

TECHNISCHE UNIVERSITÄT
CHEMNITZ

Detection of Mass Imbalance Fault in Wind Turbine using Data Driven Approach

Master Thesis

Dept. of Computer Science
Chair of Computer Engineering

Submitted by: Guhan velupillai Gowthaman Malarvizhi
Student ID: 563442
Date: 16.05.2023

Supervising tutor: Prof. Dr. W. Hardt
Mrs.Ummay Ubaida Shegupta, Mr. Johannes Fricke

Abstract

Optimizing the operation and maintenance of wind turbines is crucial as the wind energy sector continues to expand. Predicting the mass imbalance of wind turbines, which can seriously damage the rotor blades, gearbox, and other components, is one of the key issues in this field. In this work, we propose a machine learning-based method for predicting the mass imbalance of wind turbines utilizing information from multiple sensors and monitoring systems. We collected data and trained the model from Adwen AD8 wind turbine model and evaluated on the real wind turbine SCADA data which is located at Fraunhofer IWES, Bremerhaven. The data included various parameters such as wind speed, blade root bending moments and rotor speed. We used this data to train and test machine learning classification models based on different algorithms, including extra-tree classifiers, support vector machines, and random forest. Our results showed that the machine learning models were able to predict the mass imbalance percentage of wind turbines with high accuracy. Particularly, the extra tree classifiers with blade root bending moments outperformed other research for multiclassification problem with an F1 score of 0.91 and an accuracy of 90%. Additionally, we examined the significance of various features in predicting the mass imbalance and observed that the rotor speed and blade root bending moments were the most crucial variables. Our research has significant effects for the wind energy sector since it offers a reliable and efficient way for predicting wind turbine mass imbalance. Wind farm operators can save maintenance costs, minimize downtime of wind turbines, and increase the lifespan of turbine components by identifying and eliminating mass imbalances. Also, further investigation will allow us to apply our method to different kinds of wind turbines, and it is simple to incorporate into current monitoring systems as it supports prediction without installing additional sensors. In conclusion, our study demonstrates the potential of machine learning for predicting the percentage of mass imbalance of wind turbines. We believe that our approach can significantly benefit the wind energy industry and contribute to the development of sustainable energy sources.

Keywords: Mass Imbalance, Wind Turbines, Condition Monitoring Systems, SCADA Data, Rotor Speed, Blade Root Bending Moments, Wind Speed, ExtratreesClassifier, Multiclassification

Contents

List of Figures	4
List of Tables	6
List of Acronyms	7
1 Introduction	8
1.1 Wind turbine technology / Condition Monitoring	8
1.1.1 Problem Statement	13
1.1.2 Objective	14
1.1.3 Scope of the thesis	14
1.1.4 Research Questions	14
2 State of the art	15
2.1 Wind turbine technology / Condition Monitoring	15
2.2 Mass Imbalance in wind turbines	17
2.2.1 Rotor Blades	17
2.2.2 Hub	18
2.2.3 Generator	18
2.2.4 Nacelle	18
2.3 Data Driven Approaches for Mass Imbalance Detection:	19
2.4 Feature Engineering Techniques	21
2.4.1 Feature Scaling	21
2.4.2 Feature selection	21
2.4.3 Domain Specific Feature Engineering	22
2.4.4 Overview of the Thesis	24
3 State of the technology	26
3.1 Mass Imbalance:	26
3.2 Power Spectral Density	30
3.3 Welch's Method	31
3.4 Machine Learning Algorithms	32
3.4.1 Algorithm1 – Logistic Regression Classifier	35
3.4.2 Algorithm 2: KNN classifier	38

CONTENTS

3.4.3	Algorithm 3: Random Forest	40
3.4.4	Algorithm 4: Extra-trees Classifier	43
3.4.5	Algorithm 5: Support Vector Machine	45
3.5	Model Validation	51
3.5.1	K-Fold Cross Validation	51
3.5.2	Hold Out Cross Validation	52
3.5.3	Stratified K-fold Cross Validation	53
3.5.4	Leave One Out Cross Validation	54
3.6	Hyperparameter Tuning	54
3.6.1	Random search	54
3.6.2	Grid search	54
3.7	Classification performance metrics:	55
3.7.1	Confusion Matrix	55
3.7.2	Accuracy	56
3.7.3	Precision Score	56
3.7.4	Recall score	56
3.7.5	F1 score	57
3.8	Flask Framework	58
4	Implementation	60
4.1	Mass Imbalance Detection Techniques	60
4.1.1	Impact	60
4.1.2	Detection	60
4.2	Approach 1: Rotor Speed and Wind Speed	62
4.3	Approach 2: Blade Root Bending Moments and Wind Speed	68
5	Results and discussion	77
5.1	Limitations	81
6	Conclusions	82
6.1	Future Work	82
	Bibliography	84

List of Figures

1.1	The renewable energy capacity per country [1]	9
1.2	Worldwide renewable energy consumption in 2021 [1]	10
1.3	The bathtub curve [2]	11
1.4	Percentage of faults per components [2]	12
1.5	Component failure rate along with downtime [2]	13
2.1	Condition Monitoring Process[3]	16
3.1	Wind Turbine Model[4]	27
3.2	The Factor Affecting the Wind Turbine Model[4]	28
3.3	The Mass Imbalance Model[4]	29
3.4	The Types of Machine Learning[5]	32
3.5	The Machine Learning Methodologies[5]	34
3.6	Various Types of Machine Learning Algorithms[6]	35
3.7	Logistic Regression Model[7]	36
3.8	Ensemble Techniques[8]	41
3.9	Performance of ExtratreesCassifier Model[9]	45
3.10	Support Vectors[10]	46
3.11	multiclass seperation[10]	47
3.12	One vs One Approach[10]	48
3.13	One vs Rest Approach[10]	49
3.14	Cross Validation Techniques[11]	52
3.15	Hold Out Validation - Train/Test Split[11]	53
3.16	Hold Out Validation - Train/Validation/Test Split[11]	53
3.17	Leave One Out Cross Validation[11]	54
3.18	Grid Search vs Random Search[11]	55
3.19	Terminologies of Confusion Matrix[12]	57
3.20	The Infrastructure of Flask Framework[13]	58
4.1	Time Series Signature of Rotor Speed AD8 Data	62
4.2	1p Frequency of the Signal	63
4.3	PSD Plot of Mass Imbalance 2%	64
4.4	Trained Multiple Classificaion Models	67
4.5	Standard Deviation of Bending Moments for all Three Blades	71

LIST OF FIGURES

4.6	Correlation Matrix of the Features	72
4.7	Feature Importance of the Input Features	73
4.8	Class Balancing Nature	74
4.9	Trained Multiple Classification Models	76
5.1	Cross Validation plot for MASS Imbalance Model	77
5.2	Confusion Matrix of Extratreesclassifier Model	78
5.3	Classification Report	79
5.4	Mass Imbalance Detection Application	81

List of Tables

2.1	Pros and Cons of Different Methods for Detecting Imbalance	25
4.1	Summary of Input Features, Class Labels, and Feature Engineering Technique for the Mass Imbalance Fault Signature Dataset for the Approach 1	62
4.2	Hyperparameters and their Corresponding Values	65
4.3	Grid Search Hyperparameter Values	66
4.4	Summary of Approach 1	68
4.5	Input Features, Class Labels, and Feature Engineering Technique for Detecting Mass Imbalance Faults in Wind Turbines	70
4.6	Hyper-parameters and their Corresponding Values	75
4.7	Hyper-parameters Chosen by Grid Search	75
4.8	Summary of Approach 2	76
5.1	Accuracy Measurements for Different Test Datasets	80

List of Acronyms

- CM** Condition Monitoring. 15
- CNN** Convolutional Neural Network. 20
- DNV-GL** Det Norske Veritas. 29
- DTFT** Discrete Time Fourier Transform. 30
- EU** European Union. 8
- FAST** Fatigue, Aerodynamics, Structures, and Turbulence. 20
- GW** Gigawatt. 8
- KNN** K Nearest Neighbour. 23
- LCOE** Levelized Cost Of Electricity. 16
- LSTM** Long Short Term Memory. 20
- MI** Mass Imbalance. 15, 26
- ML** Machine Learning. 21
- MW** Megawatt. 10
- PSD** Power Spectral Density. 14
- SCADA** Supervisory Control and Data Acquisition. 14
- SMOTE** Synthetic Minority Over-sampling Technique. 39
- SVM** Support Vector Machine. 45
- TURBSIM** Turbulence Simulator. 20
- WT** Wind Turbine. 8

1 Introduction

1.1 Wind turbine technology / Condition Monitoring

As the world's energy needs increase for reasons of both energy security and reducing greenhouse gas emissions, a large-scale usage of renewable energy sources is imminent. One of the most practical sources of alternative energy is wind power, but in order for there to be more installations, wind power needs to be more dependable and less expensive. These may be ensured by making sure that the wind resource is used to produce electricity as efficiently as possible and by having a generally reduced operating cost. Since wind energy is one of the most environmentally benign and sustainable energy sources, Wind Turbine (WT) is expanding quickly. By 2026, the European Union (EU) wants to double the amount of renewable energy it produces. According to Energy Voice's study from 2020, this goal is being driven mostly by investments in wind energy. In the UK, the combined installed capacity of solar and wind power facilities will reach 64 Gigawatt (GW) in 2026. The installed offshore wind turbine capacity in this nation is anticipated to increase from 10.5 GW in 2020 to 27.5 GW in 2026 [14]. In recent years, wind energy has grown and developed significantly compared to other sources. Each year, the capacity of produced wind power rises by 20% globally, reaching a total of 743 gigawatts [15]. Wind power output grew by about 273 TWh (up 17%) in 2021, the greatest rise of all power generation methods and an expansion rate of 45% greater than that of 2020 [16]. Wind energy usage increased in the US in 2021, giving millions of their citizens access to affordable renewable energy sources. In 2021, the U.S. wind sector added 13,413 megawatts (MW) of brand-new wind power, raising the total installed wind capacity to 135,886 MW² [17]. Over 9% of the nation's power comes from wind, more than 50 percent in South Dakota and Iowa, as well as more than 30% in Kansas [18].

- A sustainable and clean energy source, wind energy emits no harmful emissions or contaminants.
- Wind energy is a versatile source of energy since it can be harnessed on both onshore and offshore.
- Wind energy has its potential to lessen reliance on fossil fuels while also increasing energy security over time.
- Wind energy can also help to reduce greenhouse gas emissions and possibly effects of climate change.

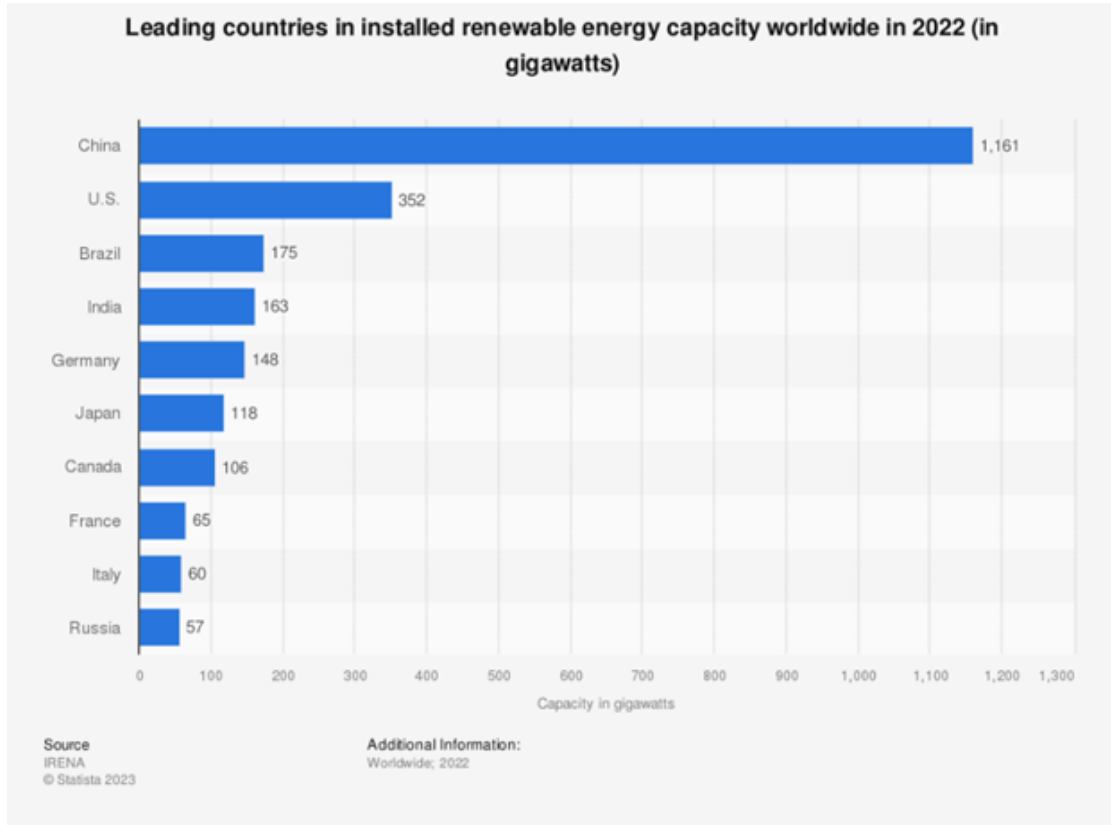


Figure 1.1: The renewable energy capacity per country [1]

Wind energy has had tremendous growth and development in Germany and widely in Europe. By the tail end of 2017, Germany had 55.6 GW of installed wind power, with 5.2 GW coming from offshore installations. Wind energy provided 25% of the nation's overall energy supply in 2019 compared to an estimated 9.3% in 2010 [?]. More current statistics show that Germany's onshore wind energy capacity has risen, rising from 56,046 megawatts in 2015 to 58,186 megawatts in 2022. In terms of renewable energy sources, wind power contributed the most to Germany's mix of energy sources in previous years [19].

According to the International Energy Agency, wind energy might provide more than 420,000 terrawatt hours annually worldwide by 2025, providing up to 18% of the world's power needs [20]. Also the benefits of renewable energy resource such as wind energy is enormous which includes the reduction of greenhouse gas emissions that improves the energy security and increase in number of jobs for the local communities [16]. Mongolian renewable energy development offers significant promise due to its plentiful resources and advantageous geographical position. The development of solar and wind power facilities provides a chance to meet peak electricity demands while decreasing dependency

on imported energy[21]. The Renewable Energy Law, and subsequent changes, have effectively drawn foreign investment and promoted sectors expansion. However, the law's provision that electricity sales be made in US dollars has created complications due to exchange rate swings. Since 2007, the average USD exchange rate has climbed by 143.41%, resulting in large increases in solar and wind power plant electricity rates[21]. To solve this issue, it is proposed that the law be altered to allow for the selling of power in local currency, thereby reducing the influence of exchange rate changes on consumer pricing. Furthermore, the inclusion of a 30 Megawatt (MW) solar power plant and a 102 MW wind power facility would raise the feed-in tariff, promoting renewable energy development. Mongolia should encourage competition in the renewable energy sector and work on lowering investment costs to correspond with dropping global equipment prices in order to achieve long-term sustainability. This approach, together with a focus on cost-effective solutions, will allow the country to capitalize on its renewable energy potential and contribute to a more economical and sustainable energy system[21].

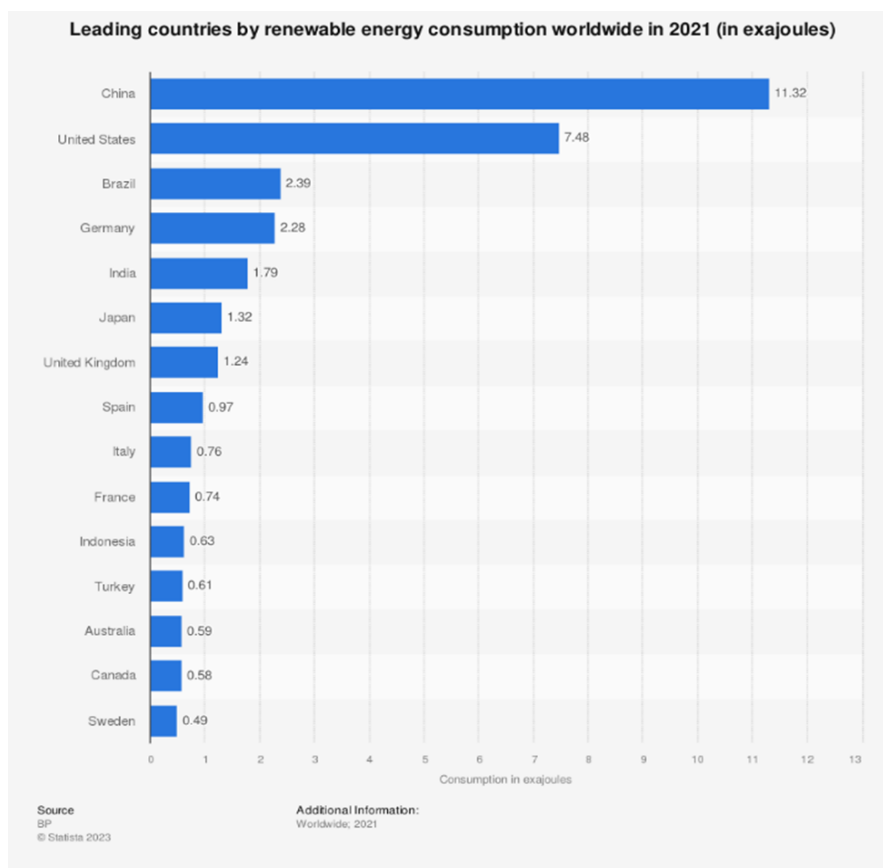


Figure 1.2: Worldwide renewable energy consumption in 2021 [1]

Additionally, the need to lower the cost of energy has caused a steady increase in WT size over the past few decades. However, this increases the structural strains on the various WT components, which quickly raises the cost of maintenance. The field of automated failure detection in WTs is rapidly developing, primarily to cut down on the downtime and maintenance expenses of the turbine. Early anomaly detection allows the maintenance crews to properly plan their work and, most significantly, increases the likelihood that catastrophic failure won't occur. WTs can fail in a variety of ways, including through bearing erosion, blade imbalance, aerodynamic imbalance, and mass imbalance [2]. The state of the wind turbines must be regularly checked in order to increase the safety concerns, reduce downtime, reduce the frequency of abrupt breakdowns and the accompanying high maintenance and logistic costs, and offer reliable power output. In the WT sector, the most advanced way for establishing maintenance strategy is reliability-centered maintenance, which comprises of preventative maintenance strategy based on performance and parameter monitoring and subsequent actions. Condition-monitoring is employed in this technique to establish the optimal point between corrective and planned maintenance procedures. WTs are generally intended to work for twenty years. Time-based maintenance, like other mechanical systems, implies that the failure behavior of WTs is deterministic. Three failure modes essentially explain the failure characteristics of WT mechanical systems. Figure depicts a hypothetical failure rate vs time in a mechanical system, where

- $\beta < 1$ indicates a falling failure rate,
- $\beta = 1$ represents a steady failure rate,
- $\beta > 1$ represents an increasing failure rate

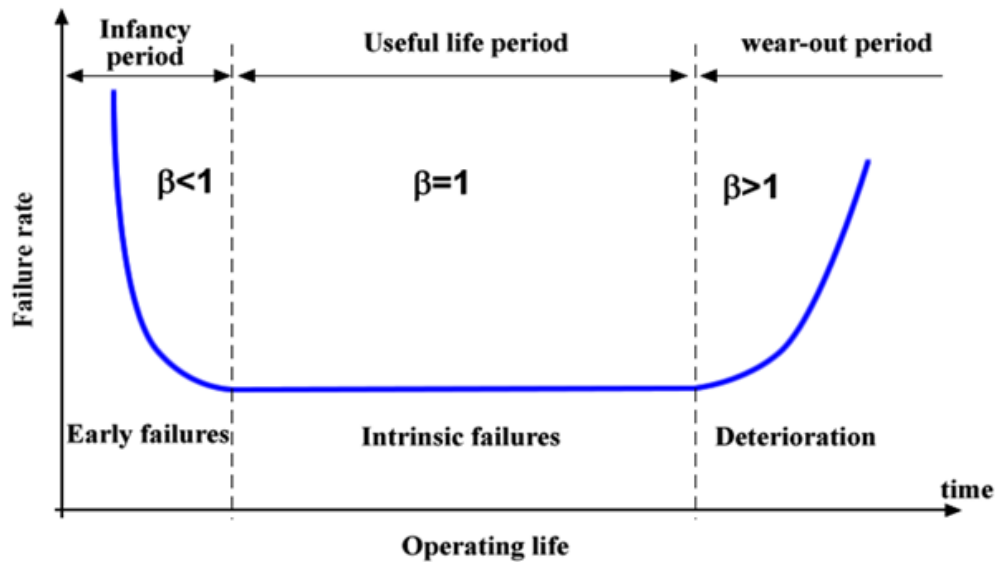


Figure 1.3: The bathtub curve [2]

- The number of fault increases linearly with the increase of size of wind turbines. There are many components in the wind turbine which is subjected to various faults with low to high severity.
- To the context of our work, the faults based on wind turbine rotor blades such as mass imbalance is discussed.
- The electrical system, control system, hydraulic system, sensors, and rotor blades are five component groups that, according to a 15-year review of 1500 WTs, account for 67% of WT failures. This is illustrated by the pie chart[2].

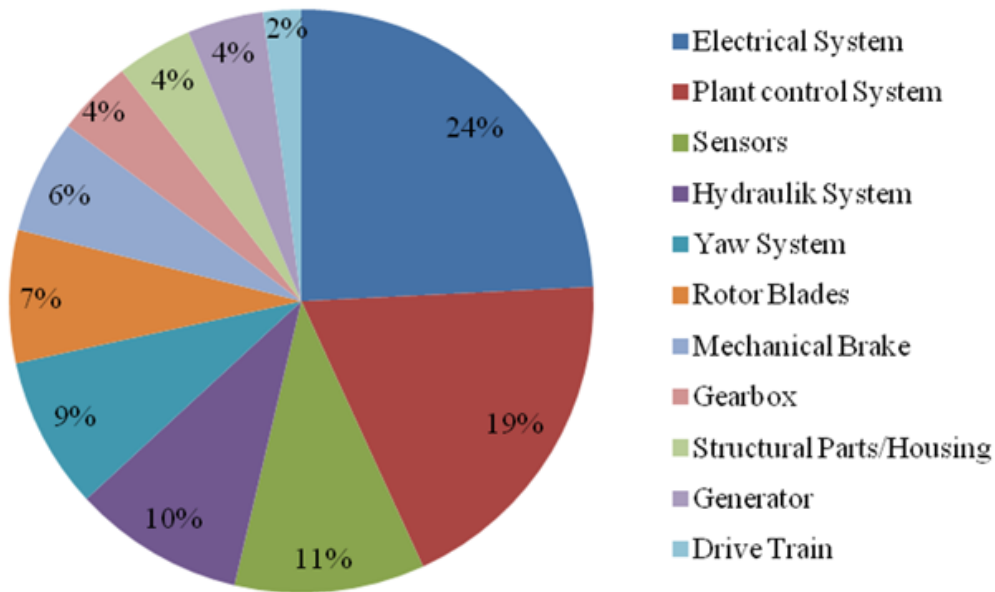


Figure 1.4: Percentage of faults per components [2]

- The various components possess different levels of faults which can be rectified by the turbine itself after restarting and some need longer downtime of the turbine. Particularly, the 19% of rotor blades fault have longer downtime compared to other components since the manual power, logistic power and maintenance cost is very high.
- Fischer et al [22] found that only 15% of WT failures account for 75% of yearly downtime. This outcome supports the findings of Haln et al [23] on the average failure rate and downtime per component. The results of this study are also in line with those of Crabtree et al. [24], who evaluated failure rates and downtime for various WT components using data from surveys of European wind-energy conversion systems.

- Figure 1.5 depicts the failure rate and downtime of various WT components. Similar findings were obtained from the reliability and downtime statistics of the Egmond aan Zee wind farm in Germany, i.e., the failure frequency of the gearbox is small, but the related downtime and expenses are large; in contrast, the failure frequency of the rotor blades is average, but the resulting downtime and costs are high when compared to gearbox failures.
- As a result, among all components, the gearbox and rotor blades has the largest proportion of lost electricity production [24].

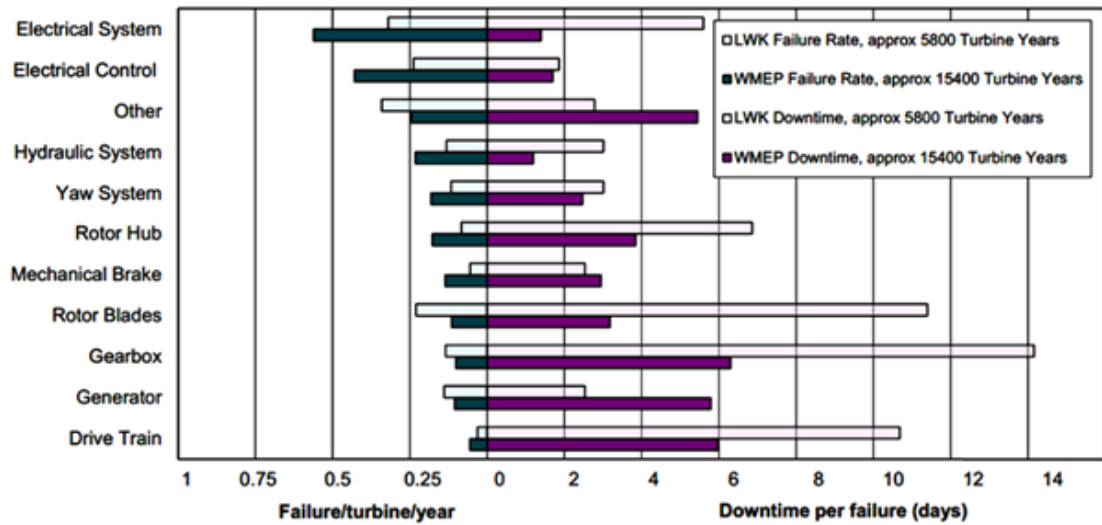


Figure 1.5: Component failure rate along with downtime [2]

1.1.1 Problem Statement

- Wind turbine mass imbalance can cause considerable damage to the blades, resulting in lower performance and increased turbine maintenance costs.
- Traditional techniques of detecting mass imbalance rely on visual examinations and manual measurements, which are prone to human error and can be time-consuming.
- The demand for frequent wind turbine blade inspections and maintenance is growing as the number of installed turbines rises globally, necessitating the development of more efficient and cost-effective methods of detecting mass imbalances.
- The application of machine learning algorithms for mass imbalance identification in wind turbines is a promising data-driven method that has the potential to cut maintenance time and costs while also enhancing detection accuracy.

1.1.2 Objective

The objectives of this thesis are as follows:

- To develop a data-driven approach for mass imbalance detection in wind turbines using machine learning algorithms.
- To evaluate the performance of the model using real-world SCADA data
- To compare the proposed approach with existing methods for mass imbalance detection in terms of accuracy, computational time, and explainability

1.1.3 Scope of the thesis

- The goal of this thesis is to provide a data-driven strategy for detecting mass imbalance in wind turbines using machine learning techniques.
- The work has been done on two approaches, one of which uses the standard deviation of blade root bending moments in the edgewise direction and mean wind speed as input features, and the other uses the Power Spectral Density (PSD) value of the 1p peak frequency of the rotor speed and mean wind speed as input features with eight different mass imbalance percentages (0%, 2%, 4%, 6%, 8%, 10%, 14%, and 18%) as output class labels.
- The final approach is tested with real-world SCADA data from Adwen AD8 wind turbine at the Fraunhofer IWES location.

1.1.4 Research Questions

- Can a data-driven strategy based on machine learning algorithms identify mass imbalance in wind turbines accurately?
- How does the proposed method compare to the existing approaches for mass imbalance detection on the basis of cost-effectiveness, accuracy and speed?
- Is the suggested method applicable to different types of wind turbines and diverse environmental conditions?
- Can the suggested method identify mass imbalance in real-time, allowing for early identification and avoidance of future turbine damage?
- How can the suggested solution for automatic mass imbalance detection and alerting be incorporated into a wind farm's current Supervisory Control and Data Acquisition (SCADA)?

2 State of the art

2.1 Wind turbine technology / Condition Monitoring

Wind turbine machines are so complex that can undergo a lot of issues particularly in wind turbine rotor blade that includes aerodynamic imbalance and Mass Imbalance (MI). Condition Monitoring (CM) is the process of monitoring the components of a wind turbine to discover changes in operation that may indicate the development of a malfunction. It is an essential part of wind turbine maintenance and operation strategy and it is obvious that detecting errors before they occur through strong Condition monitoring should result in considerable reductions in Operation and Maintenance (O&M) expenditures of the wind turbine [5]. Condition monitoring is the practice of extracting and analysing the data from sensors on the wind turbine operational state to detect faults or anomalies that could be responsible for low performance of the wind turbines. The main causes of failure in wind turbine blades are manufacturing errors and damage, both of which are subject to environmental influences, with some kinds of local damage degrading structural performance and propagating significantly [3].

- Wind energy industry experts utilizes condition monitoring for their predictive maintenance thereby allowing them to conduct replacements or timely repair which avoids long downtime and reduce maintenance cost.
- The figure 2.1 presents a breakdown of the wind turbine maintenance techniques. Condition-based maintenance is frequently used to minimize equipment failure or breakdown as well as to lower failure rates, which promotes high equipment dependability. In this sense, condition-based maintenance can be used to the maintenance of wind turbine blades

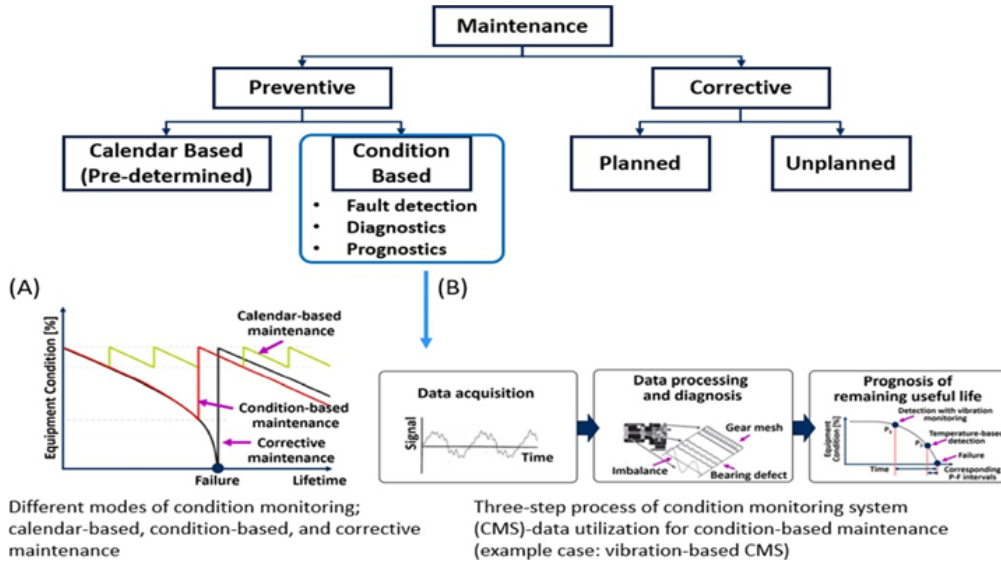


Figure 2.1: Condition Monitoring Process[3]

According to Fig 2.1, thorough condition-based maintenance is a process of fault detection through data collecting, data diagnostics, and remaining lifetime prognostics [3]. The study [25] reveals that O&M costs typically account for 20% to 25% of the total Levelized Cost of Electricity (LCOE) of current wind power systems and that the LCOE in onshore projects has decreased by 45% while in offshore it has decreased by 28%. Furthermore, the O&M costs of onshore projects have declined by 52% while in the case of offshore projects, it has declined by 45%. This study highlights the importance of limiting the maximum faults in offshore wind turbines as their maintenance is more complex. Though this study provides a comprehensive review of recent research on wind farm maintenance, it does not discuss the potential impact of emerging technologies such as machine learning on wind turbine maintenance strategies.

In wind energy industry, there are many types of condition monitoring strategies are used such as acoustic emission monitoring, vibration analysis, oil analysis, power performance monitoring and temperature monitoring [5]. Vibration analysis is one of the important and widely used condition monitoring techniques for wind turbines. Major wind turbine components such as rotor, generator and gearbox possess strong vibration when the turbine undergoes any faults. As a part of condition monitoring, the imbalanced rotor produces higher vibration levels that can be measured and analyse it and identify the issue that causes rotor imbalance i.e., Aerodynamic imbalance or mass imbalance of the rotor depending upon the frequency pattern of the vibration signal. Acoustic emission monitoring is another technique used for wind turbine condition monitoring. It detects high frequency sound waves generated by the wind turbine components during its operational state and this technique is used for fault detection such as fractures, cracks

and other defects that can lead to dangerous failures. Specifically, an unbalanced rotor produces higher level of acoustic emissions that can be detected using this technique. Condition monitoring technique such as temperature monitoring involves measuring the temperature levels of the critical turbine components such as gearbox, generator and bearings which can reveal issues like misalignment, bearing wear and other issues that leads to higher repair cost. In other hand, the power performance monitoring measures and analyse various parameters such as rotor speed, turbine's power output curve, blade pitch and wind speed to evaluate the efficiency and power output of the wind turbine. This approach helps in identify issues like blade mass imbalance, aerodynamic imbalance, and other performance related issues.

2.2 Mass Imbalance in wind turbines

One of the critical issues that affects the wind turbine performance and safety is mass imbalance. Wind turbine blade mass imbalance can be caused by manufacturing and construction faults, massive blade repair, fluid inclusions in the texture of the blades, variable static moments within a blade set, rotor division mistake, fatigue, corrosion, and icing. Because of the harmonic effects of the centre of mass being shifted from the rotor plane axis, the rotor mass imbalance generates a vibration on the generator shaft [26]. Wind turbines consist of various components which combined to generate electricity from the wind resource. The components such as rotor blades, hub, generator, gearbox, yaw systems, nacelle, tower etc which each plays a vital role in the energy production. In this section we will discuss about the turbine components which is highly impacted by the rotor mass imbalance of the turbine.

2.2.1 Rotor Blades

The rotor blades are the three (typically three) long, thin blades that connect to the nacelle hub. These blades are intended to convert the kinetic energy in the wind as it passes into rotational energy. As of 2021, the largest wind turbines being constructed in the world are 15MW turbines. The rotor blades of these turbines are little over 115m long. A 15MW wind turbine's blade tips sweep through the air at about 230 mph while operating at regular operational rates! The rotor blades play a very important role in detecting the mass imbalance in the wind turbine rotor. As the blades rotate, having any mass imbalance in the rotor blades will cause the blades to oscillate or vibrate. The generated vibration can be measured by the sensors mounted on the rotor blades which can measure the frequency of the vibrations. The abnormal vibration pattern i.e.,1p (rotational frequency) for the mass imbalance can indicate the mass imbalance fault in the wind turbine rotor[27]

2.2.2 Hub

The wind turbine hub also plays a role in detecting rotor mass imbalance. The hub is designed to balance the blade weight and ensure that the turbine rotor is symmetrical. One of the most crucial components of a wind turbine is the rotor hub, which joins the blades to the main shaft of the turbine and ultimately to the rest of the drive train, which transfers mechanical rotational power from the rotor hub to the generator.

2.2.3 Generator

The gearbox's high-speed output shaft contains rotational energy, which the turbine generator converts into an electrical current. An electric current is formed (or "induced") in a coil of wire when a magnet moves past it, according to the electrical theory of electromagnetic induction. The two main components of the generator are the stator as well as rotor. All the rotating pieces make up the rotor, while all the stationary ones make up the stator. Both strategies get the same outcome: an electric current is produced at the coils' output. Some systems employ rotating magnets against static coils of wire, and other systems use rotating coils of wire against static magnets [27]

2.2.4 Nacelle

The nacelle is the wind turbine's 'head,' and it is situated on top of the support tower. The rotor blade assembly is linked to the nacelle's front. A conventional 2MW onshore wind turbine assembly's nacelle weighs roughly 72 tons. It consists of five major components such as Gearbox, Generator, aerodynamic braking system, mechanical braking system and electrical power transmission systems. Nacelle's vibration during rotor imbalance is one of the important features for detection [27]

Many approaches for early detection of blade imbalance have been developed, the majority of which involve specific methodologies to extract the fault characteristics included in the vibration signal, generator current, or other factors. Various research used various approaches to analyse the stator current or rotor current in frequency domain, such as derivation, Hilbert envelope demodulation, dq coordinate transformation, order tracking analysis, and so on, which can enhance the fault features while decreasing the influence of fundamental frequency. Some defined the distinctions between aerodynamic imbalance and rotor mass imbalance. They were able to tell them apart by using various harmonics in the rotor speed frequency spectrum. Unfortunately, the outcome was only proven in constant wind circumstances. Other scholars investigated the link between blade imbalance and wind turbine vibration using finite element analysis or constructing a computational model [28]

In [29] the author developed an algorithm for aerodynamic and mass imbalance estimation from vibrational measurements. The developed method uses mathematical

model to connect the load caused by rotor imbalances to the resulting vibrations and inverse problem of calculating aerodynamic and mass imbalances from the vibrational data solved using nonlinear regularization theory. However, it is only evaluated with numerical examples and to validate its performance, the testing of real-world application with field measurements is necessary. Addition to that, the algorithm's accuracy depends on correct initial guess of the mass imbalance value. It should be noted that Tikhonov nonlinear function has a possibility to have several local minima where the iteration might stick, so it will be difficult to find an optimal initial value.

2.3 Data Driven Approaches for Mass Imbalance Detection:

In recent years, as a part of technology development, machine learning plays a powerful technique to leverage the business model and been increasingly applied in wind energy systems and particularly to detect rotor mass imbalance of the turbine. Much research has been made and still there is lot of research have been going along with the development of AI to improve the fault detection in an early phase to plan the predictive maintenance strategies, but the effective fault detection method has not been yet developed. The few important research that developed a data driven approaches for mass imbalance detection are to discuss. Stetco et al [5] provides a comprehensive review of the application of machine learning techniques in condition monitoring. The authors classify the machine learning models by typical steps including data sources, feature selection, feature extraction, model selection which includes classification and regression, validation and decision making. According to the study, the majority of the models employ simulated or SCADA data, with roughly two thirds of the methods using classification method and the remaining ones employing regression[5] They identified the need for further research in the area, especially in addressing the issue of imbalanced data and developing the interpretable models.

One study approached a method using Support Vector Machine (SVM) to detect mass imbalance using estimated rotor speed through a combination of electrical quantities (currents and voltages) [30]. The proposed method showed satisfactory accuracy scores in identifying various levels of imbalance from the SVM classes in the study of multi-class imbalance problem. The method must be validated using more real-world data since it was only tested on simulated data.

Another study proposed a method using convolutional neural networks to detect mass imbalance in a 1.5MW turbine [31]. They used estimated rotor speed as an input feature for the multi-classification problem. The study showed a proposed method achieving high prediction accuracy for different mass imbalance levels. However, this method requires large amount of data, and it was evaluated on a single turbine model, and further testing on a broader range of models is needed.

In a simplified model, another study proposed the use of support vector machines to detect imbalances in the rotor of the wind turbines [9]. This model used mass weight on the shaft, variable torque, and harmonic forces as a feature for the prediction. The proposed work achieved reasonable high accuracy for predicting the different mass imbalance level, but the study was limited to a simplified model and further testing on real world data is needed.

In [26] they proposed a method where they have used data augmentation using deep learning. The Convolutional Neural Network (CNN) was proposed to detect mass imbalance in the wind turbine rotor for a multiclassification problem and used input feature as the estimated rotor speed. A 1.5MW turbine model was created using the Turbulence Simulator (TURBSIM), Simulink and Fatigue, Aerodynamics, Structures, and Turbulence (FAST) software. The trained model was validated under various wind speeds ranges between 14.5 to 24.5m/s with turbulence intensities. However, the lower wind speed level such as 3 to 12m/s that are prevalent as it helps to differentiate the mass and aerodynamic imbalances. The proposed method showed an improvement in the model performance using data augmentation and fusion techniques combined with CNN architecture.

A deep learning technique for blade imbalance fault detection caused by ice accretion was proposed [32] Long Short Term Memory (LSTM) neural network was used along with an attention mechanism to extract the fault signal characteristics. The results using simulation shows that the proposed method could detect the imbalance fault with over 98% accuracy. The difficulties with the conventional mathematical technique are addressed in this paper, which provides a potential solution.

The fault tolerance model to reduce the failures is discussed in paper[33] which assess the rising complexity of computerized systems as computer technology advances, resulting in an increase in embedded systems that are not recognized or regarded as computers by people. Terms like ubiquitous computing, pervasive computing, and ambient intelligence reflect this approach. While many embedded systems aid humans or provide pleasure, there is an increasing number of essential applications where system faults could endanger individuals or cause considerable harm. Researchers have long been interested in designing robust and fault-tolerant systems to reduce the likelihood of failures. Depending on the application and study goals, many system models have been employed to analyze fault tolerance. The work proposes a formal fault modeling framework and a method for behavioral analysis under specific fault assumptions in order to contribute to the development of fault-tolerant avionics systems. The framework allows for the investigation of fault tolerance qualities in existing systems, making it easier to validate design ideas. The study does not intend to provide specific design ideas or strategies for developing fault-tolerant systems[33].

2.4 Feature Engineering Techniques

Feature engineering technique is important and crucial steps in any machine learning model. The raw data that extracts from database has lots of noise and sometimes the raw data information is not sufficient to train a good machine learning model. After the data extraction and data cleaning, the feature engineering techniques will be applied to transform the raw data into meaningful data for the model training. There are many techniques that are used and showed a good improvement in the model's performance and particularly to detect mass imbalance in wind turbine blades there are domain specific techniques such as frequency analysis contributes to most of the data driven approaches in mass imbalance detection. In Machine Learning (ML), there are various feature engineering techniques that can be used to process all kinds of problems.

2.4.1 Feature Scaling

Normalization: In data pre-processing, Normalization is an important step in any machine learning application. During the data preparation process, normalization is a scaling technique used in machine learning to change the values of numerical columns in a input dataset to use a standard scale[34]. It normally ranges between 0 to 1 and it can be applied in cases whenever the data distribution is not Gaussian. It usually affected if the data consist of outliers. During mass imbalance detection, sometimes the input data will have different range across the features. The normalization technique helps to scale down all the features to one, so the model's performance gets increased. **Standardization:** Another scaling approach is standardization, in which the values are centred around the mean and have a single standard deviation. The distribution that results has a unit standard deviation and the parameter's mean changes to zero to be the end result. This technique can be applied if data distribution follows Gaussian. It is less prone to outliers and preserves the feature relationship [35]

2.4.2 Feature selection

Feature selection is another technique used to eliminate the less impact features with respect to output so that the complexity of the model gets reduced when training. It consists of three classes such as filtering. Wrapper and embedded method.

Filtering

Filtering strategies eliminate features that are unlikely to be beneficial for the model. For instance, one can determine the correlation or mutual information between every feature and the response variable and then filter out characteristics that fall under a certain threshold. Examples of these strategies for text characteristics are discussed

in Chapter 3. Filtering approaches are substantially less expensive than the wrapper methods mentioned below, but they do not take the model into consideration. This may prevent them from selecting the proper model attributes. Prefiltering should be done cautiously so that beneficial characteristics are not accidentally removed before they reach the model training stage[36].

Wrapper Method

These strategies are costly, but they allow you to experiment with subsets of features, ensuring that you won't mistakenly prune away characteristics that are ineffective on their own but valuable when combined. The wrapper technique considers the model to be a black box that delivers a quality score for a suggested feature subset. A different strategy is used to progressively refine the subset [36].

Embedded Method

These algorithms select features during the model training phase. A decision tree, for example, does feature selection intrinsically because it chooses one feature on which to divide the tree at each training phase. Another instance is the L1 regularizer, that may be introduced to any linear model's training objective. The L1 regularizer favours models with fewer features rather than many features, hence it is also known as a sparsity constraint on the model. As part of the model training process, embedded approaches include feature selection. Certainly, it is not as effective as wrapper approaches, but they are far less costly [36].

2.4.3 Domain Specific Feature Engineering

To detect mass imbalance in wind turbine, the additional domain specific feature engineering strategies should be applied to solve the problem. This will help to extract the specific information needed to detect mass imbalance in the rotor blades. It includes time domain analysis, frequency domain analysis, also the wavelet analysis along with statistical analysis.

Time domain analysis

This method involves analysing the raw time specific vibrational data from turbine components such as nacelle, rotor speed and wind turbine tower. This method can be used to extract useful features such as mean, variance, standard deviation, skewness from the raw vibrational data that represent the overall properties of the vibrational signal. However, the time domain data will not have the useful feature to detect the

mass imbalance since the input vibrational data has frequency characteristics that will be accurate for prediction.

Frequency domain analysis

This method involves analysing the same vibrational signal generated by the wind turbine components but in frequency domain. It used to extract features such as frequency, amplitude, and frequency spectrum of the signal. To detect mass imbalance, the 1p rotational frequency is the fault signature and thus it can be extracted and used for training the model.

Statistical analysis

“Numbers never lie” is the best way to explain about the effectiveness of the statistical analysis. This method involves analysing the statistical properties of the vibrational signal. Mean and standard deviation values are very important features to represent the raw vibrational data and the machine learning model can be able to learn the patterns effectively if the statistical properties of the raw data are given. The auto correlation and cross correlation features from the signal can also be used for the detection.

Hybrid techniques

This method combines one or more feature engineering techniques to extract useful features from the vibrational data. For instance, wavelet analysis combine with statistical analysis or time domain combines with frequency domain analysis makes powerful technique to detect mass imbalance in wind turbines. The paper published in advances in computational intelligence [37] ,discusses the importance of machine learning for wind turbine fault detection. The proposed method allows autonomous learning to predict the component failures in wind turbines. The work compares the simulated failures with traditional techniques such as frequency analysis with SVM and K Nearest Neighbour (KNN) methodologies. The result show that implementing these techniques allows foreseeing a breakdown and reduces downtime and costs. In [38], a spectral technique is proposed for detecting Wind turbine rotor imbalance using only the rotor speed signal. The 1p frequency i.e., Rotational frequency signature was used to indicate the presence of mass imbalance, and multiple 1p side-bands around 3p and its harmonics were used to indicate the presence of aerodynamic imbalance. The result showed that without installing new sensors, the mass imbalance can be detected in high accuracy using the vibrational signal frequency of the existing components.

2.4.4 Overview of the Thesis

The current work attempts to build a data-driven approach to predicting mass imbalance in wind turbines. Section 1 of the paper gives an introduction of the trends and significance of wind energy, followed by a comprehensive assessment of the state-of-the-art methods for mass imbalance detection utilizing both traditional and data-driven methodologies. The findings and limitations of each strategy are explored in depth in section 2. Section 3 contains in-depth information about the work's cutting-edge technology. The tools and technologies used for data preparation, feature engineering, and machine learning are covered in this section. Data preparation entails cleaning, filtering, and scaling of the data. The extraction of features from input data that can aid in the prediction of mass imbalance in wind turbines is the goal of feature engineering techniques. The section also goes over the machine learning algorithms that are used to predict mass imbalances. Section 4 describes the proposed approach's implementation. The experimental setup, including the choice of the data set, the preparation of the data, and the application of the machine learning model, is described in detail in this section. The performance of the proposed method is evaluated using model evaluation criteria, which are also described in this section. The findings of the present investigation are presented in sections 5 and 6, respectively. The outcomes show that the suggested method can correctly forecast the mass imbalance in wind turbines. These sections also cover the suggested approach's drawbacks and potential future applications.

Method	Description	Pros and Cons
Vibration analysis	Measuring and analyzing vibration data to detect imbalance.	Non-invasive, can detect other problems. Requires baseline data, subjective analysis, not real-time.
Acoustic analysis	Analyzing sound data to detect imbalance.	Non-invasive, can detect other problems. Sensitive to background noise, requires baseline data.
Drive train torque	Measuring torque on the drive train to detect imbalance.	Can detect specific causes of imbalance. Invasive, can be expensive, requires calibration.
Blade pitch	Adjusting blade pitch to counterbalance the turbine	Real-time adjustment, can improve performance. May not be enough to correct severe imbalance, affects power output.
Machine learning	Using data-driven algorithms to detect imbalance.	Real-time detection, can learn from data, customizable. Requires data collection and preprocessing, requires training.

Table 2.1: Pros and Cons of Different Methods for Detecting Imbalance

3 State of the technology

3.1 Mass Imbalance:

Mass Imbalance (MI) may appear due to variations in mass distribution or overall mass of the wind turbine blade. This fault applies extra torque to the rotor. Under ideal conditions, the three wind turbine blades are of identical quality [38]. However, in real-world circumstances, the mass of WT blades is unbalanced owing to a variety of causes.

- There are certain mass errors across the blades due to technical glitches in the manufacturing process
- Wind turbine blades will be corroded because it exposes to harsh environments for a long time due to turbine instalments in complex location.
- Also, during extreme weather conditions for an example dust or cold weather, the blades will be covered with ice or dust. If it accumulates to a certain level, then the mass imbalance may occur [32]
- Mass imbalance can also occur as a result of damage or wear and tear on wind turbine blades over time. This can be caused by lightning strikes, blade erosion, or weariness from repeated stress of the component.
- The implications of mass imbalance can be significant, resulting in decreased energy output in some cases, greater mechanical stress on wind turbine components, and serious safety issues due to large structure. As a result, it is critical to discover and treat this issue as soon as possible.
- Mass imbalance can create extra torque on the rotor system of the wind turbine, putting more strain on the blades and other components.
- In contrast, aerodynamic imbalance refers to a state in which the airflow across the turbine blades is not uniform, resulting in uneven forces and moments on the blades.

While both mass imbalance and aerodynamic imbalance can impair wind turbine performance, they are caused by different factors and necessitate different detection and mitigation approaches. In this work we have focused on mass imbalance detection in the wind turbine blades.

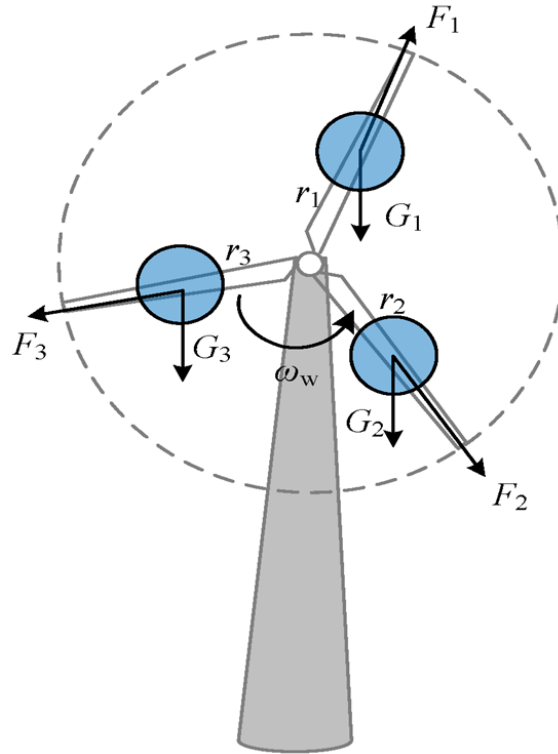


Figure 3.1: Wind Turbine Model[4]

Assume that this wind turbine is three bladed. Each blade may be equated to a mass block with a mass of m_i , and a distance of r_i from the hub centre. G_i and F_i represents each blade's gravity and centrifugal force, respectively. Each of these blades are equally affected by gravity and centrifugal force under typical circumstances. The rotating torque of the rotor is unaffected because the centrifugal force crosses the hub centre's axis perpendicularly. Since the three blades are geometrically symmetric, the torque produced by their combined gravitational attraction is zero while the rotor rotates at angular velocity ω_w [4].

The equation is: $m_1 g r_1 \sin(\omega t) + m_2 g r_2 \sin(23\pi + \omega t) + m_3 g r_3 \sin(43\pi + \omega t) = 0$ (3.1)

When a blade mass imbalance fault happens, the mass of one or more blades gets higher, which can be interpreted as the existence of mass imbalance m in a blade. The distance to the hub centre is R ; the block spins with the blade at an angular velocity, w ; as shown in Figure 3.2, the force experienced by the mass block during rotation consists mostly of gravity, mg , and centrifugal force, F_m [4]. The wind turbine transmission system is going to cause vibration along the main shaft due to gravity and centrifugal force. Because the tower's vertical stiffness is high, it will mostly generate periodic vibration of the blade and other structures in the horizontal direction [4]

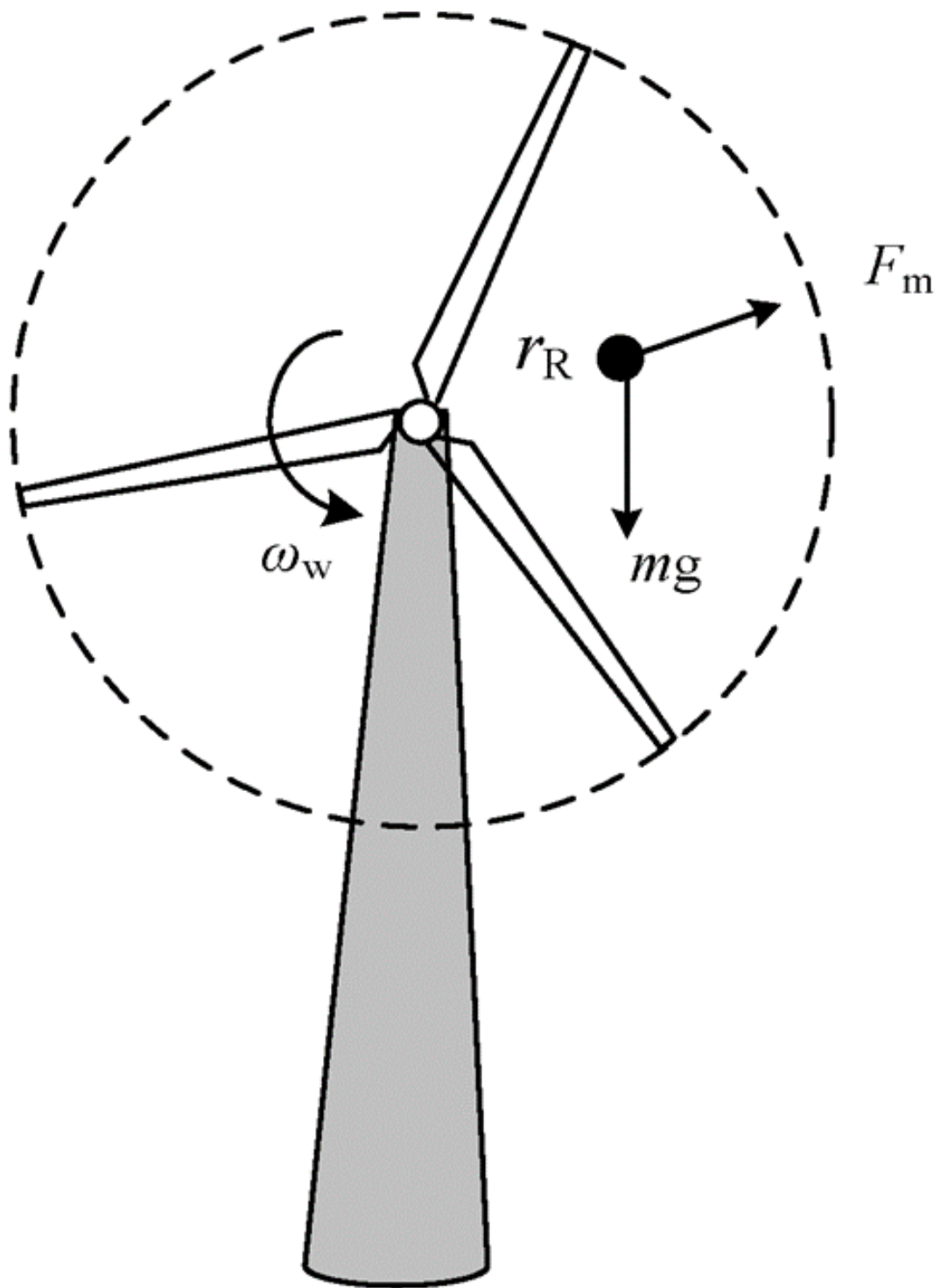


Figure 3.2: The Factor Affecting the Wind Turbine Model[4]

The equivalent mass block, m , spins at w with the blade, and the gravitational torque it generates can induce oscillations in the rotational speed of the main shaft. As seen in Figure 3.3, the mass imbalance accelerates the rotational speed of the main shaft during the downward revolution from top to bottom and decelerates the main shaft during the upward rotation from bottom to top. Figure 3.3 depicts a schematic illustration of a single WT blade's mass imbalance [4]

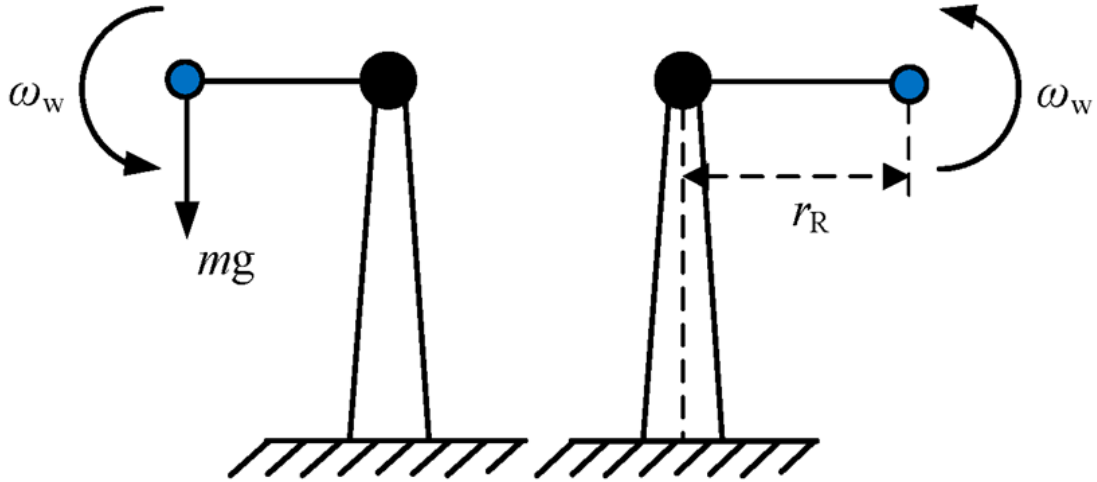


Figure 3.3: The Mass Imbalance Model[4]

According to [39], a significant number of blades were tested that were created under correct environmental circumstances and overseen by the certification organization Det Norske Veritas (DNV-Gl) to assure excellent quality, and the blade mass varied with a coefficient of variation of 2.1%. In this research, we have added a simulated mass imbalance to one of the blades with 2%,4%,6%,8%,10%,14% and 18% mass with respect to ideal mass of the wind turbine. The mass imbalance can be detect using two important features such as rotor speed and blade root bending moments addition to the variable wind speeds. To reflect the real-world scenario, the turbulence wind speed of 9.5% was introduced. By generating turbulence and shear, the change in wind with time and space over the rotor plane is represented. Turbulence is the uneven motion of the wind that causes both temporal and spatial variations in wind speed. Turbulence is a relatively random phenomena due to the extremely irregular motion of the wind and variations over multiple time and length scales, introducing a stochastic aspect to the wind conditions. The intensity of turbulent flow is defined as the ratio of the standard deviation of wind speed to the mean wind speed [40]

$$I = \frac{\sigma}{v} \quad (3.2)$$

where, I is the turbulence intensity, σ is the standard deviation of wind speed and v is the mean of wind speed.

3.2 Power Spectral Density

In machine learning life cycle, the feature engineering is the most important steps to work on. For mass imbalance detection in wind turbine, the time domain data is the initial format since SCADA stores sensor's data in time domain. In our approach, when using rotor speed feature, the vibrational data of the rotor speed should be measured. Converting time domain to frequency domain is the efficient way for the vibrational data to extract the frequency content of the signal. As we discussed, the specific frequency content such as 1p frequency (rotational frequency of the turbine) should be extracted. Depending upon the wind turbine nature, the 1p frequency will change but our wind turbine has 1p frequency at 0.14hz. So, if peak frequency of the signal present in rotational frequency, then we can conclude that the wind turbine is prone to mass imbalance. Another blade fault such as aerodynamic imbalance possess peak frequency at 3p frequency of the turbine.

Spectral analysis is the technique of finding the frequency elements of a continuous-time signal in the discrete-time domain. The majority of natural occurrences may be quantitatively described by random processes. As a result, the primary goal of spectral analysis is to determine the PSD of an arbitrary process. The power is the Fourier transform of a stationary random process's autocorrelation sequence. The PSD is a function that measures the distribution of total power as a function of frequency and so plays a critical role in the understanding of stationary random processes[41]. The power spectrum is also useful for detecting, tracking, and classifying periodic or narrowband phenomena buried in noise [42]. The PSD consist of various methods in mass imbalance detection using machine learning.

The periodogram was first used to look for underlying periodicities in sunspot data. The periodogram can be computed using one of two approaches. The indirect technique is one way. In this method, we first compute the autocorrelation sequence $r(k)$ from the data series $x(n)$ for $-(N-1) \leq k \leq (N-1)$, and then compute the Discrete Time Fourier Transform (DTFT)[42].

$$\text{Periodogram: } P^{\text{PER}}(f) = \sum_{k=-N+1}^{N-1} r^{[k]} e^{-j2\pi f k}$$

$$\text{Direct definition: } P^{\text{PER}}(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j2\pi f n} \right|^2 = \frac{1}{N} |X(f)|^2$$

$$\text{New frequencies: } D_f = \left\{ f_k : f_k = \frac{k}{N}, k = 0, 1, 2, \dots, (N-1) \right\}$$

$$\text{Periodogram with zero-padding: } x'[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ 0, & n \geq N \end{cases}$$

$$\text{New set of frequencies: } D'_f = \{f_k : f_k = \frac{k}{N}, k \in \{0, 1, 2, \dots, (N-1)\}\}$$

$$\text{Periodogram with zero-padding: } P^{\text{PER}}(f_k) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x'[n] e^{-j2\pi kn/N} \right|^2, \quad f_k \in D'_f$$

3.3 Welch's Method

Another estimator that makes use of the periodogram is the Welch technique [43]. It is based on the same concept as the Bartlett method of segmenting the data and calculating the average of their periodogram. The distinction is that the segments overlap, and the data within each segment is windowed. If a sequence $x(n)$ of length N is segmented into K subsequence, each with a length L and a D -sample overlap between the neighbouring subsequence, then [42]

$$N = L + D(K - 1)$$

where N is the total number of samples examined and K is the total number of subsequences[41]. It is worth noting that if there is not any overlap, $K = N/L$; if there is 50% overlap, $K = 2N/L - 1$. The i th subsequence is defined as follows:

$$x_i(n) = x(n + iD), \quad 0 \leq n \leq L - 1; \quad 0 \leq i \leq K - 1$$

and its periodogram is given by,

$$P^i(f) = \frac{1}{L} \left| \sum_{n=0}^{L-1} w(n) x_i(n) e^{-j2\pi fn} \right|^2$$

Because the samples $x(n)$ has been weighted by a non-rectangular window $w(n)$, $P_i(f)$ is the adjusted periodogram of the data; the Welch spectrum estimate is consequently provided by,

$$\hat{P}_{\text{Wel}}(f) = \frac{1}{KC} \sum_{i=1}^K \hat{P}^i(f)$$

where C is the normalization factor for power in the window function given by

$$C = \frac{1}{K} \sum_{n=0}^{K-1} w^2(n)$$

[41] Which has shown that the variance of the estimator is.

$$\text{Var}(\hat{P}_{\text{Wel}}(f)) \approx \begin{cases} \frac{1}{K} P^2(f) & \text{for no overlapping} \\ \frac{9}{8K} P^2(f) & \text{for 50\% overlapping and Bartlett window} \end{cases}$$

By permitting subsequence overlap, more subsequence's can be created than in the case of Bartlett's technique. As a result, the periodogram analysed by Welch's will have less variation than the periodogram evaluated by Bartlett [42]. In our approach, we have used Welch's PSD technique to extract 1p frequency of the rotor speed time varying signal to detect mass imbalance in the wind turbine since it reduces the variance of the periodogram and provide high resolution estimate of the PSD signal.

3.4 Machine Learning Algorithms

Once feature engineering of input data is completed, then we have an input data which is having useful features by eliminating the noise in the data to train machine learning model to prevent overfitting. In our mass imbalance detection problem, in the approach of using rotor speed and wind speed, the transformed input data is 1p frequency of the rotor speed and its according wind speed. The output features are the mass imbalance percentage class labels. The two primary responsibilities of machine learning for the identification of problems in wind turbines are the fault classification and the anomaly detection. This method enables early failure detection, aiding in the quick implementation of corrective actions, significantly raising the system's degree of dependability and security [37]. ML consist of two types such as supervised learning and unsupervised learning. The supervised learning consists of output class labels which is mapped to the input data whereas unsupervised learning does not have class labels and it classifies only using its input data. In supervised learning, there are two types such as regression and classification and unsupervised learning consist of clustering.

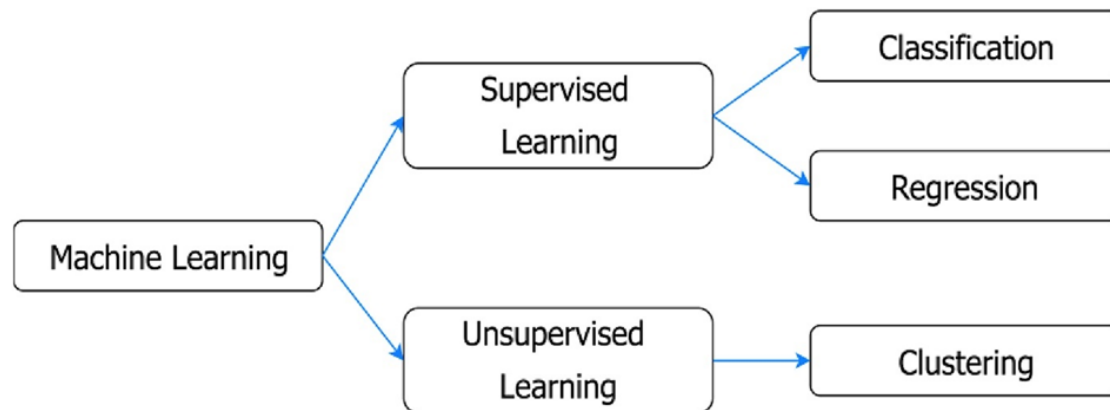


Figure 3.4: The Types of Machine Learning[5]

In wind energy context, both supervised and unsupervised methods of ML is used to solve the problem but widely used one among them is supervised learning method. In this method, we have two types such as regression and classification which will be

based on the nature of the dataset. While classification algorithms aim to predict a class label, regression techniques aim to predict a continuous variable. We evaluate the accuracy of models for classification and regression differently [43]. In wind turbines, the classification is used for the fault detection problem such as mass imbalance, aerodynamic imbalance in wind turbine blades and other components such as drive train. It classifies the output in binary or multi class labels which corresponds to the faults using various sensor components data. However, the regression is used to predict the continuous state of the component such as predicting the power output in kilowatts of the wind turbine based on rotor speed, wind speed, blade angles etc. so the two types of supervised machine learning is based on nature of the problem. The fig 3.5 represents the overview of how SCADA data is used for both classification and regression methods. In mass imbalance detection using rotor speed and wind speed as an input feature, the 1P frequency signal is extracted with its wind speed during the feature engineering techniques which is discussed before. And for blade root bending moments as an input feature, calculating the standard deviation of the bending moments and its corresponding mean wind speed. The nature of the problem is multi-classification since the multiple mass imbalance labels should be detected using machine learning. The next important step is choosing the ideal machine learning classification algorithm for our dataset. The fig 3.6 represents the entire machine learning algorithms used for both supervised and unsupervised methods.

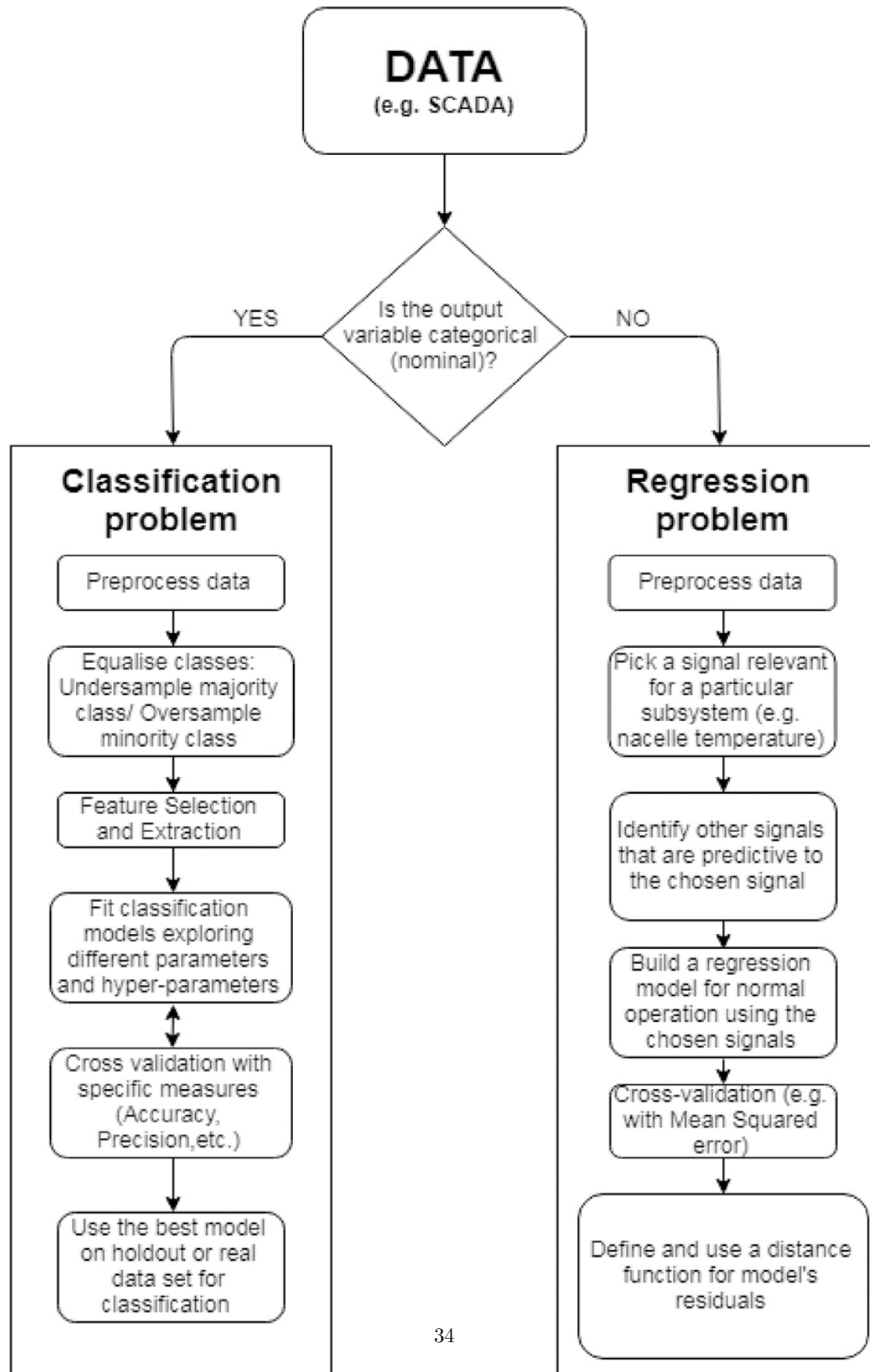


Figure 3.5: The Machine Learning Methodologies[5]

As you can see in fig below, there are many numbers of options to train the model and also it is time consuming and bad idea to try all of them. So, we should narrow down the list of algorithms depending upon 1. Objective of the project 2. Data availability and its type 3. Good performance on relevant metrics such as multi-classification 4. Explainability and Interpretability 5. Scalability and computational resources



Figure 3.6: Various Types of Machine Learning Algorithms[6]

3.4.1 Algorithm1 – Logistic Regression Classifier

Logistic regression is a supervised machine learning technique that is mostly used for classification problems, with the purpose of predicting the likelihood that an instance belongs to a specified class. Its term is logistic regression, and it is utilized for classification methods. It is called regression because it takes the output of the linear regression function as input and estimates the probability for the given class using a sigmoid function[7].

Terminologies Involved in Logistic Regression

Independent variables: The input features or predictor factors used to make predictions for the dependent variable. The dependent variable in a logistic regression model is the one that we are attempting to predict. The equation used to depict how the independent and dependent variables relate to one another is known as the logistic function. The logistic function converts the input variables into a probability value between 0 and 1, representing the possibility that the dependent variable will be 1 or 0[7]. Odds are the ratio of something happening to something not happening. It differs from probability in that probability is the ratio of anything happening to everything that could happen[7]. Log-odds: The natural logarithm of the chances is the log-odds, commonly known as the logit function. The log chances of the dependent variable are represented as a linear mixture of the independent factors and the intercept in logistic regression. The predicted parameters of the logistic regression model illustrate how the independent and dependent variables relate to one another. Intercept: In the logistic regression model, a constant factor that reflects the log chances when all independent variables are equal to zero. Maximum likelihood estimation: A technique for estimating the logistic regression model's coefficients that maximizes the likelihood of witnessing the data given the model[7]. The logistic regression model converts the continuous value output of the linear regression function into categorical values output by employing a sigmoid function, which transfers any real-valued collection of independent variables input into a value that ranges from 0 and 1.

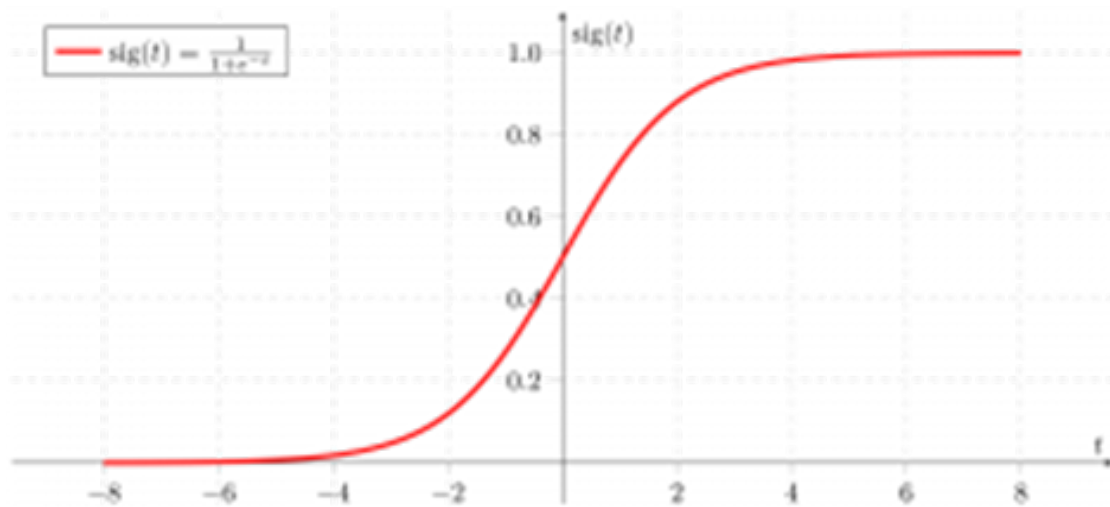


Figure 3.7: Logistic Regression Model[7]

Let the independent input features be.

$$X = \begin{pmatrix} x_{11} & x_{12} & & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & & x_{nm} \end{pmatrix}$$

$$Y = \begin{pmatrix} 0 \end{pmatrix} \text{ if Class is 1, and}$$

$$Y = \begin{pmatrix} 1 \end{pmatrix} \text{ if Class is 2.}$$

In this equation, x_i is the i th observation of

$$X, w_i = [w_1, w_2, w_3, \dots, w_m]$$

is the weights or Coefficient, and b is the bias factor, commonly referred to as the intercept[7]. This may be stated simply as the dot product of weight and bias.

$$Z = wX + b$$

The above discussed concept is linear regression and now we apply sigmoid function where input will be z and we should find the probability between the range 0 and 1 that is our dependent feature for the problem. Figure 3.7 shows sigmoid function that converts continuous variable into probability between 0 and 1.

- Sigma (z) towards 1 as z is infinity.
- Sigma (z) towards 0 as z is $-$ infinity.
- Sigma (z) bounded between 0 and 1 class.

The probability of any one class is measured by:

$$\frac{p(X; b, w)}{1 - p(X; b, w)}$$

Natural log is applied to the odds, then odds will be

$$\log \left(\frac{p(X; b, w)}{1 - p(X; b, w)} \right)$$

Then the equation of logistic regression will become,

$$\log \left(\frac{p(X; b, w)}{1 - p(X; b, w)} \right) = \beta_0 + \beta_1 X_1 + \dots + \beta_p X_p$$

The logistic regression's likelihood function states that the predicted probabilities will $p(X; b, w) = p(x)$ for $y = 1$, and for $y = 0$ the predicted probabilities would be $1 - p(X; b, w) = 1 - p(x)$.

Logistic regression makes the following assumptions

- Observations that are independent of one another: Each observation is independent of the other. This means that there is no association between any of the input variables.
- Binary dependent variables: It assumes that the dependent variable must be binary or dichotomous, which means that it can only have two values. Soft Max functions are utilized for more than two class such as multi-classification which is applied on our mass imbalance detection problem where we have more than two class.
- Linearity between independent variables and log odds: The connection between the independent variables and the dependent variable's log odds should be linear.
- No outliers: The dataset should have no outliers since logistic regression is prone to outliers.
- Large sample size: the number of samples must be sufficient.

There are two types of logistic regression such as binomial logistic regression and multinomial logistic regression. Binomial refers to binary class such as mass imbalance or no mass imbalance and multinomial refers to more than two class such as predicting different percentage of mass imbalance for our problem. When we consider multi-class, the Soft Max activation function should be used rather than using sigmoid function which is for binary class. Only when a decision threshold is introduced into the equation does logistic regression become a classification approach. The threshold value is an important feature of Logistic regression and is determined by the classification Problems itself.

3.4.2 Algorithm 2: KNN classifier

The k-Nearest Neighbours (KNN) algorithm is one of the simplest and popular machine learning algorithms used for classification problem. In our research, the mass imbalance detection consists of different output classes which can be detected by applying KNN algorithm. And also, the mass imbalance detection problem will hugely be affected by class imbalances when training such as one class is having more majority samples than other classes and so. This issue can be eliminated by implementing under sampling, oversampling and synthetic sampling techniques.

- Under sampling: To equalize the number of data points in each class, it entails removing data points from the majority class.
- Oversampling: It is the practice of reproducing data points from the minority class in order to equalize the quantity of data points in every class.

- Synthetic Minority Over-sampling Technique (SMOTE): It is an approach for producing synthetic data points for the minority class [44]

KNN learning algorithm is non-parametric. Unlike other learning algorithms, which enable discarding training data once the model is created, KNN preserves all training samples in memory. When a new, previously unseen example x arrives, the KNN algorithm chooses k training instances that are closest to x and delivers the majority of the label in classification or the average label in regression. A distance function determines the proximity of two samples. For instance, the Euclidean distance shown above is commonly utilized in practice. When the angle across two vectors is 0 degrees, the vectors point in the same direction, and cosine similarity equals 1. The cosine similarity is 0 if the vectors are orthogonal. The cosine similarity for vectors pointing in opposing directions is -1. To utilize cosine similarity as a distance measure, we must multiply it by 1. Chebyshev distance, Mahalanobis distance, and Hamming distance are some more prominent distance measures.

- The Euclidean distance is a distance metric used to determine the distance across two data points in n -dimensional space [45].
- A second distance metric used to estimate the distance across the two data points is the Manhattan distance. It calculates the distance by adding the absolute differences between the two positions' coordinates.
- The Minkowski distance generalizes the Euclidean and Manhattan distances. It can be expressed as the n th root of the sum of the absolute distances among the two points increased to the power of n [1]. It signifies the Manhattan distance when $n=1$ and the Euclidean distance when $n=2$ [45].

The researcher takes the decision on the metric of distance in addition to the value of k prior executing the algorithm.[find out] The value of k , which specifies the total number of nearest neighbours to consider, is likewise a hyper parameter that must be adjusted for any given situation. If k is too little, the method becomes susceptible to noise and outliers in the data, whereas if k gets too big, the algorithm underfits the data and produces unsatisfactory results. To determine the ideal value of k , cross-validation methods especially k -fold cross-validation could be utilized [45]. The few important points to consider when we use KNN classifier:

- One of the simplest supervised machine learning algorithms is K-Nearest Neighbour
- The K-NN method assumes commonality among the new case/data and current instances and places the new instance in the category that is most similar to the existing categories.

- The K-NN algorithm has been used for both regression and classification, however it is more commonly utilized for classification tasks.
- K-NN is a non-parametric method, which means it makes no assumptions about the underlying data.
- Because it maintains the dataset and subsequently acts on it during classification, this method is often referred to as a lazy learner since it is unable to immediately learn from the training set.

Despite all the positives, there are some disadvantages that makes KNN one step lower to other classification algorithms when dealing with mass imbalance detection.

- As the dataset increases, the computational complexity is high and therefore the KNN works slow for bigger datasets.
- Also, as the input features increases, then KNN struggles to predict the data points which ultimately leads to curse of dimensionality.
- Since it is dependent on distance, all features should be in same scale.
- Choosing the optimal K value is challenging when new data point is introduced.
- As we discussed earlier, KNN struggles when input data is imbalanced because it gives priority to majority class which will become a wrong prediction.
- KNN is very sensitive to outliers since its choses its neighbours according to distance [46]

3.4.3 Algorithm 3: Random Forest

One of the widely used supervised machine learning algorithm is random forest due to its ease of use, simplicity and high accuracy by combining multiple decision trees to classify the data. It is a powerful algorithm that can be used for both classification and regression. The Random Forest technique generates a forest of decision trees, with each tree based on a portion of the training data. The decision trees then categorize the data separately, and the outcome of each tree is merged to generate the final classification result. Random Forest is more accurate and less susceptible to over fitting than a single decision tree when numerous decision trees are combined [8]. To boost classification accuracy, the Random Forest method employs two essential techniques: bagging and random feature selection. Random forest employs bagging, often referred to as bootstrap aggregation, as an ensemble method. Bagging selects a random sample/random subset of the data set at random. As a result, each model is constructed using the samples (Bootstrap Samples) supplied by the Base Data using row sampling. This stage of row sampling

with replacement is referred to as the bootstrap. Every model is now trained individually, yielding outcomes. After merging the findings of all models, the final outcome depends on a majority vote. Aggregation is the process of integrating all of the findings and producing output based on majority voting[47] [8].

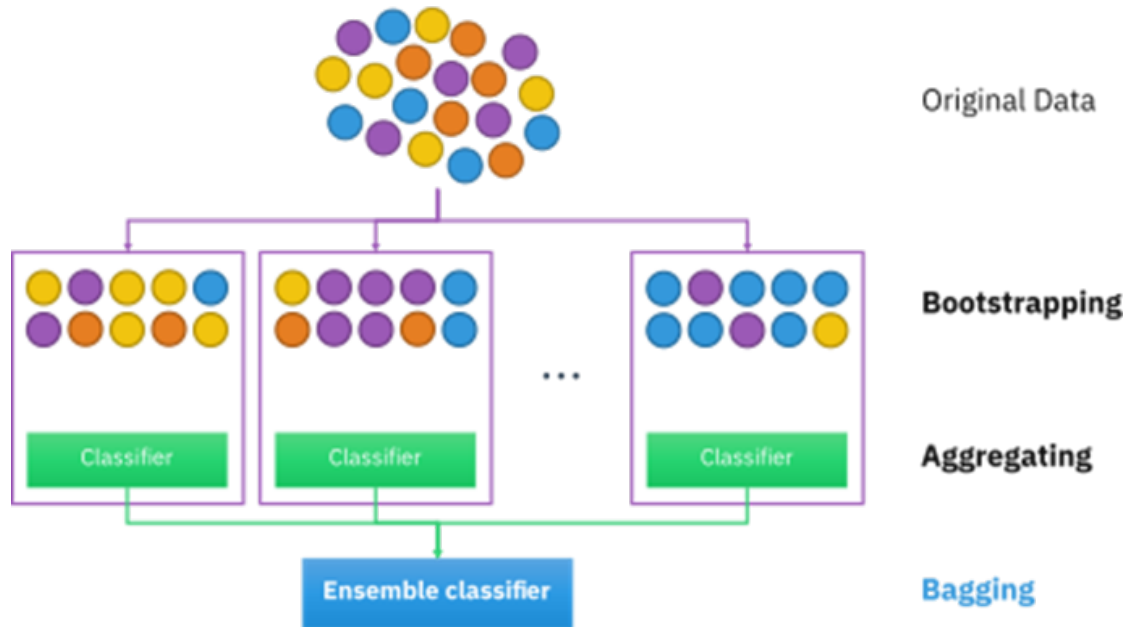


Figure 3.8: Ensemble Techniques[8]

Random feature selection, on the other hand, picks a subset of features at random for each decision tree. Random forest can lessen the danger of over-fitting and boost classification accuracy by merging the outputs of numerous decision trees constructed on various subsets of data and characteristics.

Feature Importance in Random Forest

Another important quality of random forest is they support feature importance technique. It helps to find which features are important for our prediction and the less important one can be eliminated after considering the domain aspects. In our mass imbalance detection problem, we have used rotor speed, blade root bending moments and variable wind speed as an input feature for the prediction and after the interpretation of feature importance technique, we could justify the importance for the prediction. This may be accomplished by examining how much impurity is reduced across all trees in the forest by tree nodes that employ those features. After training, it computes this score automatically for each feature and adjusts the findings such that the total importance equals one.

Hyper Parameters in Random Forest

The hyper parameters which is responsible for increasing the predictive power

- `n_estimators`: The number of trees built by the algorithm before averaging the predictions[47].
- `max_features`: The maximum number of features considered by a random forest while splitting a node.
- `mini_sample_leaf`: Counts the number of leaves needed to separate an internal node[47].
- `Criterion`: How should each tree's node be split? (Log Loss/Entropy/Gini impurity)
- `max_leaf_nodes`: The number of leaf nodes in each tree [8]

The hyper parameters which is responsible for increasing the predictive power

- `n_jobs`: this informs the engine how many processors it may utilize. If the value is 1, it can only utilize a single processor; if the value is -1, there is no limit.
- `random_state` controls the unpredictability of the sample. If the model has a fixed random state and is fed the same hyperparameters and training data, it will always deliver identical outcomes.
- `oob_score`: OOB stands for out of the bag. It is a random forest cross-validation approach. In this case, one-third of the sample is not utilized to train the data but rather to assess its performance[47]. These are referred to as out-of-bag samples [8]

Advantages

- Wind turbine mass imbalance detection is taken as a classification and regression problem, and the random forest technique can be employed well for both of this purpose.
- By employing a majority vote or averaging strategy to create the result, the algorithm helps to avoid over-fitting of the model, which is an important issue in wind turbine mass imbalance detection.
- In the context of wind turbines, the SCADA data might frequently have null or missing values owing to a variety of factors such as sensor malfunction, calibration error and the algorithm can manage such data without impacting its performance.

- The algorithm’s parallelization characteristic is especially helpful in the case of wind turbines since it allows for faster processing of massive volumes of data.
- The algorithm’s reliability is critical in identifying mass imbalances in wind turbines since it gets its output from the average answers given by a large number of trees, making it less vulnerable to noise and oscillations in the data.
- The algorithm’s variety is also effective in detecting mass imbalance since it facilitates the evaluation of various attributes while creating each decision tree.
- Due to the enormous number of attributes in wind turbine input data, the algorithm’s ability to limit the feature space by not evaluating all of the attributes is useful in such instances.

Disadvantages

- It is computationally expensive and needs more processing power when dealing with large number of datasets which will be especially the case of mass imbalance detection where we deal with large wind turbine datasets.
- The performance can be sensitive to the choice of hyper-parameters we choose during training.
- It is less interpretable which can be highly disadvantage for the mass imbalance detection in wind turbine where explainability is important [8]

3.4.4 Algorithm 4: Extra-trees Classifier

Another tree-based ensemble type algorithm is `extratreesclassifier`. It is built on decision tree as a base model, and it is almost similar to random forest algorithm with few important differences that make `extratreesclassifier` as a powerful and effective algorithm for mass imbalance classification problem. It is an acronym that stands for "Extremely Randomized Trees Classifier" and is used to do regression and classification problem. At training time, it constructs a large number of decision trees and outputs the class that is the mode of the classes (classification). Extra-Trees Classifier is similar to `RandomForest Classifier` since it both shares the ensemble properties, but it creates decision trees in a different method. When selecting a split point for a feature in Extra-Trees Classifier, the split point is picked totally at random from the range of potential values for that feature compared to the best split in random forest. This increases the unpredictability of the model, which can assist prevent over-fitting. The Extra-Trees Classifier is a meta-estimator that employs averaging to increase prediction accuracy and reduce model over-fitting. It fits a number of randomized decision trees (also known as extra trees) to different sub-samples of the dataset. It features a number of hyper-parameters that may be tweaked to enhance the model’s performance. `N_estimators`,

criteria, `max_depth`, `min_samples_split`, `min_samples_leaf`, and `max_features` are a few of these hyper-parameters which are almost the same as the random forest hyper-parameters [9]

Advantages

- It can handle binary and multi-class classification problems, which is suitable for the identification of the different levels of mass imbalance.
- The ability to work with imbalanced datasets helps in detecting the mass imbalance fault, which may occur at a relatively low frequency.
- Provides high accuracy, precision, recall, and F1 score, which are important metrics for identifying the mass imbalance fault with high confidence.
- Its ensemble method resists over-fitting and enhances generalization performance.
- The low variance of the algorithm helps in providing consistent performance across different datasets, which is important for mass imbalance detection.
- The algorithm is computationally efficient and can process large datasets in a reasonable amount of time, which is important for real-time mass imbalance detection applications.
- It can handle noisy or irrelevant features, which is important since the wind turbine dataset may contain irrelevant features that could affect the performance of other algorithms.

Disadvantages

- The better performance of the model depends on carefully choosing the model's various hyper-parameter values for the specific problem.
- Depending upon the length of datasets, there will be the bias and variance trade-off. To achieve the ideal balance between bias and variance, determining the ideal number of trees for a particular dataset might be crucial.

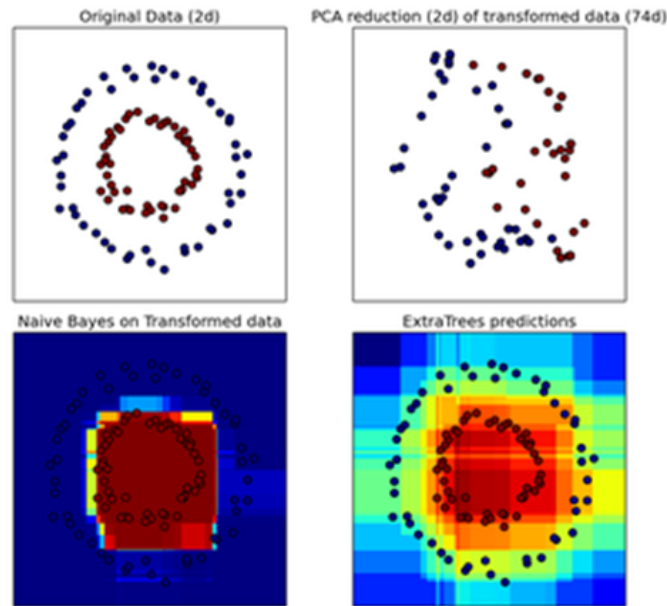


Figure 3.9: Performance of ExtratreesClassifier Model[9]

3.4.5 Algorithm 5: Support Vector Machine

Support Vector Machine (SVM) is a supervised machine learning approach that may help with both classification and regression problems. It seeks an ideal boundary known as a hyperplane between distinct classes. SVM performs advanced data transformations based on the kernel function specified, and it seeks to maximize the partition boundaries between your data points based on those kernel transformation. SVM attempts to identify a line that optimizes the separation among the two-class data set of 2-dimensional space points when there is a linear separation[10].

Objective of SVM

- The goal of SVM is to find a hyperplane in an n-dimensional space that optimizes the separation of data points to their actual classes.
- Support Vectors are data points that are nearest to the hyperplane and have the shortest distance to it.

As an example, The three points positioned on the scattered lines in the following diagram are the Support Vectors such as 2 blue and 1 green, and the separation hyperplane being the central red line[10]

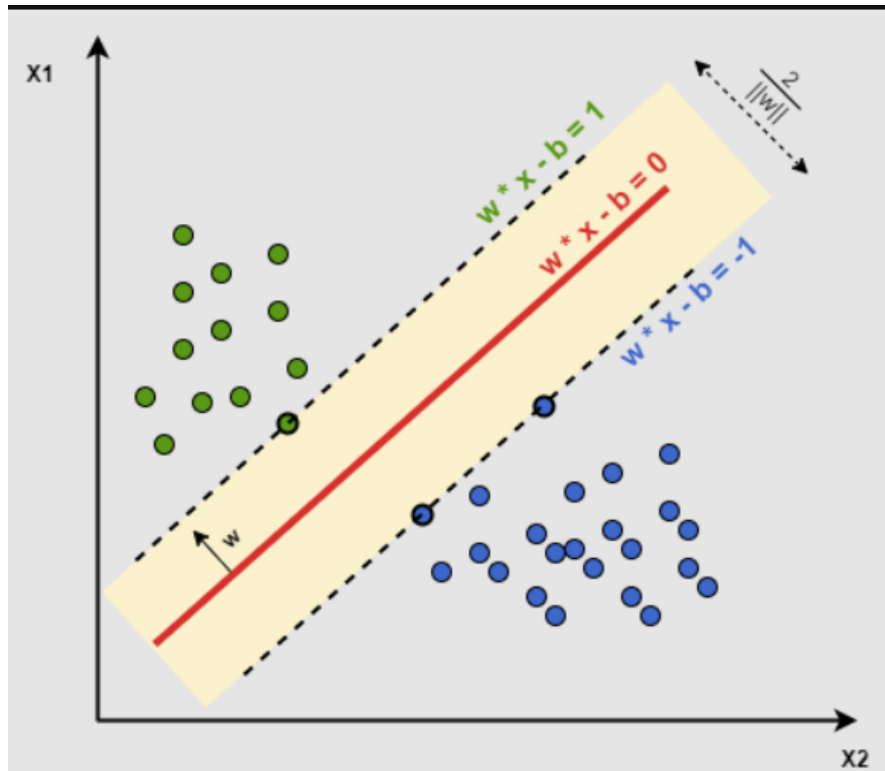


Figure 3.10: Support Vectors[10]

A kernel function is used in the computation of data point separation. There are several kernel functions, including Sigmoid, Polynomial, Gaussian, and Radial Basis Function. The smoothness and effectiveness of class separation are determined by these functions, and changing with these hyper-parameters may result in over-fitting or under-fitting of the model[48]. Normally, SVM supports binary classification problems by splitting the binary support vectors with its hyperplane. To handle multiclass problems the same technique should be applied by breaking down the multiclass problem into multiple binary class problems[48]. The following are some prominent ways for doing multi-classification on issue statements using SVM:

- One vs One approach
- One vs Rest approach
- Directed Acyclic Graph approach

One vs One Approach

This approach breaks down the multi-class problem into multiple binary classification problems. After applying this technique, the binary classifier per each pair of classes are obtained. It uses majority voting for final predictions along with the distance from the margin. The problem we face in this approach is to train many SVM models. Assume the mass imbalance problem having multi-class nature[10] For the s, t classifier:

- Positive samples are all the points in class s ($x_i : s \in y_i$)
- Negative samples: all the points in class t ($x_i : t \in y_i$)
- $f_{s,t}(x)$: the decision value of this classifier
- $f_{t,s}(x) = -f_{s,t}(x)$
- Prediction: $f(x) = \operatorname{argmax}_s (\sum_t f_{s,t}(x))$ [10]

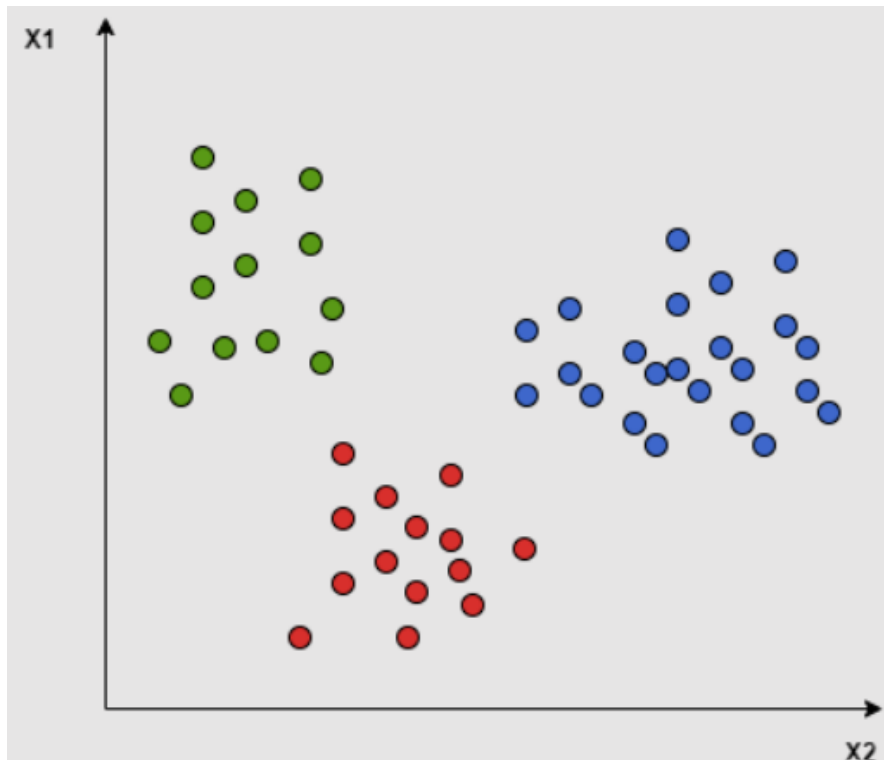


Figure 3.11: multiclass separation[10]

We require a hyperplane to separate every two classes in the One-to-One technique, ignoring the points of the third class. This signifies that the present split takes solely the points of the two classes into consideration. The fig represents the separation of

hyperplane for multi-class where red and blue line tries to maximize the separation only in between blue and red point classes only not the green point classes[48].

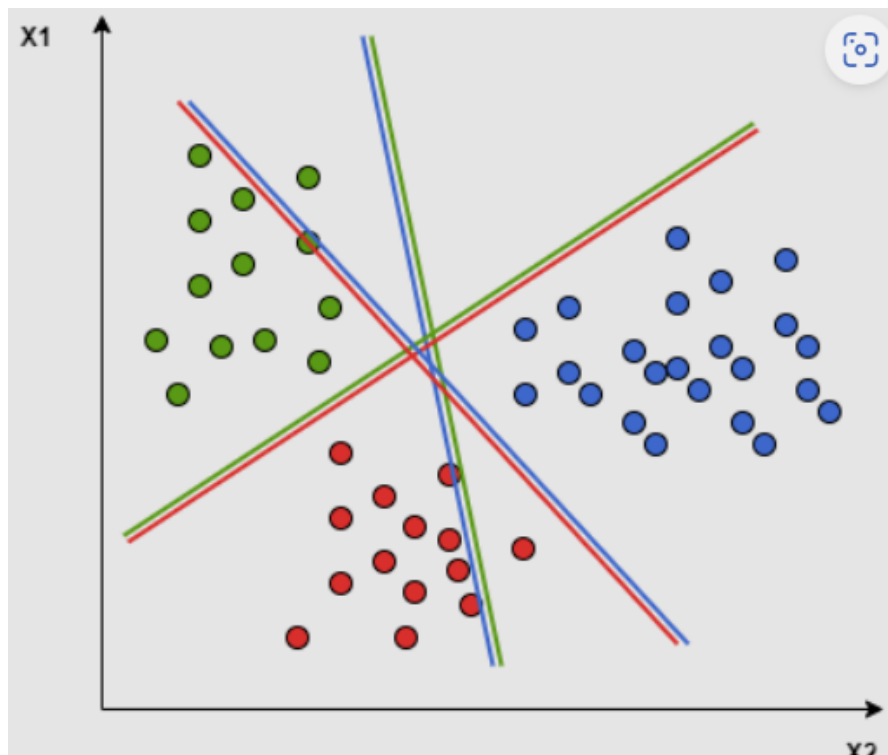


Figure 3.12: One vs One Approach[10]

One vs Rest Approach

We aim to find a hyperplane to split the classes in the One vs All strategy. This indicates that the separation considers all points and separates them into two groups, one for the points of one class and the other for all other points. In this work, N SVMs were utilized to learn binary classification problems. Each SVM is trained to learn a single class of output. SVM-1, for example, is trained to learn the class output equal to 1 versus the class output not equal to 1. Similarly, SVM-2 is trained to learn the class output equal to 2 versus the class output not equal to 2, and so on. This method allows for multi-class classification by dividing the challenge down into N binary classification tasks, with each SVM learning to identify one class from the others. There are a number of issues with training N number of SVM model using the One-vs-Rest approach that must be resolved. First off, as more classifiers must be trained, the OVR strategy's computing complexity rises as the number of classes does. This can result in longer training periods and more computing resource requirements, which in certain

circumstances may be a practical restriction. The OVA technique has the potential to produce uneven class distributions, especially if the amount of training samples for each class is not equal. In a mass imbalance dataset with 8 classes, for instance, if each class contains 120 training samples, then for every SVM trained using the OVR approach, one class will have 680 samples while the other class would only have 120 samples. As a result, the classifier could not have enough information to understand the characteristics that set that class apart from the others, which could result in poor performance on the class with fewer samples[48][10].

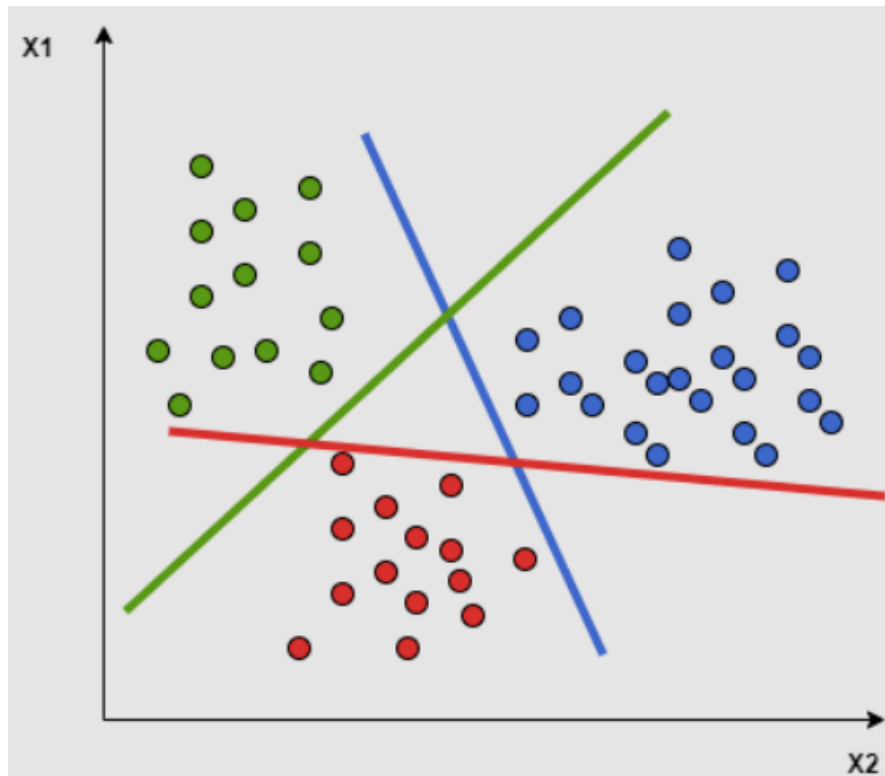


Figure 3.13: One vs Rest Approach[10]

Several ways have been suggested to tackle these issues, such as employing data augmentation methods to enhance the amount of the smaller classes or class weighting algorithms to award larger weights to the minority class while training. Furthermore, sophisticated approaches like the One-vs-One strategy and the Error-Correcting Output Codes strategy can be utilized as alternatives to the OVR strategy to train multi-class SVM classifiers. These solutions try to solve some of the shortcomings of the OVR strategy and may be better appropriate for particular types of datasets. Therefore, using the mentioned two methodologies, to categorize the data points from the L classes data set:

- The classifier in the One vs All technique can employ L SVMs.
- The classifier in the One vs One technique can employ $L(L-1)/2$ SVMs.

Directed Acyclic Graph

This strategy is more hierarchical in structure, and it attempts to overcome the difficulties of the One vs One and One vs All approaches. This is a graphical strategy in which we group the classes based on some logical grouping[10].

- Benefits: This strategy has fewer SVM trains than the OVA approach and lowers variation from the majority class, which is a concern with the OVA approach.
- challenge: If we have given the dataset in the form of distinct groups, we can directly use this strategy; however, if we do not supply the classes, then the challenge with this approach is identifying the logical grouping in the datasets[10]

Advantages of SVM

- SVM is a well-known binary classification technique that may be adapted to multiclass classification issues using approaches such as one-vs-all and one-vs-one.
- Because SVMs are less prone to over-fitting of the model, they are more resistant to noisy and high-dimensional data.
- With the application of kernel functions, SVMs can handle both linear and non-linear decision limits.
- SVMs have already been used effectively in a variety of real-world scenarios such as wind turbine fault diagnostics and prognostics including mass imbalance detection.

Disadvantages of SVM

- SVMs are computationally expensive to train the model on big datasets, especially when kernel functions are used.
- SVM effectiveness can be affected by the hyper-parameter selection of the model, such as kernel and regularization parameters.
- SVM may not be suitable for severely unbalanced datasets in which one class has considerably having majority than the other class.
- Because the decision boundary is often expressed as a complex function of the input variables of the data, SVM can be difficult to comprehend.
- SVMs are not designed to handle missing or partial number of data, and managing such scenarios may call for additional preprocessing of the data.

3.5 Model Validation

The above machine learning models are tuned by its individual hyper parameters to get the better performance of the model, but the model's performance should be validated to have the generalized performance score during testing. Therefore, such model validation technique is called cross validation where data will be splitted to multiple folds and do the train on each fold and average the performance of each fold and gives the generalized accuracy of the model. We usually have to split the input data into train, validation and test set. The train data is to train the model, validation set is to tune the hyper parameters and know the model's performance on new unseen test data and test data is to test the model's performance and it is known as hold-out validation. But there is a catch. When the input data is limited (few thousands or hundreds of samples) the splitting the data into three sets is not a good choice since each set will have only fewer samples which may impact the model's performance by not having enough data. Especially, the mass imbalance detection problem is having limited dataset after the intensive feature engineering techniques such as extracting 1p peak frequency of the rotor by averaging every 10minutes samples and calculating the standard deviation of blade root data for every 10minutes from millions of input samples. This issue will be highly eliminated by using cross validation techniques. A cross-validation is a frequent approach that might aid you when you don't have a good validation set to adjust your hyper-parameters. When there are few training instances, having both validation and test sets may be prohibitively expensive. You would want to train the model with more data. In this situation, you simply divide the data by training and test set. The training set is then cross validated to replicate a validation set. There are few types of widely used cross validation techniques which can be employed to our model such as

- K-fold cross-validation
- Hold-out cross-validation.
- Stratified k-fold cross-validation
- Leave-one-out cross-validation.

3.5.1 K-Fold Cross Validation

The whole dataset has been divided into k equal-sized sections with this approach, and each partition is referred to as a fold. It's called k-fold because it has k pieces, where k can be any integer such as 4,5,10 etc. One-fold is utilized for validation, while the remaining K-1 folds are used to train the model. This procedure is done k times until each fold is utilized once as a validation set and the other left outs as a training set [49] In our model, we tested out with five and ten fold cross validation and found five-fold

cross-validation is the best k value for our model. With five-fold cross-validation, your training data is randomly divided into five folds: F_1, F_2, \dots, F_5 . Each $F_k, k = 1, \dots, 5$ represents 20% of your training data. Then you train the following five different models [11] To train the first model, f_1 , all instances from folds F_2, F_3, F_4 , and F_5 are used as the training set, while every instance from fold F_1 is used as the validation set. The instances from folds F_1, F_3, F_4 , and F_5 are used to train the second model, f_2 , and the examples taken from F_2 are used as the validation set. You create models in this manner continuously, computing the value of the measure of interest on each validation set, from F_1 to F_5 [11].

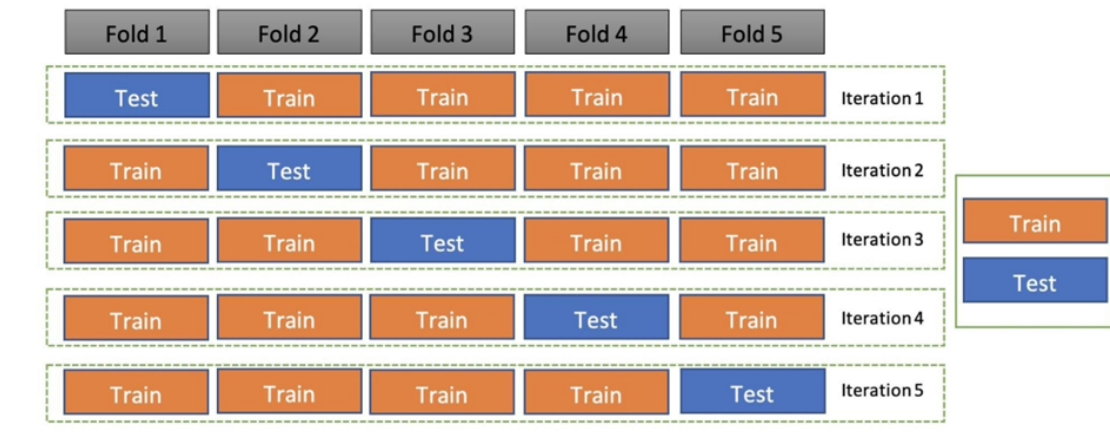


Figure 3.14: Cross Validation Techniques[11]

The k -fold cross validation is not performing well when input data consist of imbalanced datasets where the model fails to train effectively on each class. In mass imbalance detection problem, there may be an imbalanced dataset since it deals with multiple mass imbalance classes and choosing the k -fold is entirely depends on nature of the balance.

3.5.2 Hold Out Cross Validation

The whole dataset is randomly partitioned into a training set and a test set in holdout cross-validation. A good rule of thumb for data partitioning is to utilize almost 70% of the total dataset as a training set and the remaining 30% as a validation set or 80% for training and 20% for testing. Because the dataset is divided into only two sets, the model is trained on training set and test using test set [49] Also, the hold out validation can be splitted into three parts such as train, validation and test set where validation set is used to tune the hyper-parameters of the model. This validation set is splitted from the training set and it is unseen to the model so the performance on this set tells how well the model is generalized to the new unseen test data. The ratio of splitting will be

60% of training set, 20% of validation set (80% train split is further splitted by 60-20) and 20% for test set. It is highly effective when there are enough samples for each set.

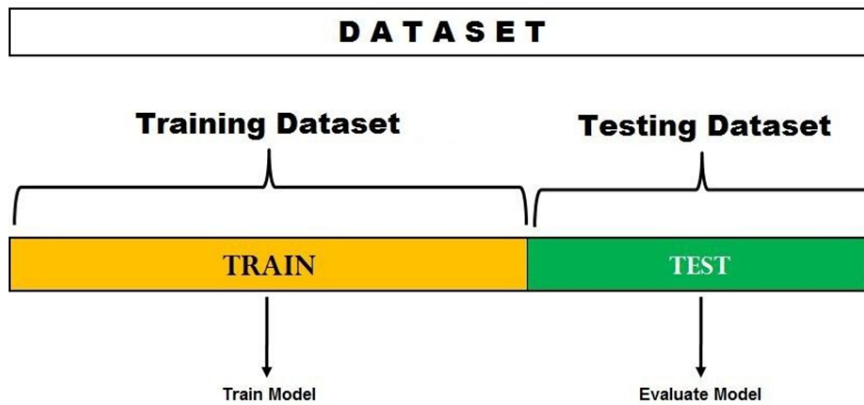


Figure 3.15: Hold Out Validation - Train/Test Split[11]

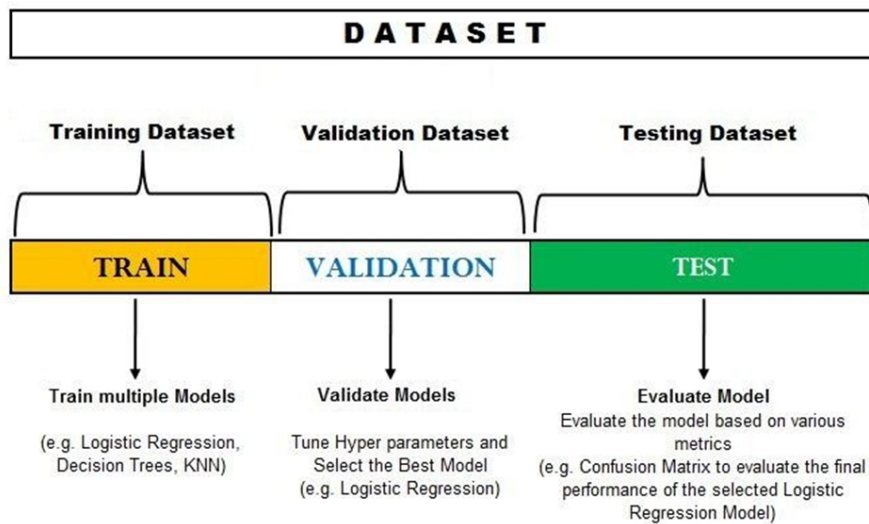


Figure 3.16: Hold Out Validation - Train/Validation/Test Split[11]

3.5.3 Stratified K-fold Cross Validation

To overcome the data imbalance issue in traditional K-fold cross validation, the stratified K-fold cross validation is introduced. This method ensures that every fold contains almost the same proportions of classes such that each fold will have an exact data representation and therefore the model's performance evaluation is effective among all classes in the data [49].

3.5.4 Leave One Out Cross Validation

As the name suggests, the model is validated by leaving one sample at a time and train the model on remaining samples and this process is done iterative to all the samples in the data. Since it leaves only one sample at a time, it is computationally expensive when the input data is very large [49].

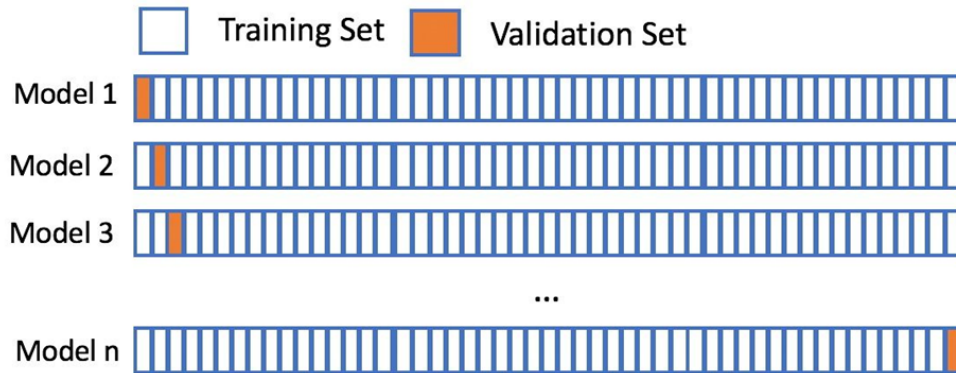


Figure 3.17: Leave One Out Cross Validation[11]

3.6 Hyperparameter Tuning

As we discussed about the hyper-parameters for the model in previous section, we then discuss about the techniques to perform hyper-parameter tuning. There are two types of techniques available such as random search and grid search to find the optimal parameters for the model to improve its performance.

3.6.1 Random search

As the name suggests, the Random search technique will search the best mentioned parameters randomly. Since it picks the parameters randomly, it is more efficient and faster method compared to grid search technique especially when there is many hyper-parameters to tune the model.

3.6.2 Grid search

Meanwhile, the grid search technique will search each and every combination of hyper-parameters for tuning instead of random pick. It is an effective method compared to random search which is not always find the optimal value at random, but grid search is highly computational expensive since it selects all the parameters to find the best combination while searching especially when there is many hyper-parameters.

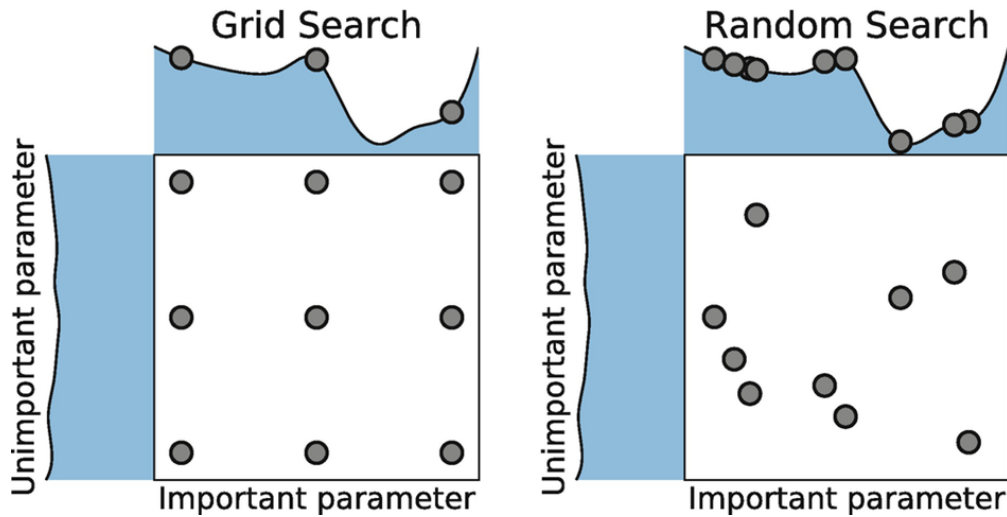


Figure 3.18: Grid Search vs Random Search[11]

In both techniques, cross validation is employed to validate the model performance for each combination of parameters. In mass imbalance detection techniques, we used K fold cross validation of $k=5$, so each fold out five folds will be assigned by every combination of hyper-parameters to find the optimal one when we use grid search, whereas for the random search, the randomly picked best combination of hyper-parameters will be employed.

3.7 Classification performance metrics:

Once the model is trained and tested with new unseen test data, now is the time to evaluate the model's performance. Since the mass imbalance detection problem is multi-classification in nature, we have to concentrate on classification metrics. There are various metrics to evaluate the performance of the classification model such as [12] 1. Confusion matrix 2. Accuracy 3. Precision score 4. Recall score. 5. F1 score

3.7.1 Confusion Matrix

The Confusion matrix is one of the most simple and straightforward metrics for determining the accuracy as well as the correctness of the model. It is utilized for classification problems where the result might be of binary or multi-classification classes. It is then comparing the predicted labels with the actual true labels and provides the summary of how well it actually classifies. It helps to measure the number of true positives, true negatives, false positives and false negatives.

- True positives (TP): True positives are when the actual class label is 1(True) and the predicted is also 1(True).Ex. The wind turbine is having a mass imbalance and the model also classifies that it has mass imbalance.
- True Negatives (TN): True negatives occur when the actual class of a data point is 0 (False) and the projected class is likewise 0 (False). Ex. The wind turbine NOT having mass imbalance and the model classifies that it has NO mass imbalance.
- False positives (FP): False positives occur when the actual class of a data point is 0 (False) while the projected class is 1 (True). Ex. The wind turbine NOT having mass imbalance, but the model classifies that it has mass imbalance.
- False Negatives (FN): False negatives occur when the actual class of a data point is 1 (True) while the projected class is 0 (False). Ex. The wind turbine is having mass imbalance, but the model classifies that it has NO mass imbalance.

3.7.2 Accuracy

In classification problems, accuracy is defined as the number of correct predictions produced by the model over all types of predictions made. Accuracy can be considered as a good metric when the input data is almost balanced.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.7.3 Precision Score

Precision is defined as the percentage of accurately classified positive events. It is the ratio of actual true positive class to the overall predicted positive classes.

$$Precision = \frac{TP}{TP + FP}$$

3.7.4 Recall score

It is the ratio of actual true positive class to the overall true classes. For mass imbalance detection, we have to obtain better recall score since the scenario like having mass imbalance in wind turbine and the predicted is not having no mass imbalance should be avoid mass imbalance in wind turbine and the predicted is not having no mass imbalance should be avoided [12]

$$Recall = \frac{TP}{TP + FN}$$

3.7.5 F1 score

F1 score: It is the harmonic mean of precision and recall. In multi-class classification model, the F1 score for each class can be calculated individually and then averaged to yield the model's overall F1 score. Based on the application, there are several methods for computing the average value.

- Micro-averaged F1 score: In this technique, F1 scores are computed separately for each of the class and then averaged using this formula:

$$F1_{micro} = \frac{2 \times TP_{total}}{2 \times TP_{total} + FP_{total} + FN_{total}}$$

- Macro-averaged F1 score: In this technique, F1 scores are computed individually for each of the class and then averaged using this formula:

$$F1_{macro} = \frac{1}{k} \sum_{i=1}^k F1_i$$

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

		Actual Class	
		1	0
Predicted Class	1	True Positive	False Positive
	0	False Negative	True Negative

Figure 3.19: Terminologies of Confusion Matrix[12]

3.8 Flask Framework

Once the machine learning model was built, it is necessary to create an application where the end user can make use of the machine learning model for various purpose such as predicting the mass imbalance in wind turbines. One of the popular and ease of use python web framework is called Flask. It helps to build web applications easily and also supports flexibility for building machine learning applications via Restful API. Flask is a WSGI framework. It refers to the Web Server Gateway Interface. Essentially, this is a method for web servers to route requests to web apps or frameworks. Flask runs on the WSGI external library and the Jinja2 template engine [13] The fig 3.20 represents the methodologies of flask application. Here, the trained machine learning model is exported to pickle file format and this file will be exposed to flask framework to get the prediction result. The front end is developed using front end framework such as HTML / CSS for the simple yet user friendly design. Once the user gives their input, the REST API call will be made to obtain the results from flask which is having the trained machine learning model as its back-end. Finally, the results obtained from the flask framework via REST API call will be shown at the front-end application [13].

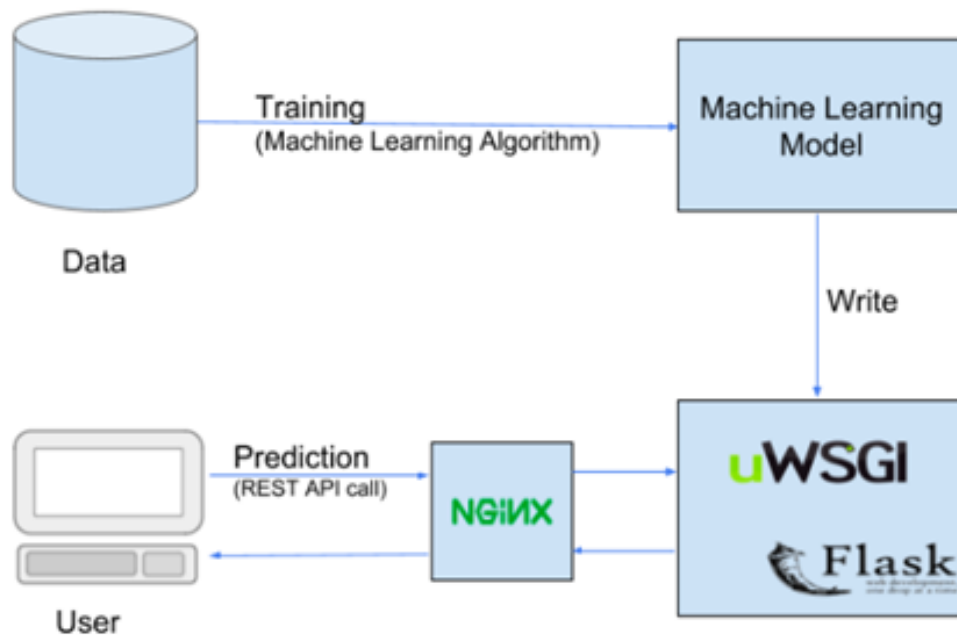


Figure 3.20: The Infrastructure of Flask Framework[13]

This project is divided into four sections:

- `model.py` — This file includes the machine learning model's code for predicting the mass imbalance in wind turbines.
- `app.py` — This file contains Flask APIs that accepts input SCADA data of AD8 via GUI or API requests, compute the anticipated value based on our model, and return it.
- `request.py` — This module calls the APIs provided in `app.py` and presents the results.
- `HTML/CSS` — This includes the HTML template and CSS style to allow the user to enter SCADA data in .csv file format and display the percentage of mass imbalance in wind turbine if exist.

Advantages

- Flask is a lightweight, flexible and ease of use framework which can be used by beginners to develop small to medium scale projects.
- It offers various extensions and libraries for the developers that makes easy to upscale the framework by adding new kinds of functionality.
- For machine learning applications, Flask framework is an excellent choice for building RESTful API's.
- It provides plenty of support and resources for building the applications.

Disadvantages

- Flask is not a good choice when we build more complex and larger applications because it is difficult to manage the workflow.
- It offers less out of the box functionality than frameworks in large scope ex. Django.

4 Implementation

The input data is simulated from Adwen AD8 wind turbine model, which is located at Fraunhofer IWES, Bremerhaven. The reason behind using the simulation data is to having output labels for the input features since the SCADA data doesn't have enough information on ground truth of the data. To train supervised machine learning, the input data must contain output class labels therefore, the input data is simulated from Adwen AD8 model with same attributes that makes the behaviour of input data as equal as SCADA data which helps model to learn more complex pattern and obtain better prediction on mass imbalances. The research work has done on two approaches.

4.1 Mass Imbalance Detection Techniques

4.1.1 Impact

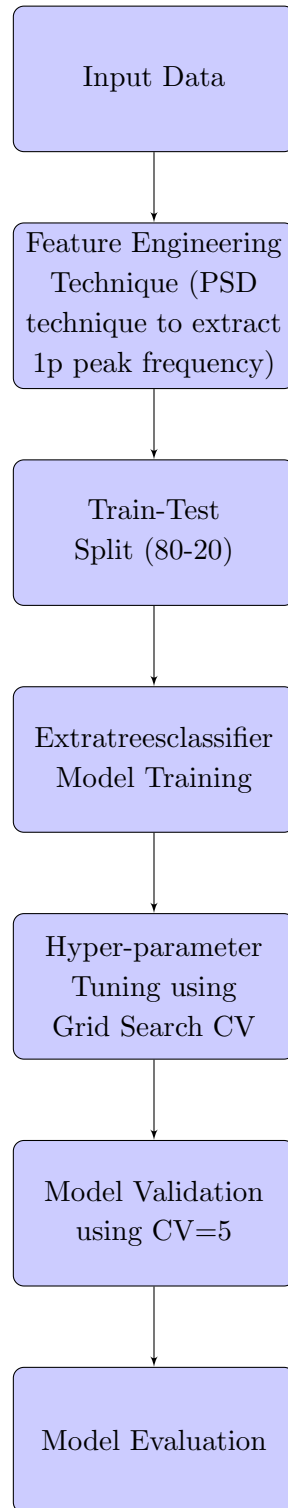
As we already know that mass imbalance is the relative variation of mass among the turbine blades due to manufacturing error and harsh environmental conditions, it has little to no effect on the power curve, but it modifies the loading and vibration pattern, placing an extra periodic load on the blades with a frequency equal to the frequency of rotation of the turbine on the tower and the drive train. These periodic stresses have a substantial impact on the fatigue life of the wind turbine components.

4.1.2 Detection

Mass imbalance is defined by the existence of periodic fluctuation in loads measured at the blade root and also in the rotor speed of the turbine.

The AD8 input data and feature engineering process is distinct in both techniques to get the most out of each feature, although the input data sampling frequency, wind speeds, and output class labels are the same throughout simulation modeling. The goal of this study is to detect mass imbalance in the AD8 wind turbine with high accuracy using SCADA test data using a trained machine learning model learned on simulation data, as they both have the same statistical features. The fig shows the overview of detection process for approach 1.

4 IMPLEMENTATION



4.2 Approach 1: Rotor Speed and Wind Speed

Feature	Description
Fault Signature	1p Peak frequency signal (rotational frequency) at 0.14hz
Input Features	Rotor speed (rpm) and Windspeed (m/s) Ranges from 4m/s to 20m/s with turbulence intensity 9.5%
Sampling Frequency	Fs=100Hz of 10 minutes period
Class Labels	Mass Imbalance levels – 0, 2%, 4%, 6%, 8%, 10%, 14%, and 18%
Feature Engineering Technique	Power spectral density Welch’s method is used to convert time to frequency domain and extract PSD values of 1p peak frequency as an input. (1p=0.14hz)

Table 4.1: Summary of Input Features, Class Labels, and Feature Engineering Technique for the Mass Imbalance Fault Signature Dataset for the Approach 1

The detection process consists of few parts such as The input simulation data from AD8 turbine model such as rotor speed in rpm and wind speed in m/s ranges from 12m/s to 20m/s with turbulence intensity of 9.5%. The quantity of input data samples is in millions since each mass imbalance percentage is measured against each different wind speeds and each simulation has 10 minutes of samples with sample frequency of 100 Hz. The fig 4.1 represents the 1minute time series representation of the rotor speed to have better visualization. For detection, we only use 10minutes samples as discussed earlier.



Figure 4.1: Time Series Signature of Rotor Speed AD8 Data

The input data is then extracted from the simulation model and performed the exploratory data analysis where we have checked whether the input data has any missing values and also the balancing nature of each mass imbalance output classes. The noise in the data is not exist because we have simulated in the nature where there is no class imbalance and no missing values. Then the million samples of input data are employed to perform the feature engineering techniques to transform the input data into meaningful way to detect mass imbalances. After careful research, as we discussed earlier the time domain data is converted into frequency domain data since time domain doesn't consist of enough information when dealing with vibration of the rotor speed. It is suggested to convert to frequency domain since it contains power of the signal. This process is performed by power spectral density of Welch's method. The fig 4.2 represents the frequency spectrum of the rotor speed where the peak frequency is equal to 1p frequency of the turbine. Thereby it is proved that the wind turbine is affected by mass imbalance.

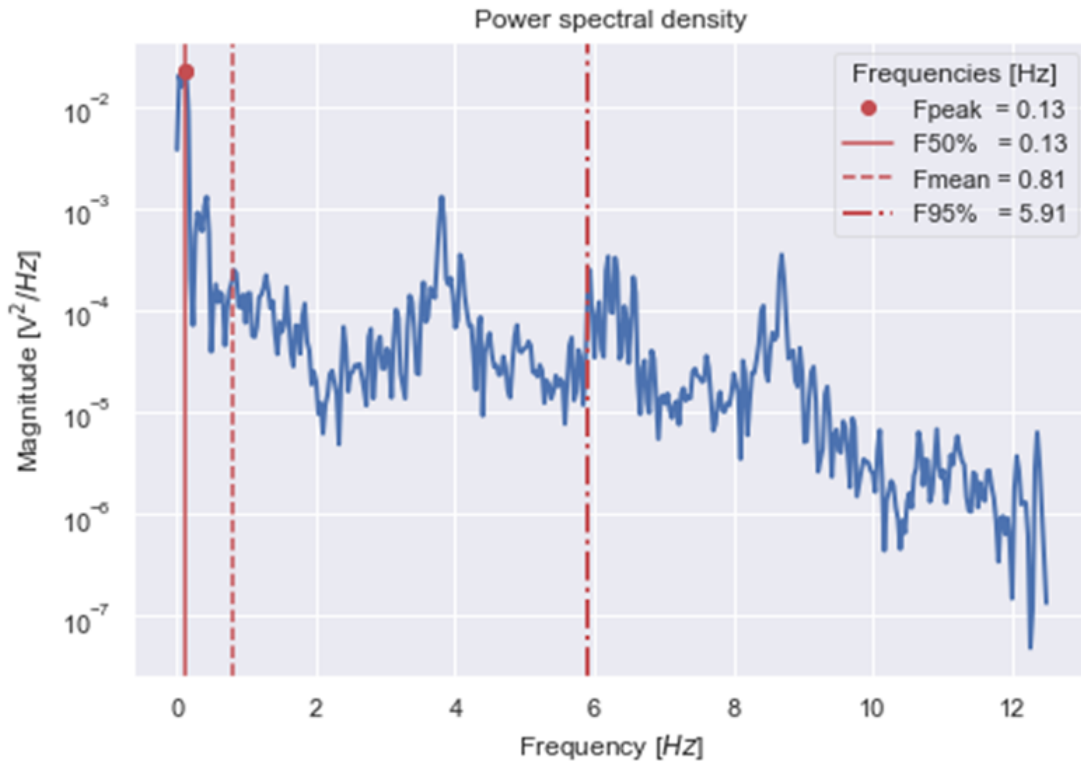


Figure 4.2: 1p Frequency of the Signal

The different level of mass imbalance is developed such as 0,2%,4%,6%,8%,10%,14% and 18% and it is labelled as each output classes for the input data to predict. The fig 4.3 represents the 2% deviation from the base mass of the blades. The blue coloured spectrum represents the balanced rotor that is no mass imbalance and orange coloured

spectrum represents the 2% mass imbalanced rotor. The data analysis is performed to check whether the mass imbalance percentages are correctly simulated. The same is followed for all other mass imbalance percentages.

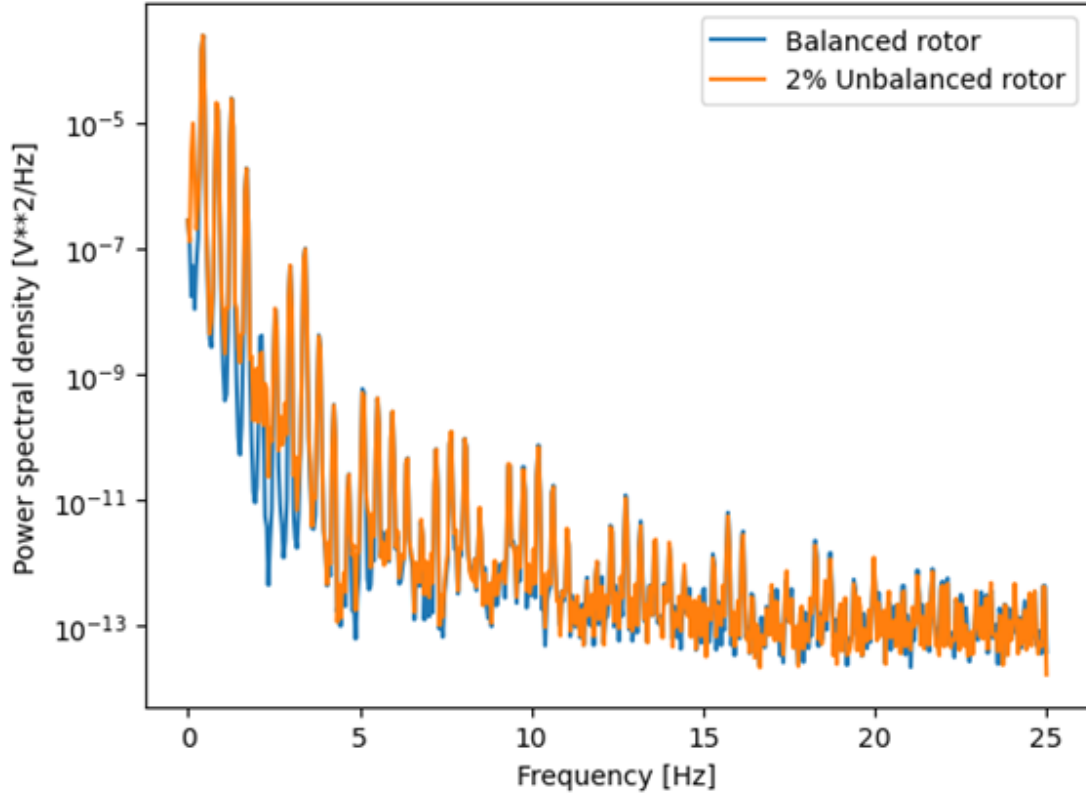


Figure 4.3: PSD Plot of Mass Imbalance 2%

After the data analysis and feature engineering, the input data with useful information for prediction is obtained. The domain knowledge of wind energy is needed to perform feature engineering techniques. Now, the input data after transformation is consist of 1p peak frequency PSD values and wind speed ranges from 4m/s to 20m/s. After the transformation, our input has 4000 samples from raw data which has in millions. The input data is then splitted into train and test set using hold-out cross validation method which is essential for machine learning model. After careful consideration, the split we have used is 80% for train and 20% for test. Another popular splitting such as 70%-30% is also used to check the best split for the model and we have seen that 70-30 split performs poorly compared to 80-20 rule since the input data sample is limited, the training data should have more samples to train effectively. Therefore, we have considered 80-20 split for our model. The various machine learning classification algorithm is used to train the model such as

- Logistic regression
- KNN classifiers
- Support Vector Machine
- Decision tree
- Random forest
- Extratreesclassifier

These models are selected based on following criteria such as

- Objective and quantity of input data since less input data samples won't be effective to train neural network model and the feature engineering performed by domain knowledge person is needed instead of automatic process in deep learning models.
- Computationally efficient since predictive maintenance should perform as quickly as possible.
- Explainability and interpretability of the model since the black box model does not gives enough information of how the model works which could be an issue if we don't know which parameters are responsible for mass imbalance detection.

Section 3 discusses the definitions of all the algorithms. Before training, the model is fine-tuned using hyperparameters using the grid search cross validation approach. Each model has its own set of hyperparameters, which are chosen based on extensive machine learning knowledge. Because it is a tree-based ensemble approach for classification, Extratreesclassifier and random forest share the same hyperparameters. To discover the best possible pair of hyperparameters for the mass imbalance model, the following hyperparameters are chosen and trained using grid search cross validation.

Hyperparameters	Values
n_estimators	100, 200, 300, 400, 500
max_depth	5, 10, 15, 20
min_samples_split	3, 6, 9
max_features	Log2, Auto, Sqrt

Table 4.2: Hyperparameters and their Corresponding Values

The table 4.2 depicts vaious parameters that is used to customize the extratreesclassifier model's hyperparameters. The grid search cv technique is then performed to

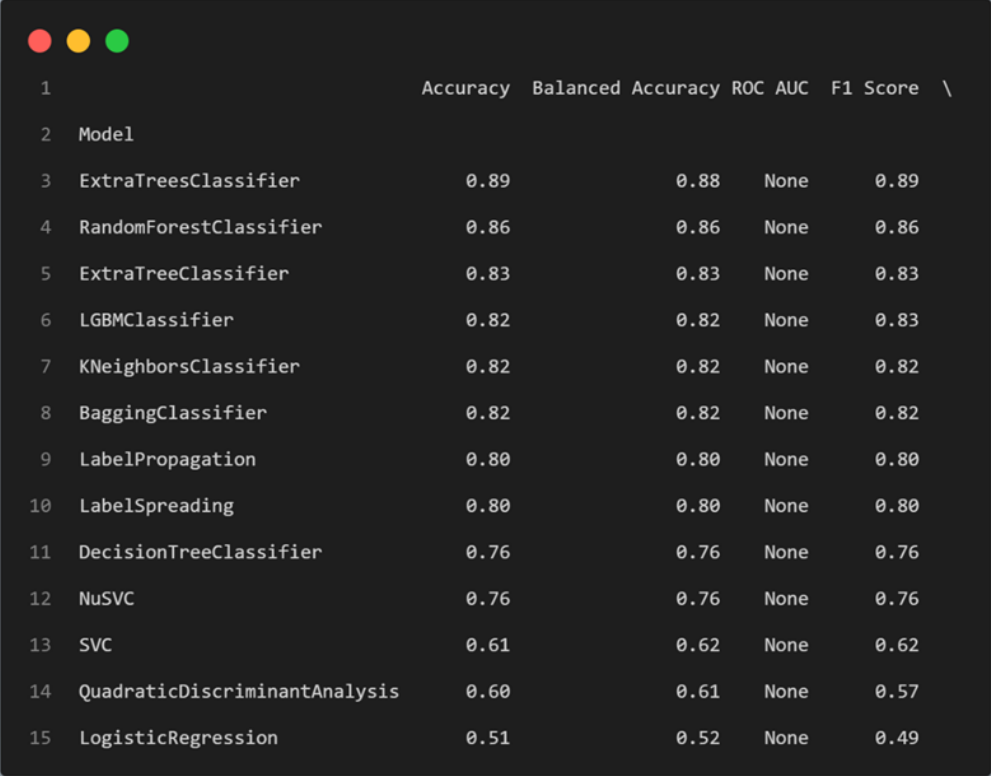
this parameters with a cross validation value of 5, thus the grid search will find the best possible pair of parameters for each fold of data by taking into account all of the parameters and printing the best pairings for our model to train the model. Grid search CV chooses the following hyperparameters for the extratreesclassifier model. Once the

Hyperparameters	Values
n_estimators	200
max_depth	10
min_samples_split	3
max_features	Log2

Table 4.3: Grid Search Hyperparameter Values

hyperparameters are determined, a machine learning model, such as extratreesclassifier, is trained on the parameters to achieve the best results. The training accuracy after training the model is 90%, indicating that the model was properly trained and did not overfit the data. The trained model is then evaluated on a 20% split unseen test set in which the model is not visible during the training phase. The test accuracy measures the model's performance on unseen data, which is significant in machine learning. The model scored 89% test accuracy for unseen test data with multiclass of more than three classes, which is a very excellent result. The fig 4.4 represents the multiple classification models which is trained and tested on our data and found out extratreesclassifier model is the best fit model for our mass imbalance detection problem. The figure depicts the top performing models, which include extratreesclassifier, random forest, LGBM classifier, KNN Classifier, and bagging classifier, in which decision tree, KNN, and extratreesclassifier models are bagged. Logistic regression is the worst performing model for two reasons.

- Due to the intricacy of the dataset, the logistic regression model may be more difficult to fit appropriately.
- It does not support the multi-classification problem; even when we try to fit the model with 8 classes, we receive the bad result displayed in the picture below.



1		Accuracy	Balanced Accuracy	ROC	AUC	F1 Score	\
2	Model						
3	ExtraTreesClassifier	0.89		0.88	None	0.89	
4	RandomForestClassifier	0.86		0.86	None	0.86	
5	ExtraTreeClassifier	0.83		0.83	None	0.83	
6	LGBMClassifier	0.82		0.82	None	0.83	
7	KNeighborsClassifier	0.82		0.82	None	0.82	
8	BaggingClassifier	0.82		0.82	None	0.82	
9	LabelPropagation	0.80		0.80	None	0.80	
10	LabelSpreading	0.80		0.80	None	0.80	
11	DecisionTreeClassifier	0.76		0.76	None	0.76	
12	NuSVC	0.76		0.76	None	0.76	
13	SVC	0.61		0.62	None	0.62	
14	QuadraticDiscriminantAnalysis	0.60		0.61	None	0.57	
15	LogisticRegression	0.51		0.52	None	0.49	

Figure 4.4: Trained Multiple Classification Models

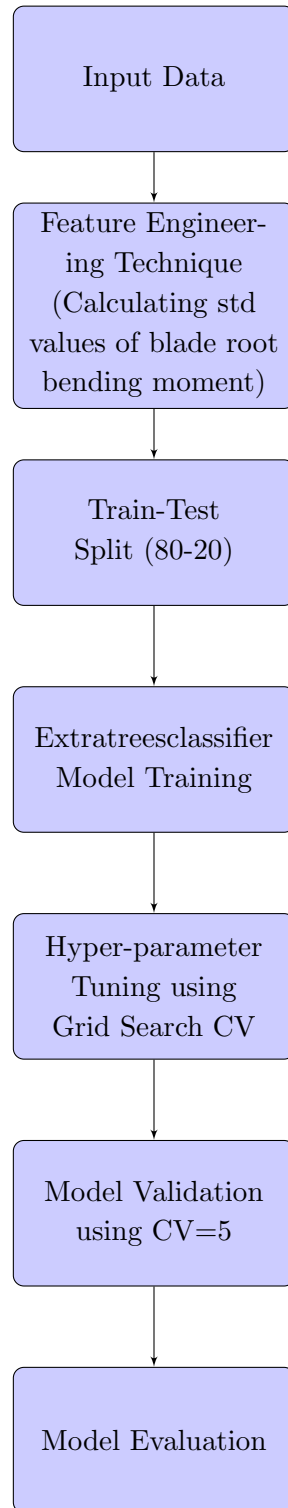
The method employs PSD values of 1P peak frequency obtained from rotor speed data, has been trained on 15 separate classification algorithms that have been independently trained, tweaked, and tested. When the performance of the algorithms is compared, it is said that extratreesclassifier exceeds all other methods by obtaining 89% accuracy. The trained model is then evaluated again with SCADA test data from the wind turbine, and it slightly under performs in the majority of the unseen test samples. It is because the wind turbine demonstrated at Fraunhofer IWES, Bremerhaven, is less susceptible to 1p vibrations where a maintenance firm that performs maintenance on the onsite wind turbine already measures and reports these data in a log sheet. As a result, the trained model may be utilized as a stand-alone model that can forecast mass imbalance with high accuracy (based on the simulation test accuracy assumption) when tested with new wind turbine data in the future.

Input data	PSD values of 1p peak frequency of AD8 wind turbine model, wind speed ranges from 12m/s to 20m/s with turbulence intensity of 9.5%.
Best performing model after hyper-parameter tuning	Extra-trees Classifier
Accuracy	Training – 90%, Test- 89%
Limitations	Real world testing is still needed because the available SCADA data is less prone to vibrations.
Output class labels	0, 2%, 4%, 6%, 8%, 10%, 14% and 18% of mass imbalance.

Table 4.4: Summary of Approach 1

4.3 Approach 2: Blade Root Bending Moments and Wind Speed

We followed the same procedure as in the prior method, with a few exceptions. We used data from the blade root sensor to forecast the mass imbalance in the wind turbine. The blade root sensors, represented by M_{xBRi} , M_{YBRi} , and M_{zBRi} , can measure the bending moment at the blade root around the local x , y , and z axes. The advantage of these sensors is that they enable the identification of faults from 10-minute average quantities, which are easier to manage and process faster than high-resolution instantaneous time-series data from the previous approach, which requires a more sophisticated data management system due to the large volume of data. These sensors provide a way for obtaining this information directly from the averaged readings, while spectral analysis employing high resolution time-series can assist in determining the presence of imbalances. The reason for selecting this characteristic as an input feature is because anytime a mass imbalance develops in a wind turbine, it causes an extra periodic moment at the blade root in the edgewise direction. Ultimately, the extreme values and standard deviation of the edgewise moment increases whereas the blade root’s torsion and flap wise moments are mostly unaffected. The edgewise bending moment at the blade root of our AD8 wind turbine is M_{xBRi} , the flapwise bending moment is M_{yBRi} , and the torsional bending moment is M_{zBRi} . It is not the default axis for all wind turbines, hence it may vary depending on the turbine.



Mass Imbalance Fault Signature	Standard deviation of the rotation of the blades gets increased
Input Features	Blade root bending moments in edgewise direction, Wind-speed (m/s) ranges from 4m/s to 20m/s with turbulence intensity 9.5%.
Sampling Frequency	F _s =100Hz of 10 minutes period
Class Labels	Mass Imbalance levels – 0, 2%, 4%, 6%, 8%, 10%, 14% and 18%.
Feature Engineering Technique	Calculated standard deviation of the blade root bending moments with 10 minutes interval in edgewise direction, mean of the corresponding wind speed.

Table 4.5: Input Features, Class Labels, and Feature Engineering Technique for Detecting Mass Imbalance Faults in Wind Turbines

As previously stated, there will be an extra periodic moment in the edgewise direction, which is indicated by its extreme values and standard deviation; the axis MxBR_i should be examined for our problem to discover the mass imbalance. The block diagram of approach 2 is shown above. The MxBR_i, or the blade root bending moment in the edgewise direction, was obtained from millions of data samples from the AD8 wind turbine model and the standard deviation of the bending moments was determined using 10minutes average amounts. The related mean value of wind speed is also determined. The output class labels are the same as in the previous method, which is a mass imbalance percentage of 0,2,4,6,8,10,14, and 18. It is obvious that the blade root measurements (MxBR) will only represent the influence of mass imbalance in the edgewise direction. The edgewise moment on the blades is caused by gravity, and the tangential force by the wind, which rotates the rotor, according to a force analysis of the blades. The influence of gravity on the edgewise moments reduce as the wind speed increases because the load due to gravity does not increase with wind speed whereas the tangential force does. Therefore, it is anticipated that the mass imbalance would be more noticeable at low wind speeds. The fig 4.5 represents the scatter plot of the standard deviation of bending moment samples with wind speeds. To obtain better representation, the samples were averaged by 1min quantities but for training the model we use only 10minute samples.

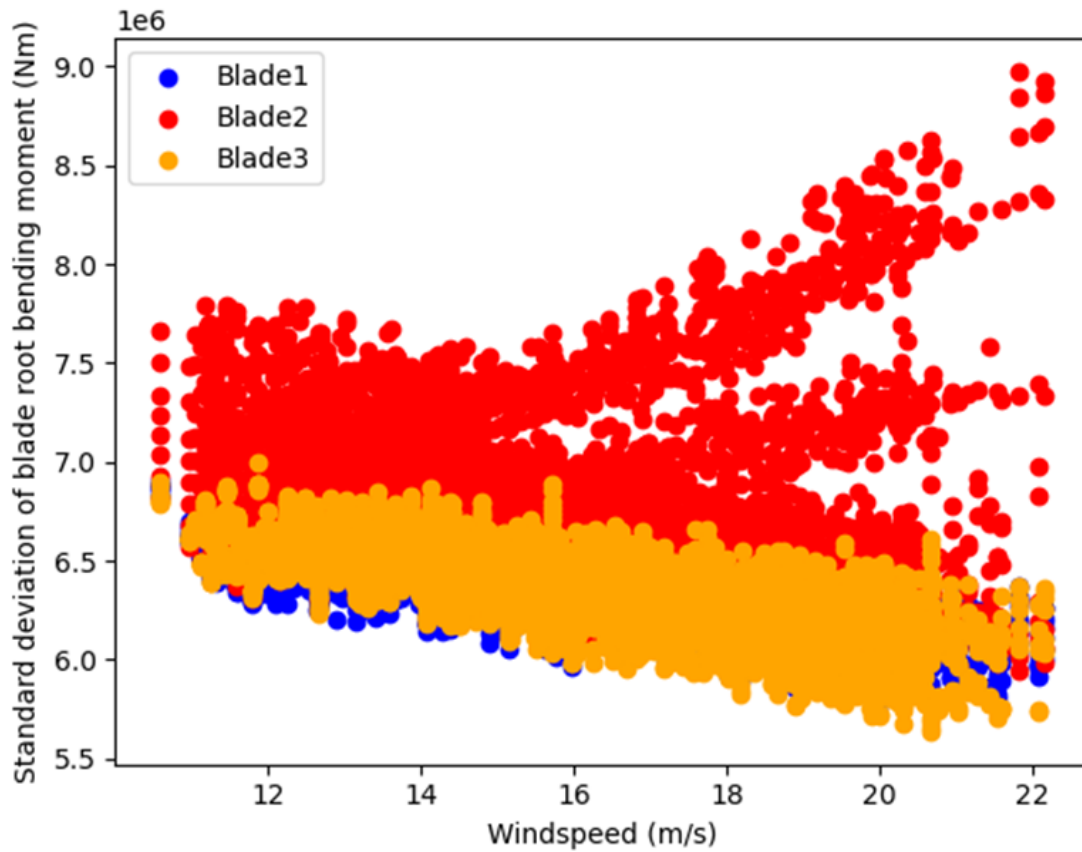


Figure 4.5: Standard Deviation of Bending Moments for all Three Blades

The graph illustrates that when wind speed rises, the standard deviation of blade root bending moment of blade 2 increases while other blades decrease. It is evident that blade 2 has a mass imbalance among the blades, and the procedure is continued based on these discoveries. As we now know that blade 2 has a mass imbalance, we determined the difference in standard deviation of blade root bending moment of blade 2 from the other two blades. The features are

- $M_{x2} - M_{x1}$
- $M_{x2} - M_{x3}$

The numbers are calculated because the machine learning algorithm will find better patterns for predicting mass imbalances, such as the difference in standard deviation of bending moment between damaged and unaffected blades.

4 IMPLEMENTATION

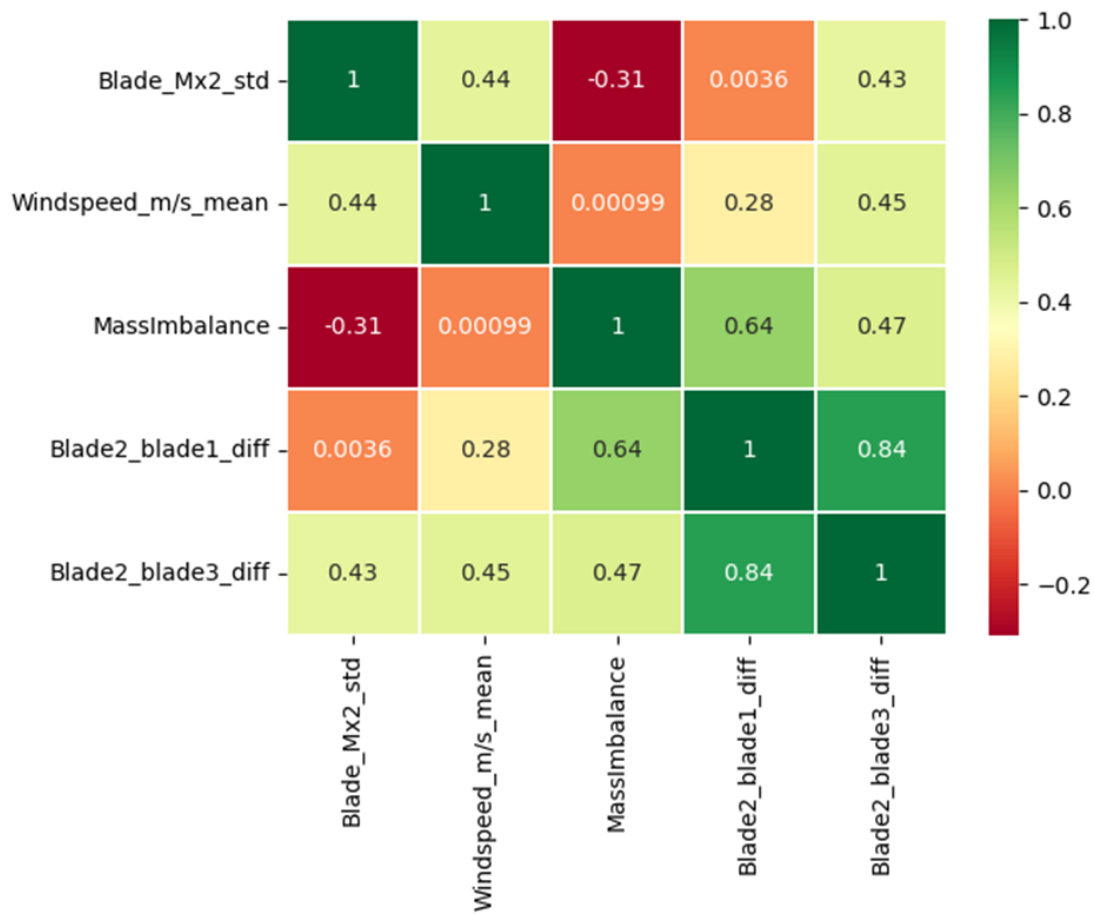


Figure 4.6: Correlation Matrix of the Features

The correlation matrix represents the correlation between the features which ranges from 0 to 1. The correlation value near to 1 means highly correlated and value near to 0 is less correlated. If any two input features are highly correlated, then dropping any one feature is feasible since they both shares the same information with the output class label. Here, the input feature such as blade_Mx2_std and wind speed have average correlation also blade2_blade3_diff and wind speed have the correlation value of 0.45. If we see the correlation between input feature and output feature, the feature such as blade2_blade1_diff and mass imbalance have correlation value of 0.64 which means the input feature blade2_blade1_diff is contributing more to the output class mass imbalance. Also, it should be noted that there is no straight relation between wind speed and mass imbalance which is proved here by having correlation value of 0.00099. The input data is then splitted into train and test using holdout validation by 80-20 rule. The machine learning algorithm such as tree-based algorithm does not need scaling of the data since it uses tree to find the patterns by splitting the nodes as deep as possible. Also, it

supports feature importance which gives the highest score possible to the feature which contributes to the prediction followed by the least as 0 to highest as 1. The fig 4.7 represents the feature importance of the input data.

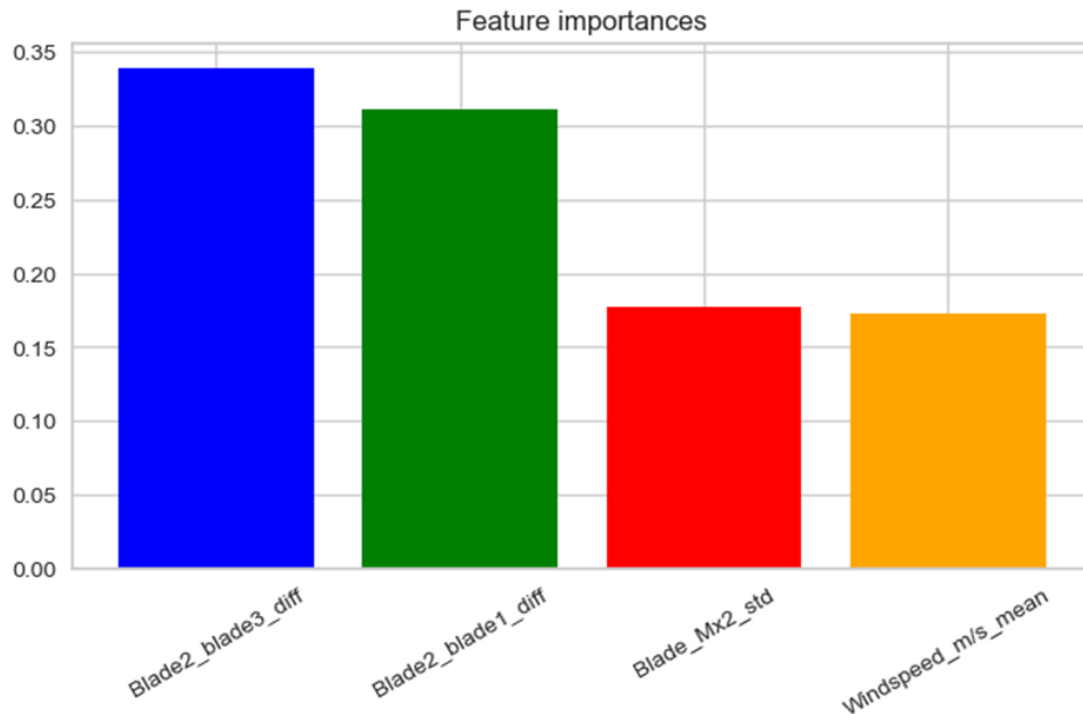


Figure 4.7: Feature Importance of the Input Features

The fig 4.7 shows that blade2_blade3_diff feature is having highest importance in prediction of mass imbalance followed by blade2_blade1_diff. The wind speed feature is having least importance in predicting the mass imbalance since wind speed is not directly relates to mass imbalance. It is must to check the balance quality of output class label before training the model. If the data is imbalanced then balancing technique or train model which supports class imbalance should be applied. In our data, we have 8 different class label such as simulated mass imbalance percentages, and it should have near to equal amount of data to have the better prediction. If any one of the classes has more amount of data, then the machine learning algorithm will assign higher priority to the majority class and leaves the minority class with lesser priority that leads to biased prediction, and it should be avoided. The fig 4.8 shows the representation of output mass imbalance classes with its count. It ensures our input data has no class imbalance.

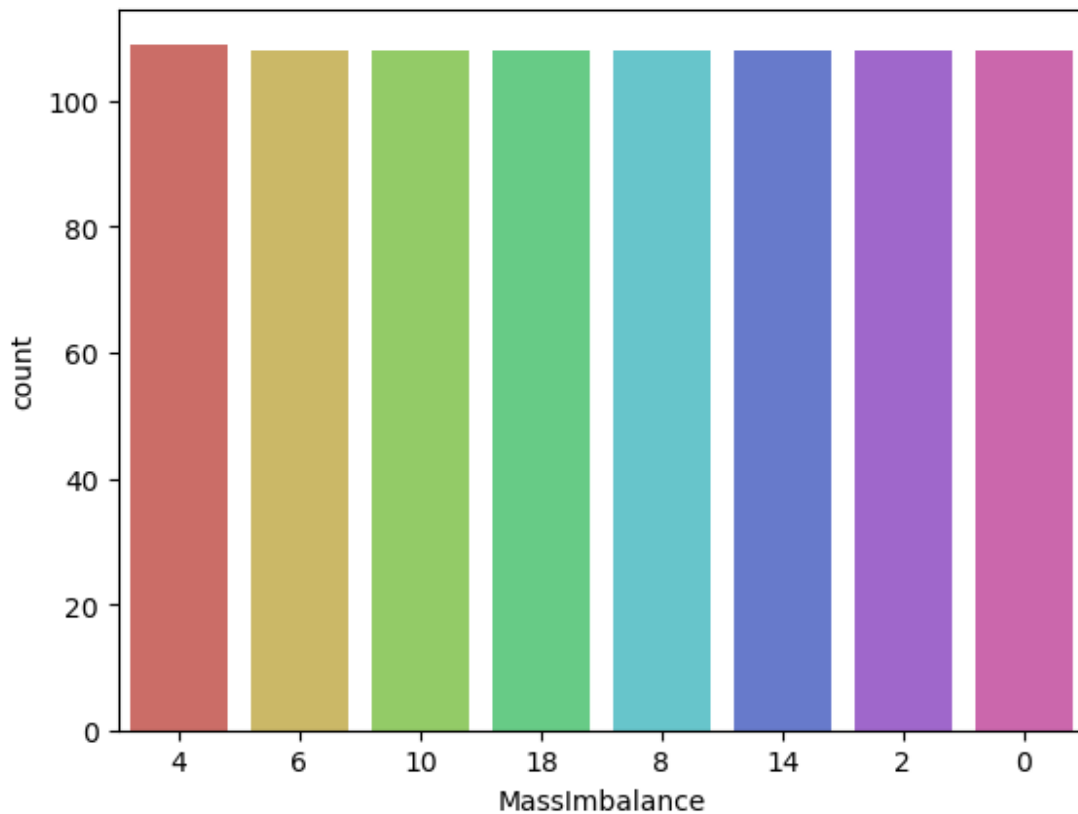


Figure 4.8: Class Balancing Nature

The fig 4.8 depicts the output classes balancing nature, and it is discovered that the data does not have any class imbalance since each output class shares the same number of input samples. As a result, the data is now ready for training the machine learning model, as there is no noise in the input data. We employed the same machine learning classification techniques as in the prior approach because the nature of the issue and dataset are almost identical. The data was trained on 15 different algorithms, and it was discovered that the random forest method and extra-trees classifier produce better results than other models. Then we chose these two models and did hyperparameter adjustment to get the optimum model for mass imbalance prediction. Because random forest and extra-trees classifier have the same hyper-parameters, they are frequently referred similar which is shown in table 4.6. Grid search cross validation is then applied to these hyper-parameters to find the best possible pairs for the model. The cross validation is applied to evaluate the model's performance using the different pairs of hyper-parameters. The final chosen hyper-parameters using grid search cv is as shown in table 4.7.

The following hyperparameters are then used to train the model to check whether the model's performance is increased or not and, in our case, there is a minor improvement in

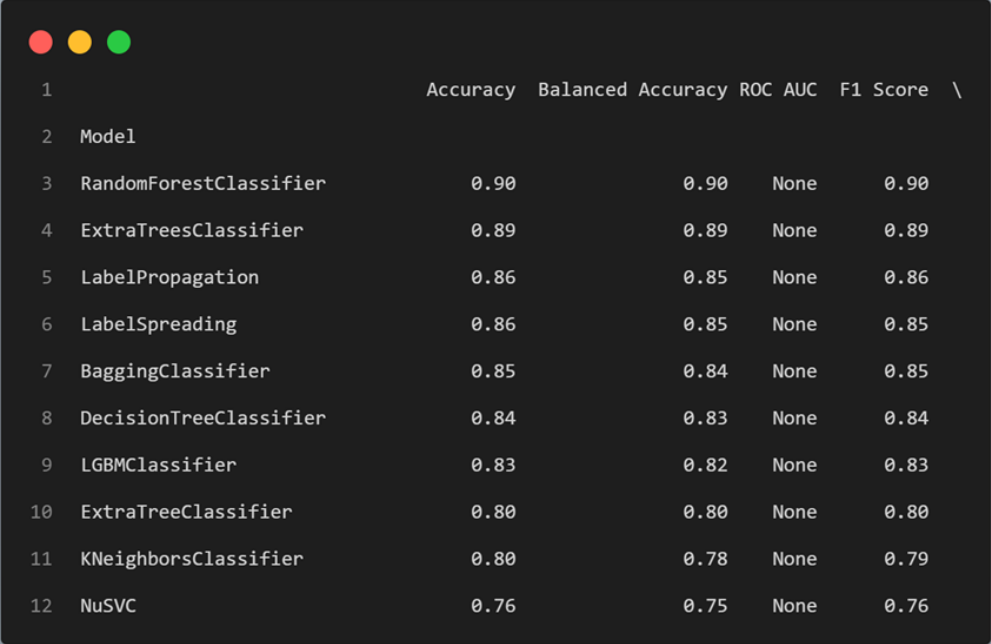
Hyper-parameters	Values
n_estimators	100, 200, 300, 400, 500
max_depth	5, 10, 15, 20
min_samples_split	3, 6, 9
max_features	Log2, Auto, Sqrt

Table 4.6: Hyper-parameters and their Corresponding Values

Hyper-parameters	Values
n_estimators	200
max_depth	15
min_samples_split	3
max_features	Auto

Table 4.7: Hyper-parameters Chosen by Grid Search

the model's performance which may be crucial in predicting the mass imbalance in real world. Finally, the model is tested with the test set and achieved the overall accuracy of 90% in random forest and 89% in extratreesclassifier. The fig 4.9 shows the performance of fifteen different classification algorithms. After the careful consideration, we have chosen extratreesclassifier over random forest since predictive maintenance strategies needs to perform as quickly as possible. So, the extratreesclassifier is computationally efficient than random forest when it is trained on huge datasets. Also, apart from accuracy other metric such as precision, recall and f1 score also considered while choosing the best performance model. Further the trained model is tested again with SCADA data, and it successfully classified the 141 samples as mass imbalance of 8% out of 144 test samples.



```

1          Accuracy  Balanced Accuracy ROC AUC  F1 Score \
2  Model
3  RandomForestClassifier      0.90          0.90  None    0.90
4  ExtraTreesClassifier       0.89          0.89  None    0.89
5  LabelPropagation           0.86          0.85  None    0.86
6  LabelSpreading             0.86          0.85  None    0.85
7  BaggingClassifier          0.85          0.84  None    0.85
8  DecisionTreeClassifier     0.84          0.83  None    0.84
9  LGBMClassifier             0.83          0.82  None    0.83
10 ExtraTreeClassifier         0.80          0.80  None    0.80
11 KNeighborsClassifier       0.80          0.78  None    0.79
12 NuSVC                      0.76          0.75  None    0.76

```

Figure 4.9: Trained Multiple Classification Models

Input data	Standard deviation of blade root bending moment in edgewise direction, wind speed ranges from 4m/s to 20m/s with turbulence intensity of 9.5%.
Best performing model after hyper-parameter tuning	Random forest / Extra-trees classifier. But preferred extra-trees classifier due to its efficient computational resource.
Accuracy	Training – 91%, Test- 90%
Predictions on SCADA Data	Classified with 98% accuracy..
Output class labels	0, 2%, 4%, 6%, 8%, 10%, 14% and 18% of mass imbalance.

Table 4.8: Summary of Approach 2

5 Results and discussion

Once the final model is trained and tested with its best pairs of hyperparameters, it is necessary to validate and evaluate the model's performance. To avoid overfitting and have better generalization, the model is validated by k-fold cross validation with $k=5$. The fig 5.1 represents the cross-validation process of our dataset.

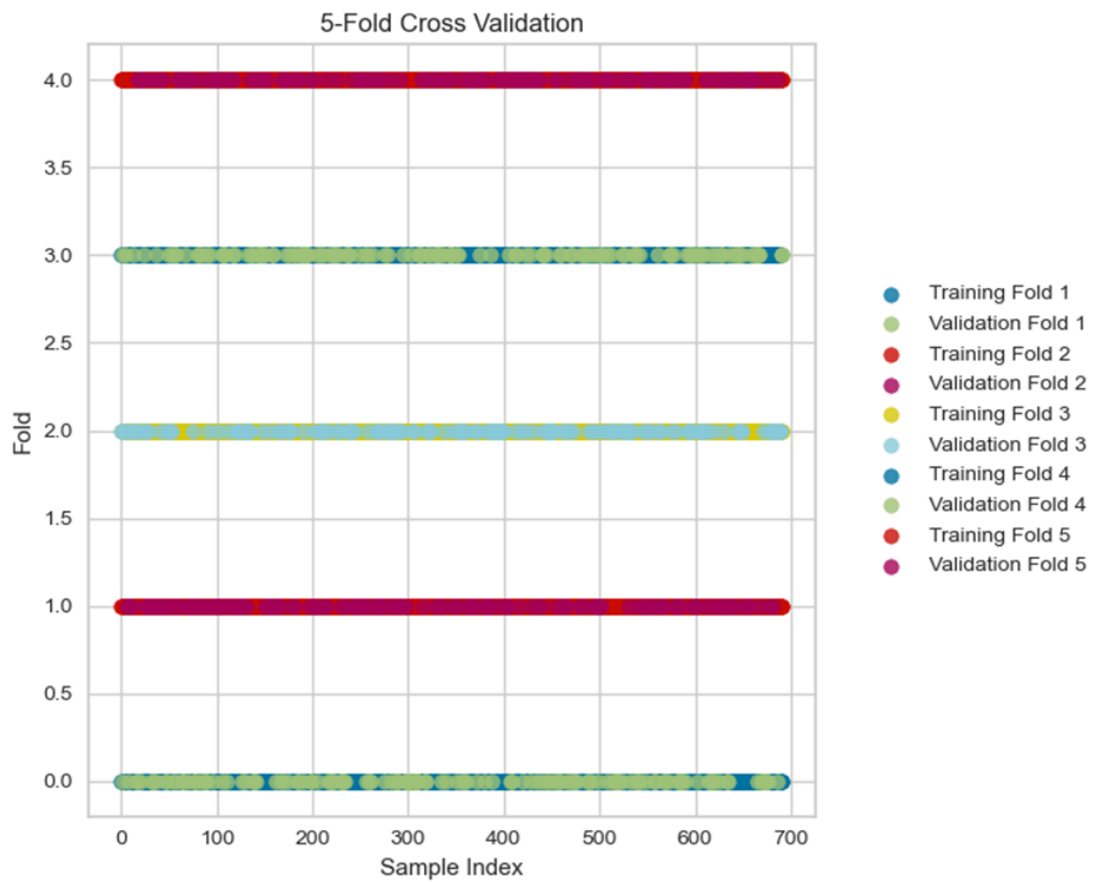


Figure 5.1: Cross Validation plot for MASS Imbalance Model

The fig 5.1 is plotted the input sample size in x axis and number of cross validation folds in y axis. Since $k = 5$, the training samples of around 700 is undergone cross validation by splitting as 5folds. The 140 input samples of each fold is used to train the model and the remaining will be assigned as a test or validation fold. The proces will

be iteratively performed and each fold’s accuracy is combined and calculated the mean accuracy which serves as the best possible accuracy the model can obtain. It is very useful when the input data is less and can obtain better accuracy by train-test split.

The next important step of any machine learning process is to evaluate the model’s performance by performance metrics. Since our mass imbalance detection is multiclass problem, the model is evaluated using classification metrics such as accuracy, precision, recall and f1-Score. Also, one of the important classification metric is confusion matrix where it describes how much samples are correctly classified the actual class and how many doesnot. The fig 5.2 represents the confusion matrix of the model.

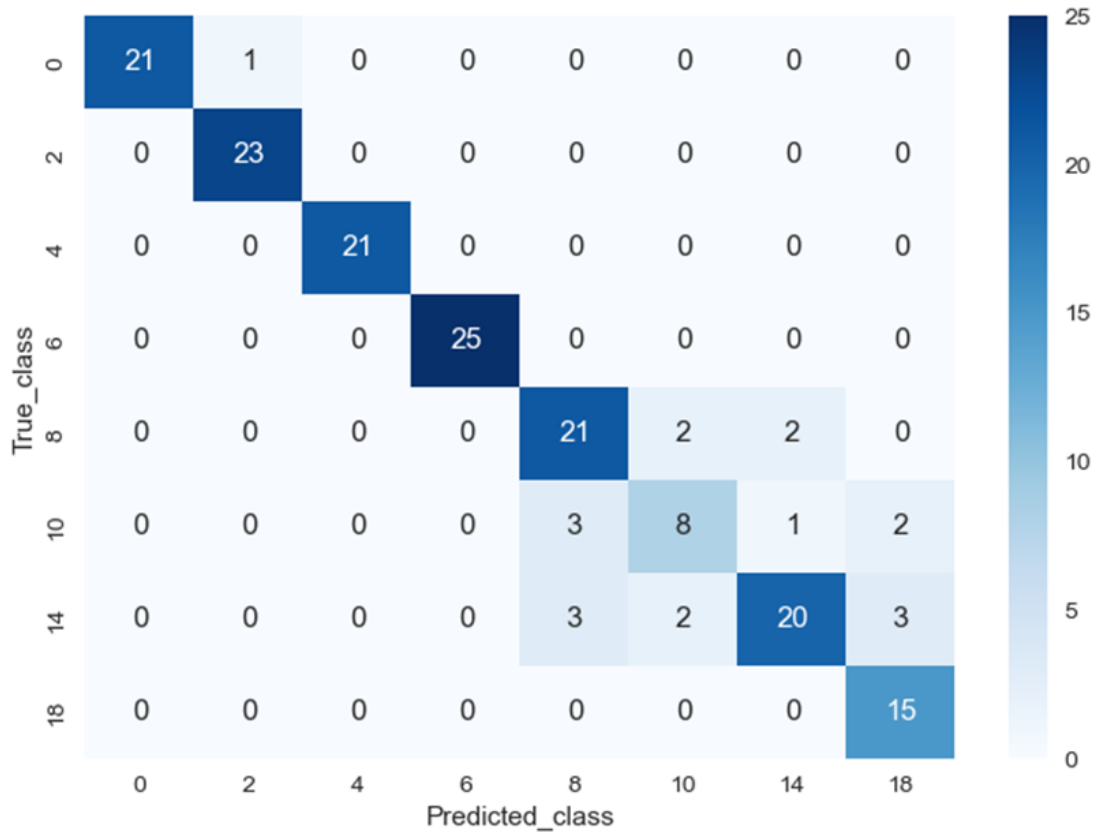


Figure 5.2: Confusion Matrix of Extratreesclassifier Model

The fig shows that the mass imbalance percentage such as 0,2,4 and 6 of all the samples are correctly classified and the remaining mass imbalance classes are slightly misclassified. This is because of not having enough samples to test the data. The misclassification is due to lesser number of test samples compared to lower mass imbalance level samples. Also, in real world scenario the above 10% may not be exist in all the cases and it is only measured and calculated for research purpose. In most of the cases, accuracy

alone won't give the better representation of the model so we have to consider other metrics such as precision, recall and f1-score. Choosing the metrics is very important and it is done after the understanding of the problem's objective. In some cases precision should be important and for other cases, recall is important and maintaining the tradeoff between these two metrics is very important in any machine learning model. In mass imbalance detection problem, having better recall score is important because the scenario like actually having mass imbalance in the wind turbine and the model predicts the wind turbine has no mass imbalance is the bad scenario which should be avoided. The fig 5.3 represents the classification report of our model which consist of accuracy, precision score, recall score, f1-score and support. The term support refers to number of test samples used for prediction.

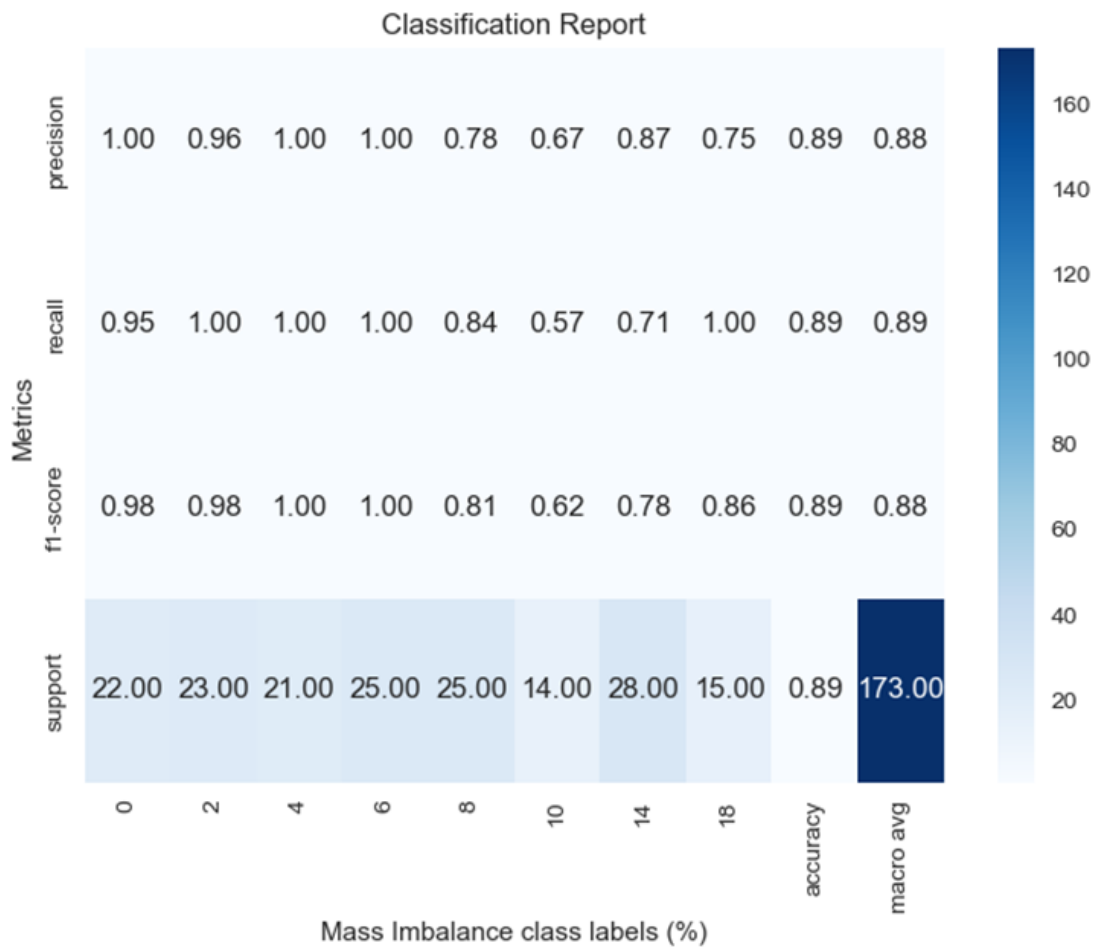


Figure 5.3: Classification Report

The classification report shows that `extratreesclassifier` performs really well on test data an achieved 88% as precision score, 89% as recall score, 88% as f1-score for the total test sample size of 173. The recall score is higher than precision which serves the need of the mass imbalance prediction research. Now, the trained model is further tested using unseen SCADA test data to predict the mass imbalance. Since SCADA data has no ground truth, it is difficult to evaluate the model with classification metrics which compares the actual vs predicted. Once the model is tested using SCADA test data, the model predicts 141 samples as mass imbalance of 8% out of 144 samples. This result is further investigated by the site enigneers at Fraunhofer IWES about the authenticity of the classification results. Also, we have tested the trained model using new unseen simulated test data and the model was able to classify the mass imbalance classes with 97% accuracy.

Accuracy with test data split using hold out validation	90%
Accuracy with newly simulated AD8 test data	97%
Accuracy with SCADA test data without having ground truth	98% (accuracy is calculated based on correctly classified vs incorrect classification) and the authenticity of the result is evaluated by site enigneers.

Table 5.1: Accuracy Measurements for Different Test Datasets

Once the model has been trained and evaluated, it is exported as a pickle file, which the flask framework uses to build a machine learning application. The same preparation procedure is repeated inside the flask app by importing the pickle file containing the `extratreesclassifier` model. The front end web page is built using HTML/CSS and has a user-friendly style, allowing the user or maintenance team member to simply input the SCADA data of the wind turbine in csv file format to obtain the mass imbalance prediction result. Because of its easy and interactive design, this application is intended for both technical and non-technical users. The figure 5.4 depicts a mass imbalance detection application built using the Flask framework and a RESTful API.



Figure 5.4: Mass Imbalance Detection Application

5.1 Limitations

- While the findings obtained are outstanding, they are restricted to the statistical features of the training wind turbine data. Furthermore, it can predict reasonably and this model's ability to classify when the attributes change is due to Extra-treesclassifier's low variance property.
- As a result, in order to enhance accuracy further, the model must be retrained on new test data.
- The legitimacy of the test accuracy on SCADA data is checked by site engineers because we do not know the ground truth, which may be subject to human error.

6 Conclusions

- The study revealed the importance of machine learning algorithms in wind turbine predictive maintenance applications. The model built in this work outperforms the state-of-the-art methods on an 8-class multiclassification task, obtaining excellent accuracy.
- The study revealed the importance of machine learning algorithms in wind turbine predictive maintenance applications. The model built in this work outperforms the state-of-the-art methods on an 8-class multiclassification task, obtaining excellent accuracy.
- It takes fewer computing resources and time than comparable models, making it a cost-effective predictive maintenance option.
- One of the model's primary advantages is that no additional sensors are required, making it simple to integrate into current wind turbine systems.
- The model is also more explainable and interpretable, making it easier for stakeholders to comprehend and share the results.
- Furthermore, by obtaining excellent accuracy on real-world SCADA test data, the model has proved its robustness and generalizability.
- Although there are still limits and more research needed to increase the accuracy and generalised results, the study's findings have provided useful insights for the future research, utilizing machine learning algorithms in predictive maintenance applications.
- Finally, this thesis has proved the use of machine learning techniques in predicting mass imbalance in wind turbines.

6.1 Future Work

- More number of data should be collected in the future to increase the model's accuracy.
- Also, depending on the nature of the input data, investigate different machine learning approaches to increase the model's performance on the data. Consider

6 CONCLUSIONS

additional factors that may affect wind turbine performance because of mass imbalance.

- Retrain the model on a variety of data, including different turbines and operating circumstances, to generalize the process for unseen turbines.
- This procedure will assist the model in learning the characteristics that are shared by all other turbines and will increase its capacity to predict the maintenance requirements in other turbines.

Bibliography

- [1] Statista, “Leading countries for primary energy consumption 2018.” <https://www.statista.com/statistics/263455/primary-energy-consumption-of-selected-countries/>, Jun 2020.
- [2] P. Tchakoua, R. Wamkeue, M. Ouhrouche, F. Slaoui-Hasnaoui, T. Tameghe, and G. Ekemb, “Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges,” *Energies*, vol. 7, pp. 2595–2630, Apr. 2014.
- [3] K. Kong, K. Dyer, C. Payne, I. Hamerton, and P. M. Weaver, “Progress and trends in damage detection methods, maintenance, and data-driven monitoring of wind turbine blades – a review,” *Renewable Energy Focus*, Aug. 2022.
- [4] S. Wan, K. Cheng, X. Sheng, and X. Wang, “Characteristic analysis of dfig wind turbine under blade mass imbalance fault in view of wind speed spatiotemporal distribution,” *Energies*, vol. 12, pp. 3178–3196, Jan. 2019.
- [5] A. Stetco, F. Dinmohammadi, X. Zhao, V. Robu, D. Flynn, M. Barnes, J. Keane, and G. Nenadic, “Machine learning methods for wind turbine condition monitoring: A review,” *Renewable Energy*, vol. 133, pp. 620–635, Apr. 2019.
- [6] Mabble Rabble, “Machine learning mindmap.” <http://mabblerabble.blogspot.com/2016/09/machine-learning-mindmap.html>, Sep 2016.
- [7] Unknown, “Understanding logistic regression.” <https://www.geeksforgeeks.org/understanding-logistic-regression/>, May 2017.
- [8] S. E R, “Random forest | introduction to random forest algorithm.” <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/>, June 2021. [Online; accessed 14-May-2023].
- [9] scikit-learn contributors, “sklearn.ensemble.extratreesclassifier – scikit-learn 0.22.2 documentation.” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesClassifier.html>. [Online; accessed 14-May-2023].
- [10] Unknown, “Multiclass classification using svm | svm for multiclass classification.” <https://www.analyticsvidhya.com/blog/2021/05/multiclass-classification-using-svm/>, May 2021.

BIBLIOGRAPHY

- [11] A. Burkov, “The hundred-page machine learning book,” 2019.
- [12] M. Sunasra, “Performance metrics for classification problems in machine learning.” <https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>, 02 2019.
- [13] “How to easily deploy machine learning models using flask.” <https://www.kdnuggets.com/2019/10/easily-deploy-machine-learning-models-using-flask.html>. [Online; accessed 14-May-2023].
- [14] R. Pandit, D. Astolfi, J. Hong, D. Infield, and M. Santos, “Scada data for wind turbine data-driven condition/performance monitoring: A review on state-of-art, challenges and future trends,” *Wind Engineering*, p. 0309524X2211240, 09 2022.
- [15] P. Ortiz, “10 wind energy facts and statistics in 2023.” <https://housegrail.com/wind-energy-facts-and-statistics/>, 02 2022.
- [16] P. Bojek, “Wind electricity – analysis.” <https://www.iea.org/reports/wind-electricity>, 09 2022.
- [17] “Wind market reports: 2022 edition.” <https://www.energy.gov/eere/wind/wind-market-reports-2022-edition>, 2022.
- [18] W. Contributors, “Wind power in germany.” https://en.wikipedia.org/wiki/Wind_power_in_Germany, 05 2023. [Online; accessed 14-May-2023].
- [19] The Independent, “Offshore wind turbines ‘have potential to meet entire world’s electricity needs 18 times over’,” 10 2019.
- [20] G. R. Hübner, L. D. da Rosa, C. E. de Souza, H. Pinheiro, C. M. Franchi, R. B. Morim, S. Ekwaro-Osire, J. P. Dias, and S. Dabetwar, “Wind turbine rotor aerodynamic imbalance detection using cnn,” *Journal of Physics: Conference Series*, vol. 2265, p. 032104, 05 2022.
- [21] U. T. Choijljav Ulam-Orgil, “Comparative study of solar and wind power plant tariff and investment costs,” *Journal Sustainable Development and Engineering Economics*, vol. 2, pp. 37–46, July 2022.
- [22] K. Fischer, F. Besnard, and L. Bertling, “Reliability-centered maintenance for wind turbines based on statistical analysis and practical experience,” *IEEE Transactions on Energy Conversion*, vol. 27, pp. 184–195, Mar. 2012.
- [23] B. Hahn, M. Durstewitz, and K. Rohrig, “Reliability of wind turbines,” *Wind Energy*, pp. 329–332, 2007.

BIBLIOGRAPHY

- [24] C. Crabtree, Y. Feng, and P. Tavner, “Detecting incipient wind turbine gearbox failure: A signal analysis method for on-line condition monitoring.” <https://www.semanticscholar.org/paper/Detecting-Incipient-Wind-Turbine-Gearbox-Failure:-A-Crabtree-Feng/4d2d68bd1d04312e2d5f93ec86dc5e5cdda96834>, 2010.
- [25] M. Costa, J. A. Orosa, D. Vergara, and P. Fernández-Arias, “New tendencies in wind energy operation and maintenance,” *Applied Sciences*, vol. 11, p. 1386, Jan. 2021.
- [26] S. Dabetwar, S. Ekwaro-Osire, J. P. Dias, G. R. Hübner, C. M. Franchi, and H. Pinheiro, “Mass imbalance diagnostics in wind turbines using deep learning with data augmentation,” *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, vol. 9, Jun. 2022.
- [27] E. Follower, “Wind energy: An introduction.” <https://energyfollower.com/wind/>. [Online; accessed May 11, 2023].
- [28] Z. Cao, J. Xu, W. Xiao, Y. Gao, and H. Wu, “A novel method for detection of wind turbine blade imbalance based on multi-variable spectrum imaging and convolutional neural network,” Jul. 2019.
- [29] J. Niebsch, R. Ramlau, and T. T. Nguyen, “Mass and aerodynamic imbalance estimates of wind turbines,” *Energies*, vol. 3, pp. 696–710, Apr. 2010.
- [30] G. R. Hübner, H. Pinheiro, C. E. de Souza, C. M. Franchi, L. D. da Rosa, and J. P. Dias, “Detection of mass imbalance in the rotor of wind turbines using support vector machine,” *Renewable Energy*, vol. 170, pp. 49–59, Jun. 2021.
- [31] G. R. Hübner, L. D. da Rosa, C. E. de Souza, H. Pinheiro, C. M. Franchi, R. B. Morim, S. Ekwaro-Osire, J. P. Dias, and S. Dabetwar, “Wind turbine rotor aerodynamic imbalance detection using cnn,” *Journal of Physics: Conference Series*, vol. 2265, p. 032104, May 2022.
- [32] J. Chen, W. Hu, D. Cao, B. Zhang, Q. Huang, Z. Chen, and F. Blaabjerg, “An imbalance fault detection algorithm for variable-speed wind turbines: A deep learning approach,” *Energies*, vol. 12, p. 2764, Jan. 2019.
- [33] D.-I. M. Fischer, “A formal fault model for component-based models of embedded systems,” May 2007.
- [34] Javatpoint, “Normalization in machine learning - javatpoint.” <https://www.javatpoint.com/normalization-in-machine-learning>, n.d.

BIBLIOGRAPHY

- [35] A. Bhandari, “Feature scaling | standardization vs normalization.” <https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/>, Apr. 2020.
- [36] A. Zheng and A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*. O’Reilly Media, Inc., Mar. 2018.
- [37] J. Vives, “Vibration analysis for fault detection in wind turbines using machine learning techniques,” *Advances in Computational Intelligence*, vol. 2, Jan. 2022.
- [38] M. R. Shahriar, P. Borghesani, and A. C. C. Tan, “Speed-based diagnostics of aerodynamic and mass imbalance in large wind turbines,” Jul. 2015.
- [39] H. Stensgaard Toft, K. Branner, R. Nijssen, D. Lekou, and C. A. Pueyo, “Probabilistic methods for wind turbine blades,” 2013.
- [40] G. Ren, J. Liu, J. Wan, F. Li, Y. Guo, and D. Yu, “The analysis of turbulence intensity based on wind speed data in onshore wind farms,” *Renewable Energy*, vol. 123, pp. 756–766, 08 2018.
- [41] Unknown, “Digital signal processing: Theory and practice.” <https://vdoc.pub/documents/digital-signal-processing-theory-and-practice-1e01641kg14g>, Unknown. [PDF].
- [42] P. Welch, “The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 70–73, 06 1967.
- [43] Statology, “Regression vs. classification: What’s the difference?.” <https://www.statology.org/regression-vs-classification/>, October 2020. [Online; accessed 14-May-2023].
- [44] J. Brownlee, “Tour of data sampling methods for imbalanced classification.” <https://machinelearningmastery.com/data-sampling-methods-for-imbalanced-classification/>, January 2020.
- [45] DataCamp, “Knn classification tutorial using sklearn python.” <https://www.datacamp.com/tutorial/k-nearest-neighbor-classification-scikit-learn>, n.d.
- [46] Genesis, “Pros and cons of k-nearest neighbors - from the genesis.” <https://www.fromthegenesis.com/pros-and-cons-of-k-nearest-neighbors/>, September 2018. [Online; accessed 14-May-2023].

BIBLIOGRAPHY

- [47] S. Mohanan, "Random forest in machine learning." <https://reflections.live/articles/1665/random-forests-for-classification-and-regression-2540-kvx3xew2.html>, Unknown.
- [48] Baeldung, "Multiclass classification using support vector machines," Oct 2020.
- [49] "Different types of cross-validations in machine learning." <https://www.turing.com/kb/different-types-of-cross-validations-in-machine-learning-and-their-explanations>. [Online; accessed 14-May-2023].



TECHNISCHE UNIVERSITÄT
CHEMNITZ

Zentrales Prüfungsamt
Selbstständigkeitserklärung

Name: Gowthaman Malarvizhi	Bitte beachten:
Vorname: Guhan velupillai	1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.
geb. am: 15.04.1997	
Matr.-Nr.: 563442	

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende **Masterarbeit** selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 17.05.2023

Unterschrift: Guhan Velupillai

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.