involves two training process using two machine learning techniques in EDP Open Dataset: ANN and AdaBoost. They are considered as two popular machine learning models in the field of fault detection.

5.1.1 Artificial Neuron Network

Hyper-parameters	Values
No. of neurons in the hidden layer	30, 40, 50, 60
Batch size	16, 32, 64

Table 5.1: ANN Debugging Contents

Firstly, this experiment uses an artificial neural network model to predict the anomaly of wind turbines. At present, there is no specific standard to build an artificial neural network architecture. Different architectures are needed for different types of classification problems. In this study, grid search is used to debug some hyper-parameters (such as the number of neurons, and batch size). The debugging contents are shown in Table 5.1. Then, the optimizer, initial learning rate, activation function, and loss function are compared. Finally, the ANN model with two layers is used. Although adding more hidden layers can make the model more in-depth, the overall performance changes little, and more training cost and complexity are needed. The output layer of the model uses the Sigmoid activation function to get a value between 0 and 1. The output greater than the classification threshold is classified as abnormal, and the rest are classified as normal labels. In addition, the 5-fold cross-validation is used for training, and the one with the highest accuracy among the five models is saved as the best model.

Hyperparameters	Values
No. of neurons in the hidden layer	60
Batch size	32
Epochs	50
Learning Rate	0.01
Dropout	0.3
Activation function in hidden layer	ReLU
Activation function in output layer	Sigmoid
Optimizer	RMSProp
Loss function	Binary Cross Entropy

Table 5.2: Optimal Parameter Setting of ANN Model

Another purpose of this experiment is to explore the influence of different resampling methods on the model. The data after resampling and feature selection are trained. According to the results of cross-validation, the final optimal parameter setting is shown in Table 5.2, it achieves the best validation result when using undersampling and SMOTE+ENN. Then, the best model obtained after training is applied to the test set, and the criteria values are shown in Table 5.3. Because the experiment only

Sampling Methods	Precision	Recall	F1 Score
Base Model	**0.93**	0.79	0.86
Random Undersampling	0.87	0.87	**0.87**
SMOTE	0.85	0.90	**0.87**
SMOTE+ENN	0.79	**0.92**	0.85

Table 5.3: Testing Dataset Results for ANN Model (Classification Threshold = 0.5)

focus on abnormal samples in anomaly detection, the predicted results of normal samples can not be used as the main evaluation basis. According to the results in the table, it can be found that the dataset without oversampling has the highest precision rate, but the recall rate is the lowest, which may be due to the poor learning ability of the model and more bias toward most classes in prediction. In contrast, the model trained with the data after resampling has high precision and recall rate.

Figure 5.1 shows the ROC curve of 4 sampling methods generated by *sklearn* library in Python. It is obvious that the AUC of the model without resampling is much smaller than others. Although the resampling method including SMOTE takes more time to generate new samples, the actual training effect is not better than the result of random undersampling.

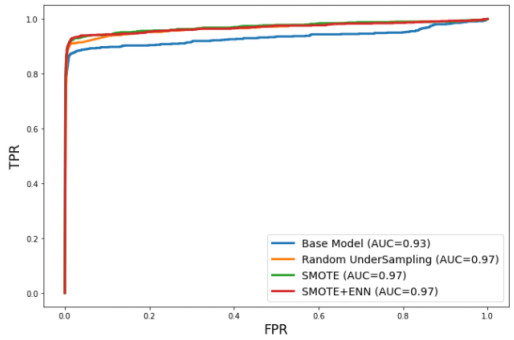

Figure 5.1: ROC Curve for ANN Model

5.1.2 AdaBoost

Hyperparameters	Values
Base estimator	Decision Tree Classifier
No. of estimators	100
Learning Rate	0.1

Table 5.4: Parameter Setting of AdaBoost Model

In the case of AdaBoost, grid search is not applied, and all the set parameters are shown in Table 5.4. What is noteworthy is the "base estimator" parameter. In this experiment, 100 decision tree classifiers are used as weak classifiers. In addition, the learning rate is set to 0.1. Like ANN, this work adopts four training sets with different resampling methods for training, and the test results are shown in Table 5.5.

It can be seen from the table that the model trained in the balanced data set after resampling has low precision but high recall. However, the model trained in the imbalanced data set achieved 87% precision and 85% recall and has the best F1 score among the four models. Thus, the AdaBoost model does not require classes to be balanced in the dataset. This is because, in each iteration, the samples of minority classes are given more weight, which makes them more concerned in the next iteration. This idea is also discussed in [44].

Sampling Methods	Precision	Recall	F1 Score
Base Model	**0.87**	0.85	**0.86**
Random Undersampling	0.75	0.92	0.82
SMOTE	0.71	0.92	0.81
SMOTE+ENN	0.71	**0.93**	0.82

Table 5.5: Testing Dataset Results for AdaBoost Model (Classification Threshold = 0.5)

5.2 Knowledge Transfer to LDT Dataset

Features learned from the source domain will be transferred to the target domain, so there is no need to use LDT data for any additional training. In this experiment, the ANN model trained by random undersampling data set is used for knowledge transfer, because it has higher prediction results and minimum training cost compared with other models.

Month	Precision	Recall	F1 Score	Accuracy
January	0.33	0.93	0.49	92.18%
February	0.31	0.99	0.47	91.10%
March	0.20	0.98	0.33	83.85%
April	0.51	0.79	0.62	96.12%
May	0.13	0.30	0.19	89.34%
June	0.40	0.86	0.54	94.20%
Overall	**0.31**	**0.88**	**0.46**	**91.82%**

Table 5.6: Testing Result in LDT Dataset

After the LDT data is put into the model as the test set, the predicted results are compared with the labels obtained after isolated forest processing, and the specific results are displayed in the Table 5.6. It can be seen from the table that the model has achieved considerable prediction results in January, February, March, April, and June, and more than 90% of the anomalies from January to March are correctly identified. The prediction effect of the model in May is very poor because there are a lot of errors in the data in May (such as power generation less than 0 and wind speed less than 0), which makes a lot of data filtered during preprocessing, and only a small amount of data are put into the model to test, resulting in a great contingency in the results. Finally, in the data of the whole half-year, the prediction accuracy of the model reached 91.82%. Although the precision of 31.49% means that there are a large number of false alarms, which may increase the maintenance cost, it is gratifying that 88.37% of abnormal samples are detected, which means that the model can reduce the occurrence of errors to prevent huge disasters. Because the prediction is completely based on the knowledge learned in another domain, it also proves that the model has good generalization ability. Moreover, Figure 5.2 shows the confusion matrix of the evaluation in the target domain.

5.3 Comparison with the Model Trained on the LDT Dataset

Another method to directly measure the effect of transfer learning is to train the dataset in the target domain again and compare the results of the test set in the training with the results of knowledge transfer, so as to conclude whether knowledge transfer can save training costs while ensuring the results. Although the starting point of this experiment is that LDT data set is too complex and huge, the author hopes to save computing cost through migration learning. However, due to the need to verify the experimental results, a lot of time was spent to design the control experiment.

For the whole LDT dataset, comparative experiments are carried out. Firstly, the data in all LDT datasets are preprocessed, 14808 abnormal samples and 14808 normal samples are obtained through random undersampling, and trained with the same feature space and parameter settings as the previous

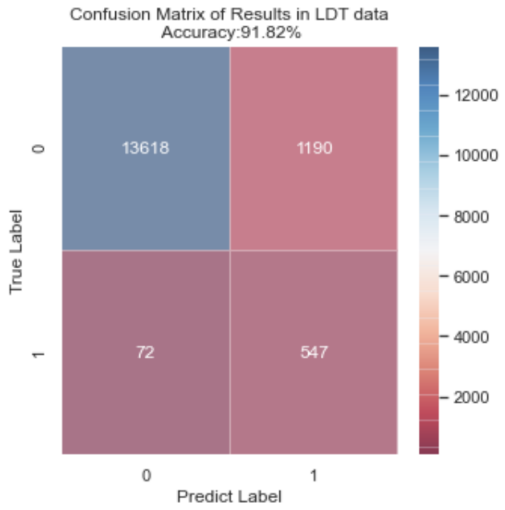

Figure 5.2: Confusion Matrix of Results in LDT data

experiments. The final training results after cross-validation are shown in Table 5.7.

Precision	Recall	F1 Score	Accuracy
0.50	0.88	0.64	0.92

Table 5.7: Testing Result in LDT Dataset

Compared with the results after transfer learning in Table 5.6, it is surprising to find that for the detection rate (recall) of abnormal samples, the model obtained by knowledge transfer is basically consistent with the model obtained by retraining. Therefore, it can be concluded that the knowledge transfer of this experiment is successful to some extent. For the precision of the model, the retrained model is improved from 33% to 50% compared with the knowledge transfer model. Although this is a significant improvement, it still means that there is a 50% false alarm rate in the model, which will undoubtedly increase the maintenance cost of the wind turbines. The high false alarm rate means that the model training does not achieve the best effect, which may be caused by the following reasons: (1) The original LDT data is put into the final training of the model after a lot of calculation and preprocessing, which may lead to the difference between the final training set and the real data. (2) When labeling the LDT data, the final result may be different from the real sample distribution. (3) There is too much noise in the dataset, which leads to the low quality of the overall data and affects the training of the model.

5.4 Summary

In this chapter, the AI algorithm discussed is applied to EDP Open data and the LDT dataset. Firstly, after obtaining the preprocessed EDP dataset, two common machine learning classification models ANN and AdaBoost are used for training. ANN model achieves 92% recall rate and AdaBoost model achieves 93% recall rate. These prediction results prove the effectiveness of the machine learning method in wind turbine fault detection. Then, this paper adopts the ANN model with the best training results and uses the pre-trained model parameters to transfer the knowledge to the LDT dataset. The model achieves 88% of the test results in the target domain. Finally, this work trains the target domain (LDT dataset) using the same feature space, compare the training results with the results of transfer learning, and finds that the results of transfer learning are only less than the retrained model in precision rate, but the recall rate is similar to the retrained model, which proves that the ANN model has certain generalization. For different datasets, the model has high prediction accuracy and recall rate, but the precision is relatively low, which means that there is a high false alarm rate, which may be a problem to be solved in the future.

Chapter 6

Summary and Reflection

This paper is committed to using the machine learning model prediction method to predict the real-time fault based on the SCADA data of wind turbine, and transfer the learned knowledge from the source domain to the target domain, which can save the training time in the target domain. This study shows that using machine learning method is very effective in the field of fault prediction of wind farm. It can carry out predictive maintenance according to historical SCADA data without detailed understanding of wind farm system. In addition, transfer learning is also very promising for the wind power industry. For example, the two datasets used in this experiment come from two completely different wind turbines in different regions, but they can achieve good results on the same model. When the wind farm does not have enough SCADA data for training or the SCADA data does not have clear abnormal labels, the wind power operator can consider using the method of transfer learning for maintenance.

6.1 Contribution

This work mainly makes the following contributions:

- Ann and AdaBoost machine learning classification models are applied to wind turbine fault prediction, and reach 93% recall rate in thhe EDP open dataset.
- The knowledge of learning in the source domain are transplanted to the target domain, and the recall rate of 88% is achieved in the LDT dataset, which verifies the effectiveness of transfer learning.

6.2 Future Work

Our future work will focus on:

- More machine learning classification models will be used for training to pursue higher anomaly prediction rate and lower false alarm rate, so as to explore the best model design.
- The machine learning model will be transferred to more wind power datasets to construct the model with higher generalization.
- The work will be expended to SCADA datasets in other industrial fields, and explore the possibility of introducing artificial intelligence models in other industries.

6.3 Self Reflection

The objectives of the project have been changing throughout the year. From the beginning, I tried to use a more complex neural network model to model the normal behavior of SCADA data. Later, I

gave up because of the poor effect of model construction and the deformity of the data set itself. When I found an open source dataset that met the requirements of the classification model on the network, I felt that I changed the project goal. Then I tried to apply the original incomplete dataset through migration learning, and achieved success. The task was carried out according to the GANTT chart in the project proposal. Although it changed due to special circumstances or difficulties, the overall goal was achieved. The whole project adopts an agile development framework similar to scrum, and a meeting with the tutor is arranged for a sprint every week. The main difficulty encountered in the whole project is the analysis of the logs in the data set. I need to check the contents of the logs one by one from the Internet and mark the data. In addition, through data processing and model building, my Python coding ability can be greatly improved in self-study. Through the discovery of problems and discussion with my tutor, my problem-solving ability has also been improved. I think the overall arrangement and time management of the project are relatively successful. In addition to being able to finish my work on time before deadline, I can also finish my homework in other courses well. In short, this is a very valuable experience, and I sincerely thank my supervisor for his advice and help.

References

[1] GWEC. Global wind report 2021. `https://gwec.net/global-wind-report-2021/`. 2021.

[2] James F Manwell, Jon G McGowan, and Anthony L Rogers. *Wind energy explained: theory, design and application*. John Wiley & Sons, 2010.

[3] Fausto Pedro García Márquez, Andrew Mark Tobias, Jesús María Pinar Pérez, and Mayorkinos Papaelias. Condition monitoring of wind turbines: Techniques and methods. *Renewable Energy*, 46:169–178, 2012.

[4] Rama S Velmurugan and Tarun Dhingra. Maintenance strategy selection and its impact in maintenance function: A conceptual framework. *International Journal of Operations & Production Management*, 2015.

[5] Qi Hao, Yunjiao Xue, Weiming Shen, Brian Jones, and Jie Zhu. A decision support system for integrating corrective maintenance, preventive maintenance, and condition-based maintenance. In *Construction Research Congress 2010: Innovation for Reshaping Construction Practice*, pages 470–479, 2010.

[6] Alan Jović, Karla Brkić, and Nikola Bogunović. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. IEEE, 2015.

[7] Wisdom Udo and Yar Muhammad. Data-driven predictive maintenance of wind turbine based on scada data. *IEEE Access*, 9:162370–162388, 2021.

[8] Junchao Guo, Dong Zhen, Haiyang Li, Zhanqun Shi, Fengshou Gu, and Andrew D Ball. Fault detection for planetary gearbox based on an enhanced average filter and modulation signal bispectrum analysis. *ISA transactions*, 101:408–420, 2020.

[9] Stephan Schmidt, P Stephan Heyns, and Konstantinos C Gryllias. A methodology using the spectral coherence and healthy historical data to perform gearbox fault diagnosis under varying operating conditions. *Applied Acoustics*, 158:107038, 2020.

[10] Wei Teng, Xian Ding, Hao Cheng, Chen Han, Yibing Liu, and Haihua Mu. Compound faults diagnosis and analysis for a wind turbine gearbox via a novel vibration model and empirical wavelet transform. *Renewable energy*, 136:393–402, 2019.

[11] Walid Touti, Mohamed Salah, Khmais Bacha, and Abdelkader Chaari. Condition monitoring of a wind turbine drivetrain based on generator stator current processing. *ISA Transactions*, 2021.

[12] Meik Schlechtingen and Ilmar Ferreira Santos. Comparative analysis of neural network and regression based condition monitoring approaches for wind turbine fault detection. *Mechanical Systems and Signal Processing*, 25(5):1849–1875, 2011.

[13] Ling Xiang, Penghe Wang, Xin Yang, Aijun Hu, and Hao Su. Fault detection of wind turbine based on scada data analysis using cnn and lstm with attention mechanism. *Measurement*, 175:109094, 2021.

[14] Michael Wilkinson, Brian Darnell, Thomas Van Delft, and Keir Harman. Comparison of methods for wind turbine condition monitoring with scada data. *IET Renewable Power Generation*, 8(4):390–397, 2014.

[15] Kyusung Kim, Girija Parthasarathy, Onder Uluyol, Wendy Foslien, Shuangwen Sheng, and Paul Fleming. Use of scada data for failure detection in wind turbines. In *ASME 2011 5th International Conference on Energy Sustainability*, pages 2071–2079. American Society of Mechanical Engineers Digital Collection, 2011.

[16] Pavlos Trizoglou, Xiaolei Liu, and Zi Lin. Fault detection by an ensemble framework of extreme gradient boosting (xgboost) in the operation of offshore wind turbines. *Renewable Energy*, 179:945–962, 2021.

[17] Zhen-You Zhang and Ke-Sheng Wang. Wind turbine fault detection based on scada data analysis using ann. *Advances in Manufacturing*, 2(1):70–78, 2014.

[18] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Andrew Kusiak and Anoop Verma. A data-driven approach for monitoring blade pitch faults in wind turbines. *IEEE Transactions on Sustainable Energy*, 2(1):87–96, 2010.

[21] Joyjit Chatterjee and Nina Dethlefs. Deep learning with knowledge transfer for explainable anomaly prediction in wind turbines. *Wind Energy*, 23(8):1693–1710, 2020.

[22] Kevin Leahy, R Lily Hu, Ioannis C Konstantakopoulos, Costas J Spanos, and Alice M Agogino. Diagnosing wind turbine faults using machine learning techniques applied to operational data. In *2016 ieee international conference on prognostics and health management (icphm)*, pages 1–8. IEEE, 2016.

[23] Niya Chen, Rongrong Yu, Yao Chen, and Hailian Xie. Hierarchical method for wind turbine prognosis using scada data. *IET Renewable Power Generation*, 11(4):403–410, 2017.

[24] Zijun Zhang and Andrew Kusiak. Monitoring wind turbine vibration based on scada data. *Journal of Solar Energy Engineering*, 134(2), 2012.

[25] Yingning Qiu, Yanhui Feng, Peter Tavner, Paul Richardson, Gabor Erdos, and Bindi Chen. Wind turbine scada alarm analysis for improving reliability. *Wind Energy*, 15(8):951–966, 2012.

[26] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[27] Ldt turbine scada-1sec. `https://pod.ore.catapult.org.uk/product/2`. 2018.

[28] Qiang Yang. An introduction to transfer learning. In *International Conference on Advanced Data Mining and Applications*, pages 1–1. Springer, 2008.

[29] He Ren, Wenyi Liu, Mengchen Shan, and Xin Wang. A new wind turbine health condition monitoring method based on vmd-mpe and feature-based transfer learning. *Measurement*, 148:106906, 2019.

[30] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, jun 2002.

[31] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

[32] Emin Elmar oglu Mammadov. Predictive maintenance of wind generators based on ai techniques. Master's thesis, University of Waterloo, 2019.

[33] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.*, 2:408–421, 1972.

[34] Pavlos Trizoglou, Xiaolei Liu, and Zi Lin. Fault detection by an ensemble framework of extreme gradient boosting (xgboost) in the operation of offshore wind turbines. *Renewable Energy*, 179:945–962, 2021.

[35] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, Berlin, Heidelberg, 2009: 1-4.

[36] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[37] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[38] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

[39] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[40] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *icml*, volume 96, pages 148–156. Citeseer, 1996.

[41] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[42] Edp open data - wind turbine failure detection. `https://opendata.edp.com/pages/challenges/#description`. January 2019.

[43] Fehlerliste v90. `https://dokumen.tips/documents/fehlerlistev902944665r11.html`. April 2015.

[44] Yanmin Sun, Mohamed S Kamel, and Yang Wang. Boosting for learning multiple classes with imbalanced class distribution. In *Sixth international conference on data mining (ICDM'06)*, pages 592–602. IEEE, 2006.