

Wind Turbine Gearbox Fault Prognosis Using High Frequency SCADA Monitoring Data

Ayush Verma

Technische Universiteit Delft



Wind Turbine Gearbox Fault Prognosis Using High Frequency SCADA Monitoring Data

Master's thesis report

by

Ayush Verma

to obtain the degree of Master of Science in Sustainable Energy Technology
at the Delft University of Technology,
to be defended publicly on August 31, 2021 at 05:30 PM.

Faculty of Aerospace Engineering, Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology

Student number: 5159989
Project duration: November 9, 2020 – August 31, 2021
Thesis committee: Prof. dr. Donatella Zappalá, TU Delft, supervisor
Prof. dr. S. J. Watson, TU Delft, Advisor
Dr. Shawn Sheng, NREL, Advisor

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

Preface

This thesis is the final work of my MSc. Sustainable Energy Technology program at the Delft University of Technology. In the past 10 months, I have had the opportunity to explore an entirely new and exciting field of artificial intelligence. From knowing almost nothing about the field of data science, to developing a machine learning based prognostic model was a steep and interesting learning curve. I undertook this endeavor to prepare myself even better to tackle complex and multifaceted challenges. Thanks to this experience, I was able to put my analytical skills to test and now I can confidently say that I am ready to contribute to energy transition in the best way possible. While I am very proud of my personal achievements during this project, it would not have been possible without the help and support of several people.

First and foremost, I would like to thank my supervisor Dr. Donatella Zapalla for the time and energy she dedicated to me. The weekly meetings always motivated and encouraged me to investigate further. Her guidance through this project and the feedback she gave me was absolutely invaluable and has allowed me to improve my work and results. She was very supportive and understanding throughout the project span and I am really grateful for that.

Furthermore, I want to thank NREL for sharing the data for this research. In particular, I want to express my gratitude to Dr. Shawn Sheng for his advice and recommendations that helped in improving my analysis with this dataset. I would also like to thank Dr. Simon J. Watson for his valuable insights during my mid-term presentation.

A special gratitude and love goes to my parents and brother for their unwavering financial, rational and emotional support to follow my aspirations in life. I would also like to express my deepest appreciation for my friends, without whom this journey would not have been as fun and exciting. I am particularly grateful to Ugné for her encouragement and care that helped me through the difficult times during this project. I also want to thank her for brainstorming ideas for this project and reviewing my work.

*Ayush Verma
Delft, August 2021*

Summary

Wind energy presents itself as a dominant driver in the energy transition process. However, increased reliance on wind energy, both offshore and onshore, pose a question on their operational performance and reliability. In this regard, the field of wind turbine prognostics and health management has gathered much interest from the scientific community. Due to the increase in data collection, storage and availability, more artificial intelligence algorithms are being developed to process this data and derive meaningful conclusions. While SCADA systems are readily installed in commercial wind turbines, and rich datasets are collected, its use for condition monitoring is still at the early stages. To this end, a 100 Hz SCADA dataset from Control Advanced Research Turbine (CART2) installed at the NREL's Flatirons campus, Colorado, USA is used, and its suitability for condition monitoring and prognostics is evaluated.

The objective of this research is to develop a prognostics framework to predict gearbox failure using SCADA data and identify suitable indicators, sensitive to its condition. The methodology consists of three steps: (a) developing an artificial neural network (ANN) based normal behaviour model to emulate the behaviour of turbine in a healthy state; (b) feature selection and engineering to select suitable indicators to evaluate real operation deviations from the modelled normal behaviour and improve ANN model performance and (c) finally, developing a robust approach using one-class support vector machine (OC-SVM) model to correctly identify the anomalous operation of the turbine caused due to incipient fault and therefore, establish an appropriate threshold to provide enough lead time to plan and conduct maintenance activities. Additionally, a sensitivity analysis with varying SCADA data sampling periods is carried out to determine an optimal frequency for condition monitoring purpose rather than 10 min averaged data.

The results show that a 3-layer feed-forward ANN can efficiently learn the complex mapping between the input and output features, obtaining RMSE values of less than 0.1 and R^2 values greater than 0.95 for each model development phase, namely training, validation and testing. Four residual error features - maximum error, minimum error, root mean squared error and error distribution are used as inputs for the OC-SVM model to understand the complex boundary between normal and anomalous operation. The percentage of anomalies computed for each week of operation, 4 months before failure, show an increasing trend as the turbine approaches failure, with the most significant increase in anomaly rates reported 4 weeks before the failure. To this end, a real-time monitoring scheme based on linear regression and bootstrapped confidence intervals is developed to track the progression of anomalies and set off a maintenance alert as the first indication of incipient fault becomes evident. The scheme alarms for maintenance a month before the actual failure, providing enough lead time to plan and maintain the gearbox.

A sensitivity study is carried out for a range of sampling periods ranging from 100 Hz to 10 min. The results demonstrate that high-frequency SCADA data is beneficial for condition monitoring of the gearbox, but only if the noise in the data can be excluded. On the other hand, despite the loss of information due to the averaging effect for large sampling periods, SCADA data aggregated over a 30 s period could be utilised to predict the gearbox failure a month in advance. Furthermore, the ANN model performance is found to be sensitive to the number of data samples available for training.

List of Figures

1.1	Offshore wind installed capacity growth by country (Europe)	1
1.2	Failure rate and downtime per failure for different WT components	2
2.1	Typical wind turbine drive train	5
2.2	Three-stage WT gearbox with one planetary and two parallel stages	6
2.3	Stop time (downtime) per event for onshore and offshore wind turbines	8
2.4	Costs associated to maintenance strategies	9
2.5	WT health through the progression of time	10
2.6	WT prognostics and health management approaches	13
2.7	Series of steps representing ML process	15
2.8	Confusion matrix of a binary classification	16
3.1	Calculation process within a neuron	22
3.2	The architecture of an artificial neural network (ANN)	23
3.3	Common activation functions used in ANN models	24
3.4	Synthetic sample generation using the SMOGN algorithm	27
3.5	Support vectors and max-margin hyperplane	28
3.6	OC-SVM learned boundary from training data and anomaly identification	29
4.1	Model architecture for gearbox failure prognostics	34
4.2	Normal behaviour model-based monitoring	36
4.3	Bathtub curve showing instantaneous failure rate over the lifetime of a WT component and cumulative failure distribution	38
5.1	CART2 sensor location	42
5.2	Weeks of operation and corresponding dates in which the data was available for healthy data (blue) and weeks to failure and corresponding dates for faulty data (orange)	44
5.3	Unfiltered wind turbine power curve for healthy data with raw data sampled at 100 Hz data and re-sampled at 1 Hz data	45
5.4	Blade pitch vs LSS rotational speed for healthy and faulty data (re-sampled at 1 Hz): (left) in original data showing pitch anomalies; (right) after filtration	47
5.5	Wind speed vs power output for healthy and faulty datasets (re-sampled at 1 Hz): (left) before filtration and (right) after filtration	47
5.6	Data density plot for healthy and faulty data across all power domains	49
5.7	Data density for original and modified training data after SMOGN application	49
5.8	Schematic of three-layer feed-forward neural network	51
6.1	Observed and predicted gearbox oil temperature values for healthy data (top) and power output (bottom)	56
6.2	Distribution of error for all modelling phases of ANN	57
6.3	Observed and ANN-based NBM predictions for gearbox lubrication oil temperature for (a) 12 weeks to failure (b) 11 weeks to failure (c) 8 weeks to failure and (d) 6 weeks to failure	58
6.4	Observed and ANN-based NBM predictions for gearbox lubrication oil temperature for (a) 5 weeks to failure (b) 4 weeks to failure (c) 1 week to failure and (d) 0 weeks to failure (day of failure)	59
6.5	Maximum error vs RMSE - for both healthy and faulty data	60
6.6	Anomalous and normal operation points detected by OC-SVM model considering faulty data	61
6.7	Percentage of anomalies detected for each week leading up to failure using OC-SVM	61

6.8	Comparison of percentage of anomalies detected by OC-SVM and standard statistical thresholds based on RMSE and standard deviation (Std)	62
6.9	Real-time monitoring scheme implemented for monitored weeks (a) 5 weeks to failure (b) 4 weeks to failure (Maintenance alert set off as the percentage of anomalies exceed the 95% CI bounds) (c) 1 week to failure and (d) 0 weeks to failure (day of failure) . . .	64
6.10	ANN model performance: (a) R^2 and (b) RMSE for all model development phases for different SCADA data sampling rates	67
6.11	ANN model performance: Error standard deviation for different SCADA data sampling periods	68
6.12	Heatmap for the percentage of anomalies detected by OC-SVM for different SCADA sampling rates for each week before failure	69
A.1	Entire neural network visualised	75
B.1	Real-time monitoring scheme implemented with SCADA data aggregated over 30 s for monitored weeks (a) 5 weeks to failure (b) 4 weeks to failure (Maintenance alert set off as percentage of anomalies exceed the 95% CI bounds) (c) 1 week to failure and (d) 0 weeks to failure (day of failure)	80

List of Tables

2.1	SCADA systems versus CMS	11
2.2	Relevant SCADA and CMS monitoring signals from a WT gearbox	12
2.3	Summary of SCADA based monitoring methods utilizing ANN-based NBM for wind turbines	18
5.1	Key characteristics of the CART2 turbine	42
5.2	Data overview as collected for each operational year and associated number of raw data points	43
5.3	Summary of the number of data samples in healthy and faulty datasets	48
5.4	Model features for the ANN-based NBM	48
5.5	Key characteristics of ANN model	51
5.6	Data summary for training, validation, testing and implementation of ANN model	52
5.7	Residual error metrics	53
6.1	ANN model performance metrics for each model development phase	55
6.2	Summary of the number of data samples in healthy and faulty data sets for different sampling periods	66

Contents

List of Figures	vii
List of Tables	ix
1 Introduction	1
2 Literature review	5
2.1 Failure modes of WT gearboxes	5
2.2 Maintenance strategies	7
2.3 Wind turbine monitoring	9
2.3.1 SCADA and CMS measurements for gearbox failure analysis.	11
2.4 Approaches for wind turbine prognostics and health management	12
2.4.1 Physics-based model.	13
2.4.2 Data-driven models.	14
2.5 Discussion	19
3 Theoretical background	21
3.1 Artificial neural networks (ANN)	21
3.2 Pre-processing approach for imbalanced regression.	26
3.3 Support vector machines (SVM).	27
3.4 Linear Regression and Bootstrapped Confidence Intervals	30
4 Methodology	33
4.1 Framework for gearbox failure prognostics	33
4.1.1 Module 1: Data pre-processing	35
4.1.2 Module 2: Normal behaviour model	36
4.1.3 Module 3: Anomaly detection	37
4.2 Sensitivity study	39
5 Data analysis & Model development	41
5.1 Module 1: Data pre-processing	41
5.1.1 Data Overview	41
5.1.2 Categorization of healthy and faulty instances	44
5.1.3 Data cleaning	45
5.1.4 Feature selection and Pre-processing.	48
5.2 Module 2: Normal behaviour model	50
5.3 Module 3: Anomaly detection and Prognosis	52
5.4 Discussion	53
6 Results and Discussion	55
6.1 ANN model performance.	55
6.2 Model Implementation	57
6.3 Anomaly detection results	60
6.3.1 Comparison of OC-SVM result with other methods.	61
6.4 Real-time monitoring scheme	62
6.5 Sensitivity study	65
7 Conclusion & Recommendations	71
A Neural network optimization	75
B Real-time monitoring scheme	79
Bibliography	81

Acronyms

AI Artificial Intelligence.	O&M Operation and maintenance.
ANN Artificial Neural Network.	OC-SVM One Class Support Vector Machine.
BTA Boosting tree algorithm.	OLS Ordinary least squares.
CAGR Compounded Annual Growth Rate.	OPEX Operation and maintenance expenditure.
CART Standard classification and regression tree.	PCA Principal component analysis.
CBM Condition-based monitoring.	PHM Prognostics and Health Management.
CI Confidence interval.	PM Performance monitoring.
CM Condition Monitoring.	R² Coefficient of determination.
CMS Condition Monitoring Systems.	RES Renewable Energy Sources.
COE Cost of energy.	RMSE Root Mean Squared Error.
HSS High speed shaft.	RUL Remaining Useful Lifetime.
LSS Low speed shaft.	SCADA Supervisory Control and Data Acquisition.
MAE Mean Absolute Error.	SGD Stochastic gradient descent.
MHD Mahalanobis distance.	SMOEN Synthetic Minority Over-Sampling Technique with introduction of Gaussian noise.
ML Machine Learning.	SMOTER Synthetic minority over-sampling technique for regression.
MSE Mean square error.	SVM Support Vector Machine.
NARX Non-linear auto-regressive exogenous model.	WF Wind farm.
NBM Normal Behaviour Model.	WT Wind Turbine.
NN Neural net.	WTG Wind turbine gearbox.
NREL National Renewable Energy Laboratory.	
NWTC National Wind Technology Center.	

1

Introduction

Renewable Energy Sources (RES) have emerged as a promising alternative to conventional fossil fuels to combat the problems of climate change, producing a transformation in the energy sector. Wind energy stands out among the different energy resources for its excellent characteristics. It is being defined as one of the fastest-growing renewable energy sources globally, with a compound annual growth rate (CAGR) of more than 21% for the past two decades [1]. In Europe, wind energy is growing fast as well and this progress is not projected to stagnate. Recently, the European Commission tabled the 'Fit-for-55' package, a series of legislative proposals aimed at a 55 % reduction in emissions by 2030. This would lead to a major acceleration in the expansion of wind energy, resulting in the installation of 451 GW of wind power capacity compared to the current installed capacity at 180 GW [2]. With such high installed capacities, the energy sector will become more reliant on wind power production. This highlights the problem of wind turbines' reliability.

To achieve such total installed wind capacity goals, both onshore and offshore sectors are expected to grow. However, the future of onshore wind farms is restricted, particularly due to land constraints as well as public resistance to developments near residential areas. On the contrary, offshore wind farms not only have the vast area of ocean for future development and better wind conditions but also experience less opposition as they are located far off the coast, away from where people live. Many European countries are already increasing their installed wind capacity by exploiting the advantages of offshore wind. Figure 1.1 shows the growth of offshore wind capacity in Europe, which has grown tenfold in the past decade, from 2.5 GW to 25 GW. However, offshore wind has some disadvantages as well. Turbines installed offshore operate in harsh conditions - extreme winds, storms, corrosive environment - which can lead to the breakdown of mechanical and electrical components. This reduces the reliability of offshore wind, which is important when considering the large capacities to be installed in the future.

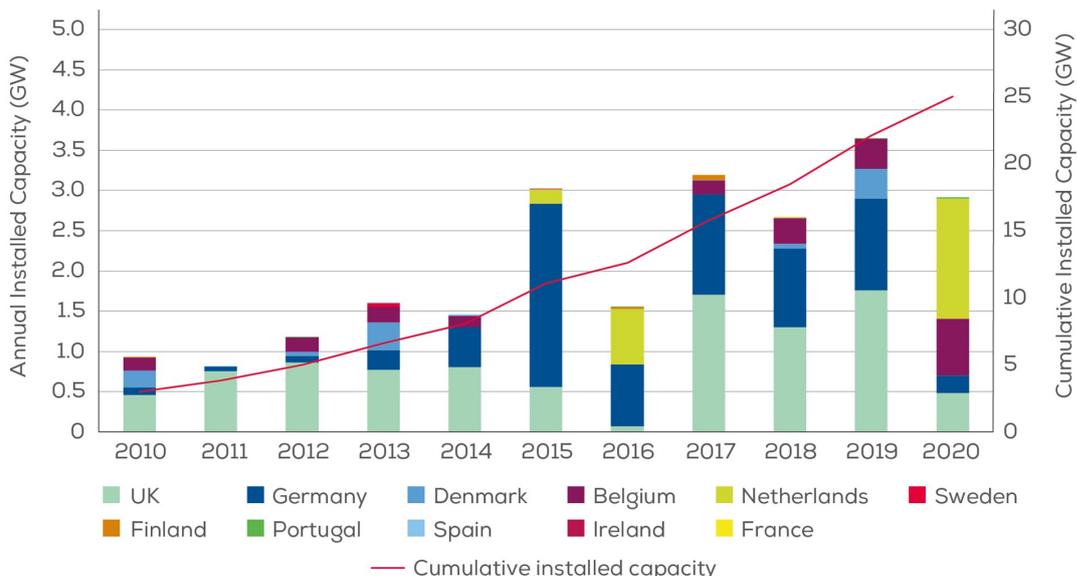


Figure 1.1: Offshore wind installed capacity growth by country (Europe) [3]

Various studies have been conducted to identify the most critical components in a WT. Research for WT reliability suggests that the most critical components are the control system, gearbox, electric system, generator, and hub and blades. Figure 1.2 shows that the gearbox failures not only occur most frequently but also incur the longest downtime. Therefore, the gearbox is the most critical WT component, followed by the generator, rotor blades and pitch system [4].

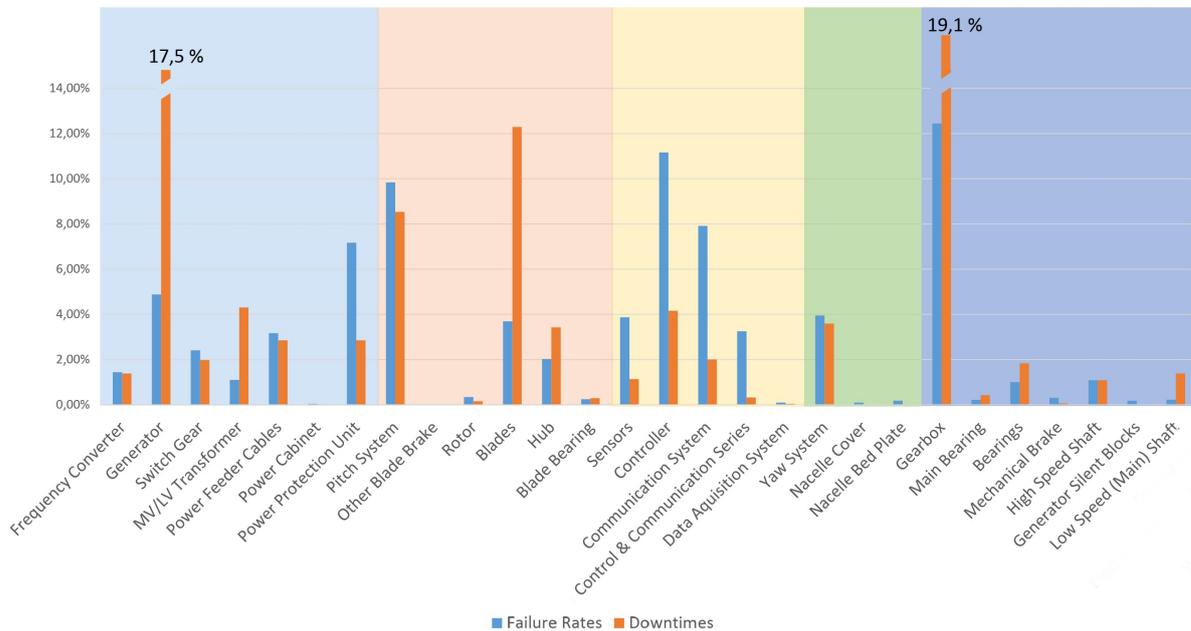


Figure 1.2: Failure rate and downtime per failure for different WT components (adapted from [4])

There are further challenges pertinent to the offshore wind sector related to the transportation, installation and operation of such wind turbines. The difficulty of accessing these WTs and their remoteness from centres where they are monitored, give rise to high operation and maintenance (O&M) costs. Moreover, investigation on O&M costs for wind farms suggests that they could account for 25% to 35% of the total cost of power generation [5]. These O&M cost percentages could increase due to recurring failures in the different components or sub-components of a WT. Therefore, early detection of potential WT failures through condition monitoring (CM) and adopting an appropriate maintenance strategy is paramount to the wind farm owners and operators.

Condition monitoring (CM) can be defined as an "activity, conducted manually or automatically, for observing the actual state of a component which could provide a reliable indication of a failure, so that actions can be planned and downtime is minimised" [6]. Various manufacturers have developed Condition Monitoring Systems (CMS) to monitor several key signals for wind turbines, including drive-train vibrations and oil quality for some of the main components. Data collected using CMS is normally sampled at quite high frequencies (usually in the range of 5-25 kHz). These systems are usually installed as additional monitoring devices to the standard WT configuration. The costs of such systems are usually on the scale of 11,000 Euros per turbine, which has deterred operators from installing such CMS [7]. However, all utility-scale WTs have a standard Supervisory Control And Data Acquisition (SCADA) system. These systems are an extensive data resource, recording more than 200 variables in intervals of 10 minutes, generating rich historical data. Although the commercial use of SCADA data has been limited to monitoring the performance of wind turbines, there is a growing interest in the wind industry to fully exploit the potential of this rich data source. In fact, there are few examples in the industry that make use of SCADA data for condition monitoring and fault detection, but such applications are still at nascent stages [8]. One of the reasons behind this is that SCADA data is sampled at a much lower frequency than the CMS data and collects general information about the turbine's operation rather than component-specific data. Nonetheless, SCADA data being a potentially low-cost solution, requiring no additional sensors and providing a rich data resource, has attracted the scientific community's attention

to develop approaches to interpret valuable information using it for wind turbine CM, thus, fault prediction.

Numerous strategies have been implemented for WT failure detection and prognostics based on physical and data-driven (or a combination of both) models [9]. Machine learning (ML) and artificial intelligence (AI) have shown successful results in different fields of application - including the maintenance sector, to promote data-driven decision making. ML models such as artificial neural network (ANN) can yield research unprecedentedly due to its versatile methodology and abundance of data collected in WT, which is required for ML implementation. Such models can recognise complex patterns in the data and learn the relationship between different variables. With the continual improvements in computational performance and the efficiency of algorithms, there is a general shift in society towards an era of digitalization. The use of extensive information and computation technology will play a vital role in the energy transition in the future[10].

The thesis problem statement is summarized as follows: Energy transition promotes wind energy as a dominant technological driver to mitigate climate change. However, as the reliance on wind energy increases, WT reliability becomes a concerning factor due to the downtime and high O&M costs associated with turbine failures. Adapting an intelligent maintenance strategy using CMS requires a substantial amount of costs and resources. On the other hand, standard SCADA data for WT CM represents a more effective, low-cost and widely implementable solution.

The objective of this report is to develop a framework for prognostics of gearbox failure employing data-driven models based on different ML approaches that capture patterns of fault characteristics using high-frequency SCADA data. The project's scope is as follows: Firstly, to identify suitable SCADA signals, which would be sensitive to changes in the health condition of the gearbox and could be used for early detection of incipient failure. Secondly, to investigate different ML models and develop a framework to flag anomalous gearbox operation and provide enough lead time before the actual failure. Lastly, to perform a sensitivity analysis for SCADA data sampled at different frequencies to identify the optimal sampling frequency for wind turbine gearbox failure prognostics. The models are trained, tested and validated using data collected from Control Advanced Research Turbine 2 (CART2), situated in the NREL's Flatirons campus Colorado, USA and provided by the National Renewable Energy Laboratory (NREL). The data set comprises high-frequency SCADA data sampled at 100 Hz, rather than the conventionally used data averaged over 10 min intervals.

The outline of the report is structured as follows:

- Chapter 2 is a literature review on different maintenance strategies that can be employed, common failure modes of WT gearbox and the approaches used for wind turbine prognostics and health management.
- Chapter 3 provides the necessary theoretical background to understand the principles of different machine learning models and concepts employed in the project.
- Chapter 4 introduces the framework developed for gearbox failure prognostics and describes the methodology adopted in this research.
- Chapter 5 describes the methodology and approach followed to create the model for predicting gearbox failure, along with an overview of the data set used to train and test the developed framework. Moreover, it discusses the feature engineering process and hyperparameter tuning of the model.
- Chapter 6 presents the results of the proposed approach, including model performance metrics. Model sensitivity analysis to datasets of different frequencies is also presented and discussed in this chapter.
- Chapter 7 concludes this research and discusses the key takeaways. Furthermore, recommendations for future work are also compiled in this chapter.

2

Literature review

This chapter aims to provide an overview of the literature explored during the project. It presents a necessary background for understanding failure causes of the gearbox and maintenance strategies to prevent such failure from occurring. As the problem of this thesis is concerned with maintenance prediction for which a data source is required, an overview of employed monitoring systems and characteristics of the data it entails is provided. Also, previous research work that has been done in the field of prognostics and health management of wind turbines is reviewed. The outline of the chapter is divided into four sections: Section 2.1 entails different failure modes of the WT gearbox, Section 2.2 provides a summary of different maintenance strategies that are currently adopted in the wind energy sector, Section 2.3 discusses the monitoring systems and data obtained through such systems, and finally, Section 2.4 provides an overview of the prognostic approaches that have been explored in the available literature.

2.1 Failure modes of WT gearboxes

To conduct proper maintenance to prevent failure, it is important to understand the operation and possible failure causes and modes of the wind turbine gearbox. The gearbox constitutes one of the main components of a wind turbine. A typical wind turbine drive train configuration is shown in Figure 2.1, where the gearbox is placed between the rotor and the generator. One of the main differences between wind turbine gearboxes (WTGs) and those used in other applications is that WTGs step up shaft rotation from low speed, high input torque to high speed, low output torque. In contrast, most other gearboxes operate in the reverse direction. High transient loads are thus absorbed by the gearbox, leading to extreme loading conditions in the bearings, which are relatively unique to wind turbines. Additionally, the variable wind conditions lead to short-duration extreme loading on gearbox components such as gear teeth, a common cause of WTG failure [11].

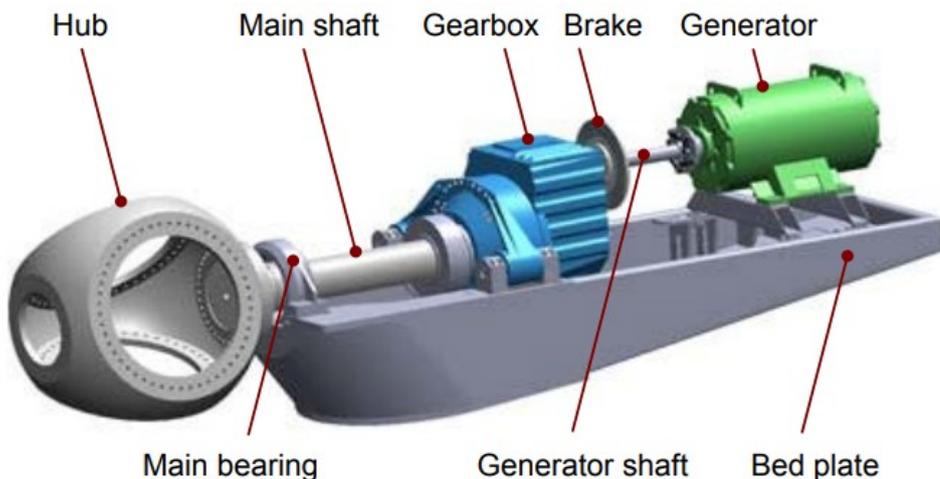


Figure 2.1: Typical wind turbine drive train [12]

The wind turbine gearbox links the low-speed shaft to the high-speed shaft; generally, the large step-up is achieved via three separate stages. The most common configuration is a planetary first stage,

with three planetary gears, followed by two parallel stage gears while all the rotating components are supported by bearings. The lubrication is provided by a system that circulates oil around all gears and bearings. The schematic of a typical wind turbine gearbox is shown in Figure 2.2. In general, most gearbox failures occur due to incipient faults in either the bearings or gears. Failures of gearboxes can be attributed to various reasons, including the following:[13]

- Underestimated design loads;
- Torque overloads;
- Wrong material;
- Manufacturing errors;
- Damage during transportation and assembly;
- Misalignment of components in the shaft.

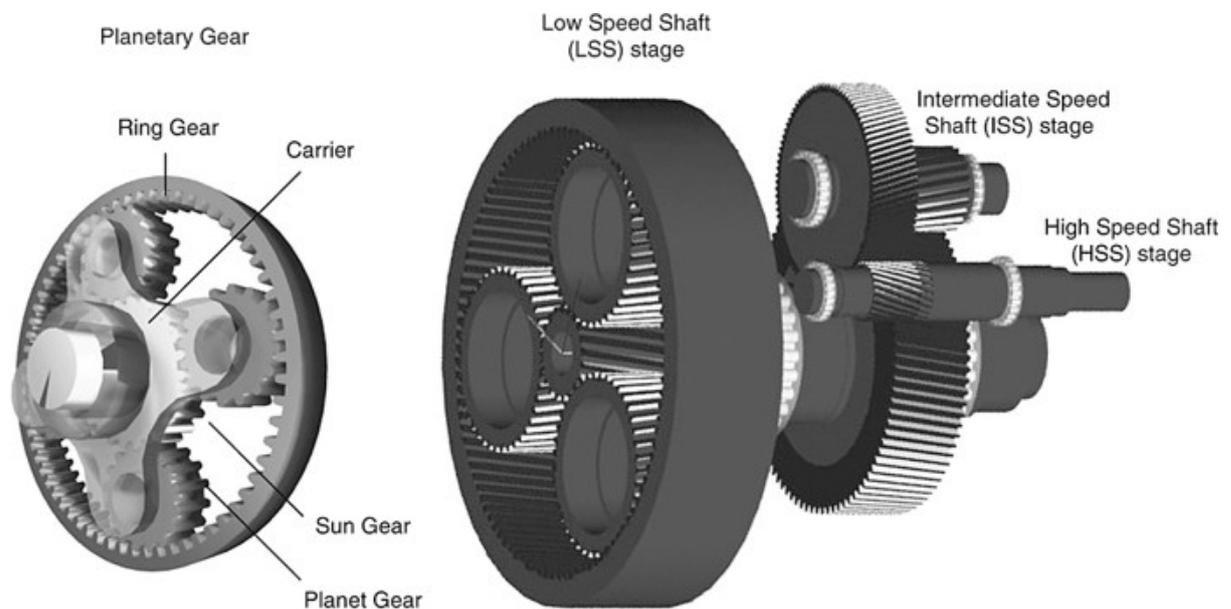


Figure 2.2: Three-stage WT gearbox with one planetary and two parallel stages [14]

Bearings are the components within the gearbox that are more prone to failure. Such breakdowns account for approximately 75% of all gearbox-related failures [15]. WT bearings can be subjected to defects or induced by corrosive, high speed, and high temperature operating conditions. The performance degradation of a bearing is a continuous irreversible process. Four of the most common failure modes are [16]:

- **Axial cracking.** It is caused by improper fits, improper shaft grooving, and microstructural alterations (white etching cracks), characterised by a crack.
- **Scuffing.** This is caused by material transfers from one surface to another under frictional heating. This is a type of adhesive wear.
- **Spalling.** It is caused by deflection and misalignment, inclusions, and defects in the material sub-surface. It is characterized by flaking and pitting material from the raceways and rolling elements.
- **Micropitting.** This occurs due to inadequate lubrication, leading to no separation between the contact surfaces, causing plastic deformation and break-off.

Another critical gearbox component is the gears. Gear failure in wind turbines occurs due to a combination of factors, such as crack initiation and propagation, surface fatigue, surface wear, structural fatigue, and loss of lubrication [17]. The most common gear failure mechanisms are listed below [18]:

- **Fatigue cracks.** This is the most common failure mode and refers to the localized stresses that develop when two curved surfaces get in contact and deform slightly under loads. These contact stresses are cyclic in nature and over-time lead to fatigue cracks.
- **Teeth breakage.** It occurs as a result of gear-tooth deterioration. Such deterioration can take place due to various reasons such as poor design, overloading, misalignment, and tooth surface defects.
- **Abrasive/adhesive wear.** This is caused by small particles through the gear mesh. These could be dust particles, sand, scale from castings, or even debris in lubrication oil. Such an action of one material sliding over another with surface interaction and adhesion can lead to localized contact areas.
- **Scoring (Scuffing).** It is characterized by the rapid metal removal from the tooth surfaces caused by severe adhesion between one tooth and another. Although mild scuffing is non-progressive and does not constitute a primary failure in gears, severe scuffing can lead to plastic deformation of the gear teeth.
- **Surface fatigue.** This is marked by the failure of material as a result of continuous surface stresses. It is characterized by the removal of material and the formation of cavities. Destructive pitting usually starts below the pitch line, progressively increases the size and number of pits destroying the tooth shape.

2.2 Maintenance strategies

The reliability of wind turbines is a critical factor in the economic success of wind energy projects. On the one hand, low levels of reliability might lead to numerous breakdowns that require extensive maintenance, however high reliability might be prohibitively costly to achieve. The maintenance costs are a significant part of wind turbines operation and maintenance expenditure (OPEX). In fact, operations and maintenance costs contribute approximately 25% to 35% to the cost of energy. Moreover, WT reliability affects wind farms' overall performance and power output, resulting in additional costs from lost revenue [19].

A recent reliability study done by Dao *et al.* [20], compared the WT failure rates and downtimes, for both offshore and onshore applications, broken down by sub-assembly. The study included WT failure data from 18,000 WTs, corresponding to over 90,000 turbine-years. The analysis revealed that the average failure rate and downtime per assembly for offshore WTs is greater than that for onshore. Figure 2.3 compares the stop time per event (the duration that the turbine does not generate any power due to its shut-down) for both onshore and offshore applications. On average, the stop time for an offshore WT was found to be almost double that of an onshore WT. This is due to the severe offshore operating conditions and the difficulty in repair / maintenance accessibility. Furthermore, the study showed that generators and gearboxes are highly critical components for offshore wind turbines due to the associated stop time in case they experience a functional failure.

To tackle economic and production losses caused due to unexpected turbine failure, it may be prudent to employ effective maintenance strategies. Different types of WT maintenance are mentioned extensively throughout the literature and synthesized particularly well in [21]. Maintenance strategies can be classified into reactive, preventive, and predictive.

Reactive maintenance strategy is the traditional approach to maintenance and means repairing a machine or a component when failure has occurred. Due to maintenance action planning after shut-down, such a strategy results in the longest downtime before the wind turbine starts operating again and therefore incurs a major cost in the form of production losses. Another potential disadvantage of repairing upon failure is that the machines may fail catastrophically due to the failure of components other than those that initially failed. Considering all of the above, a reactive maintenance strategy is not acceptable when high availability of wind turbines is required.

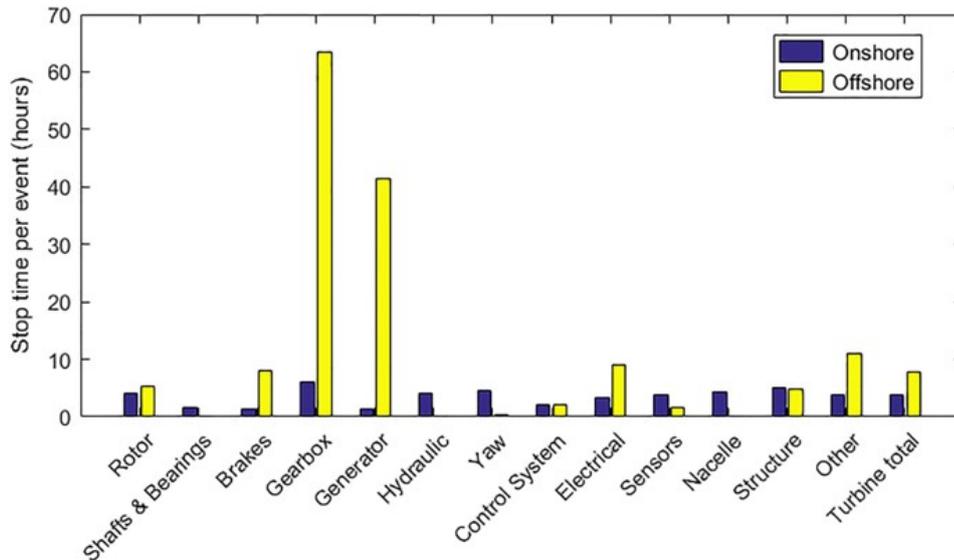


Figure 2.3: Stop time (downtime) per event for onshore and offshore wind turbines [20]

Also known as schedule-based maintenance, **preventive maintenance** is a time-based approach that oversees the replacement of certain critical components at scheduled intervals. Such interventions are periodically undertaken at intervals shorter than the estimated time between failures. Albeit its advantages, the main disadvantages of this strategy are: (1) the time-to-failure of a certain component must be accurately predictable; (2) systematic replacement of "healthy" components yields an excessive consumption of unnecessary resources and (3) the frequency of maintenance activities is quite high to ensure success. Furthermore, many unneeded maintenance activities might introduce new faults in the machine, and several unforeseen failures could still occur. When considering a wind turbine operating offshore, such a maintenance approach would lead to high costs and resources, and hence, a different strategy for scheduling maintenance would be preferable.

Predictive maintenance, which is also referred to as condition-based maintenance, seeks to carry out activities at an optimal time, neither too late, i.e. after a component has already failed, nor too early - when the component still has a significant remaining useful life. Through regular condition monitoring, this strategy aims to predict the potential breakdown of a component. Predictive maintenance can be exercised through condition-based monitoring (CBM). CBM is defined as the process of monitoring a machine parameter or a set of parameters. When a significant deviation from expected behaviour is observed - the development of a failure is identified. The main advantages of CBM are: (1) significant reduction in the probability of catastrophic failure by early fault detection; (2) accurate prediction of the component's health for more cost-effective maintenance activities; (3) valuable insights on root causes of failure as well as reliable input for component design and improved operation of machinery; and (4) reduced downtime associated with turbine failure ensuring higher availability. While additional engineering activities are needed before physical maintenance, the advantages gained propose an attractive alternative to reactive or preventative maintenance strategies [22].

The aforementioned maintenance strategies can be compared from an economic perspective. The cost of energy (COE) produced from wind farms can be expressed as [23]:

$$COE = \frac{ICC * FRC + OM}{AEP} \quad (2.1)$$

where, for a given annual energy production AEP (€/year); ICC (€) is the initial investment cost; FRC (%/year) is the annual fixed charge; and OM (€/year) is the annual COE cost.

The costs associated with each maintenance strategy are presented in Figure 2.4 as a function of the number of failures. In a reactive maintenance strategy, many faults might occur, which would lead

to a high cost of repair and a low cost of prevention. On the contrary, in the case of preventive maintenance strategy, the number of failures is considerably reduced while the cost of preventing makes it expensive. As seen in the figure, a predictive maintenance strategy is optimum in improving the reliability, availability, and maintainability of WT's while simultaneously reducing the variable costs [23] [24].

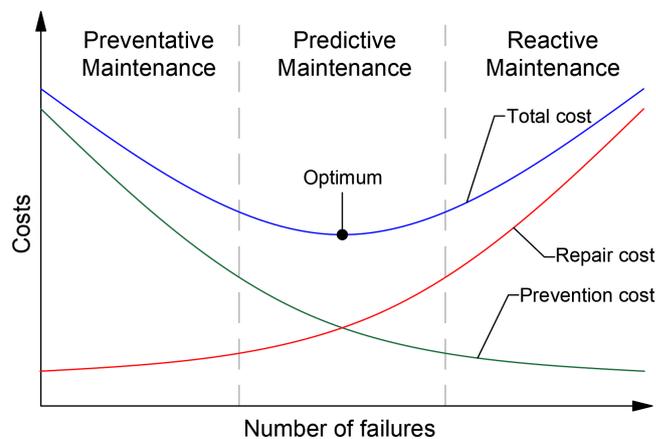


Figure 2.4: Costs associated to maintenance strategies, *Adapted from* [6]

2.3 Wind turbine monitoring

Predictive maintenance is based on wind turbine monitoring and aims to perform maintenance activities before an incipient fault develops into a functional failure. WT monitoring means supervision of the machine health through recorded measurements of physical parameters from the sensors installed on the machine. The signals received from the WT are analyzed to detect deviation from normal operation of mechanical components such as bearings, gears, shafts, etc. Comprehensive knowledge of fault development and its localization within a certain machine component can be gained by analysing such deviations. In a broad sense, the main purposes of a WT monitoring system are:

1. **Detection**, where the occurrence of a fault in the machine component or sub-component is identified. This is essential as the detection of a failure is a prerequisite to taking any preventive action.
2. **Diagnosis**, which is different from detection as it provides more detail on the nature of the fault and the precise location, which can help prepare better to plan any maintenance activities if required.
3. **Prognosis**, which is the prediction of the remaining useful lifetime or time to failure of a machine component. Based on this knowledge, an effective maintenance strategy can be employed, which would lead to an efficient allocation of cost and resources.

In Figure 2.5 a characteristic curve of machine health as a function of time is shown. The curve shows that the machine performance or condition declines over time, eventually leading to a functional failure. This exhibits a sense that failure is a process, not an event. Many of the failure modes take time to develop to a level or degree that causes the functional failure of the machine. WT monitoring is concerned with tracking this progression and predict failures before the functional failure occurs. The curve shows a linear degradation path from the point when the failure starts to occur to the point when the first signs of failure can be detected. Afterwards, the machine health decays exponentially to the point where a functional failure occurs. 'Fault detection' is concerned with identifying criteria that determine when the functional failure occurs, whereas 'Fault prognosis' focuses on analysing when the first signs of failure can be detected and corrective actions can be implemented to prevent the failure. Although extensive research has been done on fault detection and diagnosis, the work on fault prognosis is still at quite an early stage.

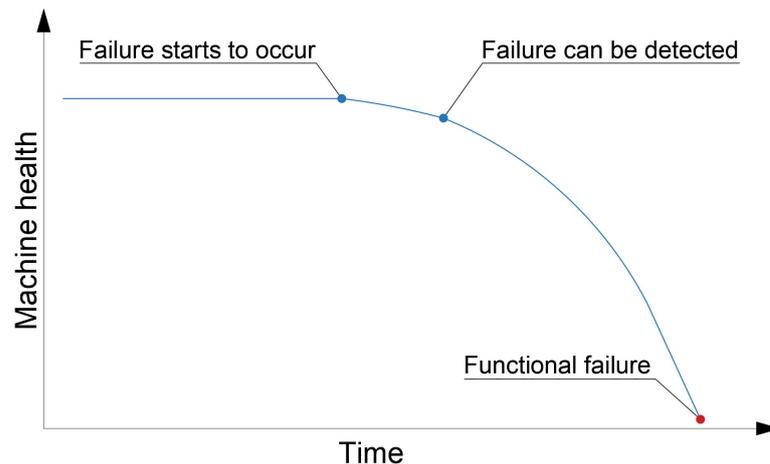


Figure 2.5: WT health through the progression of time (adapted from [25])

WT monitoring and prognostics can be achieved using monitoring systems that measure and record signals which are sent to the operator to allow remote control and management of WT operations. Two monitoring systems are currently employed in the commercial market [26]:

1. **Supervisory Control and Data Acquisition (SCADA) system.** SCADA systems were initially designed for operating purposes only and used by operators to supervise and control WT systems remotely. The parameters monitored by a SCADA system, in general, include active and reactive power output, generator current and voltages, along with temperature measurements of various critical machine components such as that of lubricant oil, bearing, nacelle, etc. These systems usually record signals sampled at high frequency (1 Hz) and store as 10 min averaged values.
2. **Condition Monitoring System (CMS).** Based on the experience of monitoring rotational equipment, the parameters recorded by a CMS are mainly vibration-based signals, although some are used in combination with oil particle counts. These systems focus on remote measurement of critical indicators of the health of WT components, seeking to identify incipient faults before they lead to a catastrophic failure. These are measured at a higher frequency compared to the SCADA data ranging in scales of kHz. CMSs can either be classified as an online system or an offline system. Online CM is carried out by constantly monitoring the components with the help of equipment that continuously transmits measurements to a data server. On the other hand, offline CM requires a posterior laboratory analysis.

The two monitoring systems each have different advantages and disadvantages, which are summarised in Table 2.1 [27][9]. The suitability of SCADA systems for monitoring WT performance has led to these systems being installed in many wind turbines by default, providing an available data source without any additional cost. In contrast, CMSs are expensive and require expert supervision to analyse and derive conclusions from such a data source. Commercially available WT CMS cost over €10K (e.g. SKF WindCon costs around €14K), which would cost a wind farm (WF) developer (for a WF consisting of over 100 WTs) millions of euros to equip each WT with such a system. Also, CMS data is usually sampled at 20kHz, which results in an amount of data 12 million times more than the one stored by SCADA systems. This poses difficulties in storing and analysing data, and requires expert engineers [28]. However, using such a dataset allows for fault isolation and diagnostics, which is not possible when using SCADA data as it records signals that provide general information about the turbine used mostly for performance monitoring (PM) [29]. PM employs SCADA systems to train models for normal operating states and utilize them to detect outliers or abnormal behaviour of WT. Nevertheless, currently, with the motivation of potentially reducing the costs of employing a purpose-based CMS, efforts are being taken to develop robust algorithms to use SCADA measurements for failure detection and prognostics [30].

Although SCADA system is a reliable data source collecting extensive information about the wind turbine performance, its applicability to monitoring the condition of the wind turbine components is still

Table 2.1: SCADA systems versus CMS

Monitoring Systems	Advantages	Disadvantages
SCADA systems	Readily accessible, capable of identifying normal behavior and outliers through performance monitoring	Incapable of fault isolation and diagnostics
CMS systems	Capable of fault isolation and diagnostics, capable of capturing high frequency dynamics usually not achievable using SCADA system	Requires extensive sources and expertise to perform data analysis and result assessment, expensive instrumentation

limited. A reason for this is the low temporal resolution (10 min) of the data collected, along with the negative effects of averaging for failure detection and prognostics approaches utilising these systems. Wind turbines are subjected to dynamic loading conditions because of the rapid wind speed fluctuations, which do not get reflected in the averaged signals. Hence, intermittent or transient anomalies cannot be detected [31] [32]. To capture the dynamic nature of the wind turbine operation and improve the detection capabilities, high-frequency SCADA data can be used. Recently, there have been attempts to investigate WT performance monitoring (for example, power curve modelling) using SCADA data at high resolution [33][34][35]. However, only a few attempts have been made to use this high-frequency data for condition monitoring [36].

2.3.1 SCADA and CMS measurements for gearbox failure analysis

SCADA systems provide a large number of signals which could be used to extract information about the condition of the WT components. As this study is concerned with gearbox failure, it is important to note that the most common gearbox-related SCADA measurements are as follows [37]:

- Gearbox lubrication oil temperature
- Gearbox bearing temperature
- Gearbox oil pressure
- Low/High-speed shaft torque and rotational speed

Wind turbine gearbox-related CMS signals originate from two sources: vibration transducers and oil debris counters. The accelerometer positions for gearbox monitoring can vary; however, a common configuration for recording vibration-based CMS is [38] [28]:

- Low-speed shaft (LSS) end, transverse
- High-speed shaft (HSS) end, vertical
- High-speed shaft (HSS), transverse
- High-speed shaft (HSS), axial

Furthermore, a gearbox oil debris counter can be used to record both ferrous and non-ferrous particle sizes circulating the gearbox lubrication oil. By examining the particle counts for both ferrous and non-ferrous particles, useful information for diagnosing the development of any fault in bearing rollers, cages, or gears can be extracted. Table 2.2 summarizes the commercial SCADA and CMS measurements useful for monitoring a WT gearbox monitoring.

Table 2.2: Relevant SCADA and CMS monitoring signals from a WT gearbox [14]

Monitorable failure modes	Planetary gear failure	Planetary bearing failure	ISS bearing failure	HSS bearing failure	Lubrication system malfunction
SCADA signals	Oil temperature	Oil temperature	Oil temperature	HSS bearing temperature	Oil pressure level, oil filter status
CMS signals	LSS vibration; non ferrous particle oil debris count	LSS vibration signal; ferrous particle oil debris count	LSS or HSS vibration signals; ferrous particle oil debris count	HSS vibration signals (vertical, transverse, axial); ferrous particle oil debris counts	
Additional signals	Rotor speed; generator speed; nacelle temperature; power output; wind speed				

2.4 Approaches for wind turbine prognostics and health management

The main goal of prognosis is to evaluate how long a faulty component can work under reliable operating conditions while still achieving desired performance metrics [39]. To evaluate the health of a wind turbine drive train during its operational life, a suitable monitoring system should be put in place as a tool for prognostics, as discussed in the previous section. While data retrieved from the monitoring systems is the foundation for prognostics (and detection and diagnostics), the methods used to extract relevant information from the collected data are as much if not more important.

There are many approaches to evaluating wind turbine prognosis. Prognostic methods are usually classified into three main distinct categories: physics-based approach, data-driven approach, and hybrid approach. Data-driven models are further subdivided into two categories: stochastic models and machine learning models. The classification is demonstrated in the Figure 2.6. All of these methods can potentially employ either SCADA or CMS signals or even a combination of both.

Physics-based approaches rely on understanding the governing physics of the system and its degradation mechanism. The model can derive mathematical relations that describe the deterioration process, provided that sufficient knowledge on the system components and sub-components, their operational behaviour, and abnormal functioning and failure modes are available. The major drawback of employing this model is that each failure mode has to be analyzed independently, and for many failure modes, a clear degradation mechanism is not present [40]. Even if a certain failure mode can be associated with a specific degradation mechanism, its explanation through a physical model is not always possible. Albeit the disadvantages of this approach, a relevant advantage is that it establishes synergies with the design process and help optimize future designs by understanding failure mechanisms. Also, if a physical model explaining a certain failure mechanism is achieved, it would aid in identifying the fault localization and further degradation using smaller data sets.

Data-driven approaches do not make use of any underlying physics that governs the system and uses the information within data collected from a machine instead. Despite the lack of knowledge of physics dictating the machine's operation, data-driven approaches have been proven to be efficient and successful for fault detection and prognosis tasks by estimating the machine's remaining useful

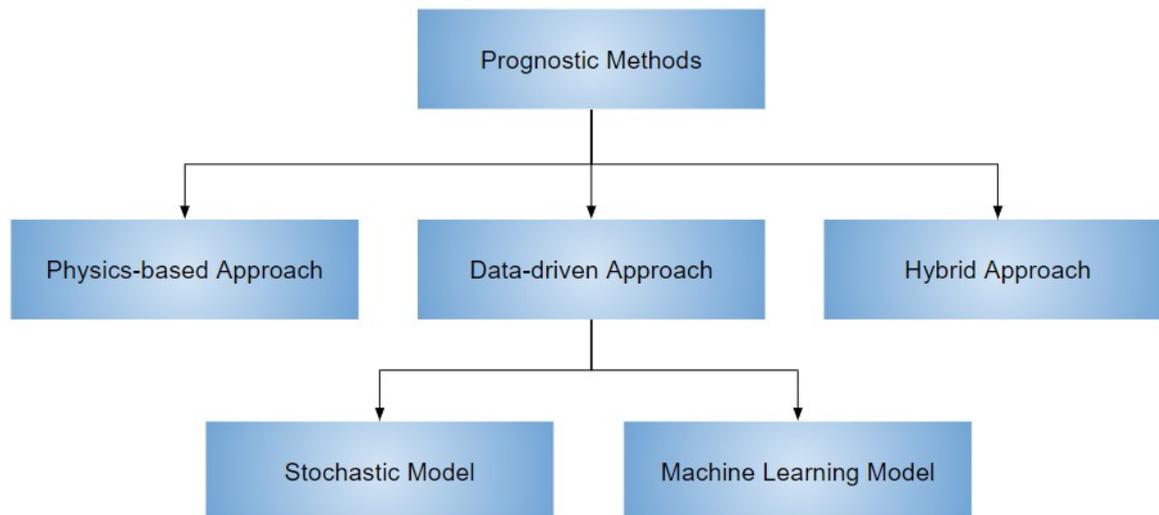


Figure 2.6: WT prognostics and health management approaches

lifetime (RUL) [37][41][42][43]. This is because the physics of failure associated with WT drive trains are often too complicated to model and interpret. This highlights the advantages of employing data-driven models, which tend to learn the patterns within the data to identify failure characteristics of the machine and overcomes the limitations inherent to the lack of physical knowledge. One of the main disadvantages of this modelling approach is that it requires relatively large data sets to represent the full operational range of WT and its variability for training, making it computationally intensive. Also, AI models are considered *black-box* models due to their lack of transparency which makes them difficult to comprehend [44].

Hybrid approach is a term used to define any given combination of the aforementioned methods to incorporate the strengths of both of these approaches and develop a robust model for fault prognosis.

2.4.1 Physics-based model

Physics-based prognostic models attempt to construct mathematical models to describe failure progression, e.g. degradation indicators based on decreased gearbox efficiency, spall progression, and crack growth. Firstly, system and subsystem configurations and their respective material properties are defined. Then, potential failure modes and their causes in terms of the failure physics are identified at the individual component level, associated with operating and environmental conditions under which failure is likely to occur [45]. In the following, some of the physics-based prognostics models proposed in existing literature are reviewed.

Feng *et al.* [14] proposed a thermodynamic model to understand the cumulative high and low cycle fatigue damage caused by stochastically varying torque imposed in wind turbines. This was achieved by monitoring the transmission efficiency and rotational speed and relating them to gearbox temperature rise (with respect to ambient temperature). Results showed that gearbox oil temperature rise could indicate a decrease in the gearbox transmission efficiency. By observing the temperature rise trends against binned power output and rotor rotational speed, the deterioration of the gearbox was visible at least 3 months before a planetary gearbox failure.

Gray and Watson [46] employed a mathematical model for WT gearbox damage calculation for a specific failure mode, i.e. bearing high cycle fatigue damage due to edge loading. To do so, they firstly identified different failure modes, their causes, and the damaging operating conditions of the WT gearbox, after which the expected reduction in the bearing life was estimated based on the Lundberg-Palmgren rule. The efficacy of the proposed method was demonstrated through an experimental study on six wind turbines, which experienced severe gearbox failure, the cause for which was recorded as

heavy debris in lubricating oil.

Butler *et al.* [47] utilized SCADA data, including hydraulic brake temperature, blade pitch position, main shaft rotational speed, and hydraulic brake pressure, to formulate a mathematical model to describe a WT main bearing degradation progression. The study applied a particle filtering technique to address the mathematical model and future load uncertainties to enhance the WT main bearing RUL projection, revealing strong evidence of failure with a 30 day lead time.

The examples provided above show that physics-based models for WT prognostics and health management can accurately predict failure in WT components or sub-components. One of the advantages of the physics-based approach, compared to the data-driven approach, is that it does not require extensive amounts of data to develop a predictive model. However, their application is limited as it is difficult to capture the dynamic working operation of wind turbines with a set of mathematical equations, which assume the physics of the model to be consistent across the component. Moreover, these models are defect-specific and often complex to design.

2.4.2 Data-driven models

Data-driven models tend to identify the underlying patterns (or set of structures) in the data and use it to understand relationships between system variables without knowledge of physical behaviour. It can be defined as a process of building an inductive model that learns from a limited set of data without any specialist intervention. Data-driven models can be broadly classified into two categories: stochastic and machine learning-based models. In the next sections, these two approaches are discussed in more detail.

Stochastic-based prognostic techniques

These techniques have been extensively explored in the field of PHM. Various techniques such as Markov chain, Gamma process, and Wiener process have been frequently used to characterize identifiable degradation trends [48]. Furthermore, stochastic models based on linear or polynomial regression models have been employed to construct a normal behaviour model (NBM), which is then used to detect anomalies based on signal residuals between normal and faulty operations [49].

Hu *et al.* [50] developed a performance degradation and a real-time RUL model based on temperature characteristic parameters. Firstly, the temperature trend data was obtained using the moving average method and accounting for the uncertainty of wind speed and wind direction that causes the temperature of wind turbine bearings to vary widely. The performance degradation model was then developed using the Wiener process and its parameters were calculated using the maximum likelihood estimate. Finally, the RUL prediction model was built on the principle of temperature monitoring value exceeding the first warning threshold based on an inverse Gaussian distribution function. The results demonstrated that the model performs well for periods closer to the actual failure, with a prediction error of 2-3 days.

Son *et al.* [51] proposed a probabilistic prognosis model by first creating a degradation indicator using principal component analysis (PCA) and then using the Wiener process degradation modelling to capture the noise and the non-monotonic trend of the degradation indicator. After identifying appropriate features from a list of 21 sensor signals, PCA was used for dimensionality reduction, thus reducing the number of features from 7 to 2. The data set was then split into six parts, each representing different operational modes of the machine. For each operational mode, the failure space in the two-dimensional principal component space was defined. The degradation indicator was based on the Euclidean distance of remaining data points from the barycenter of the failure space points. Therefore, the value of the degradation indicator tends to 0 as the machine approaches failure. The Wiener process was selected as the stochastic model to map the machine's RUL. Based on the trend observed for the degradation indicator, a non-linear drift parameter optimization was employed. The proposed model could obtain better RUL predictions than the models which used the same data set [52].

Machine learning-based prognostic technique

Machine learning (ML) models learn the underlying patterns in the data which are useful in understanding the relationship between different variables, which can then be applied to a dataset different to the one on which the learning occurred [9]. In the taxonomy of ML models, *supervised learning* predicts output variables using labelled input data. In contrast, *unsupervised learning* draws inferences from the data without using any labelled input (such as clustering algorithms). Model learning translates into fitting model's parameters to a specific data set, iteratively updating the parameters while continuing the process for several cycles using the same data until a specific predefined function is minimized. Figure 2.7 shows the ML process represented as a series of four sequential steps. The first step is to acquire all relevant data and clean it by removing erroneous data points (outliers) as these may severely affect the model performance. Once the data set is ready for further analysis, the next step is to identify the most sensitive features pertinent to the task at hand. Consequently, an appropriate machine learning model is selected. There are two main groups of ML models for fault detection and prognostics [29]: classification / prediction and regression-based anomaly detection. Model selection from these two categories depends on the problem the model is used to solve. The model is then trained and its performance is validated using relevant metrics which are specific to the task performed, for example, a classification model can be validated using metrics such as accuracy, model recall, precision, while a regression model utilises validation metrics such as mean absolute error (MAE), root mean squared error (RMSE), coefficient of determination (R^2).

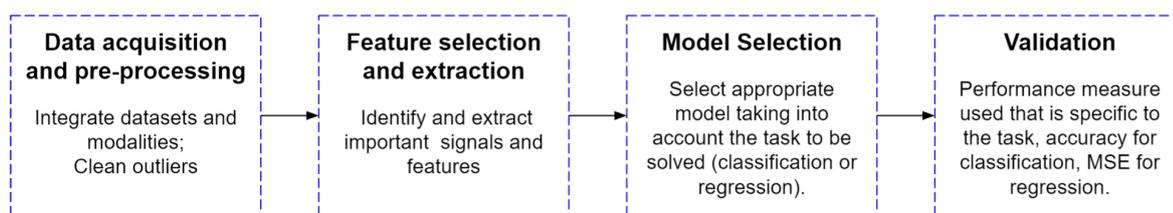


Figure 2.7: Series of steps representing ML process

Classification models include data pre-processing (dealing with missing data, outliers), classes equalization (ensuring balanced classes so that model is balanced), feature selection and extraction, classification model fitting and cross-validation (to test the model's generalizability). Several condition monitoring applications have been explored using classification. Using SCADA data, Kusiak *et al.* [42] built models that could identify / predict faults at different granularity levels (fault and no-fault prediction; fault category (severity); and specific fault prediction). They reported that neural net (NN) ensembles outperformed boosting tree algorithms (BTAs) and support vector machines (SVMs) when building level 1 models (that discriminate at higher granularities: failure / status). For level 2 models (identifying the category of status and failures), CART (standard classification and regression tree) was identified as the most accurate, followed by SVMs and BTAs. At the granularity of level 3, models identify specific types of statuses and faults; BTAs were identified as the best.

Zhao *et al.* [43] introduced a new anomaly operation index (AOI) notion to quantitatively measure the wind turbine's historical performance and predict future performance based on a statistical model. They proposed a data-driven anomaly detection technique, which adopts a clustering method (Density-Based SCAN) to distinguish anomaly data and normal data from unlabeled historical SCADA data, and a classification method (SVM) to classify anomaly AOI and normal AOI in run-time. Finally, using Auto-Regressive Integrated Moving Average (ARIMA) to analyze real-time AOIs, prediction of future AOIs, and estimation of RUL can be achieved. The proposed model was evaluated using wind farm data with 33 wind turbines, demonstrating that it could achieve sufficient lead time (10 - 20 days) for wind farm operators to schedule maintenance before generator failures occur.

To evaluate the performance of a classification model, various metrics can be used. Some of the most widely used metrics are shown below in Equations 2.2-2.6 and these are derived from the following

observations [9] as shown in Figure 2.8. A confusion matrix is a summary of predicted results of a classification model in a specific table layout that allows visualization of its performance measure. It can be used for both a binary classification problem (2 classes) or a multi-class classification (more than 2 classes).

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 2.8: Confusion matrix of a binary classification

- TP (true positive), can be interpreted as the model predicting positive class and it is true.
- FP (false positive), can be interpreted as the model predicting positive class but it is false.
- FN (false negative), can be interpreted as the model predicting negative class but it is false.
- TN (true negative), can be interpreted as the model predicting negative class and it is true.

Accuracy is simply defined as the ratio between the correctly classified points to the total number of data points. It is shown in the Equation 2.2. Although accuracy is simple to calculate, it has its own limitations - (1) when working with highly imbalanced datasets (i.e. majority points belonging to only one class), the model assigns all data points to the majority class and the reported accuracy is high, making it an unreliable metric for evaluating model performance (2) accuracy provides the probability of the predictions of the model and therefore, it cannot measure how good the predictions of the model are.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.2)$$

Precision is defined as the fraction of correctly classified instances from the total classified instances labelled as positive and is expressed as Equation 2.3. Improving the precision is appropriate when the focus is on minimizing false positives (FP).

$$Precision = \frac{TP}{TP + FP} \quad (2.3)$$

Recall indicates how many observations with the positive classes are correctly classified as positive. This is defined as the ratio of TPs to the total number of observations that correctly belong to the positive class and is given by Equation 2.4. Improving the recall of a model is advantageous when the focus is on minimizing the false negatives (FN).

$$Recall = \frac{TP}{TP + FN} \quad (2.4)$$

F1score combines both precision and recall into one measure and is obtained by their harmonic mean, given by Equation 2.5. This metric has been often used to define information retrieval, document classification and query classification performance [53].

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (2.5)$$

Specificity is defined as the ratio between correctly negative labelled points and all data points that are positive in reality and is expressed in Equation 2.6. This is used as a measure when all TNs must be covered, i.e., when no false alarms can be tolerated [54].

$$Specificity = \frac{TN}{TN + FP} \quad (2.6)$$

Regression-based approaches models normal behaviour of different components and sub-components when they are assumed to be in a healthy state. Based on given inputs (independent variables such as wind speed, power output, shaft torque etc.), regression models are built to predict the numeric output of a target or dependent variable when the components are assumed to be performing at their optimum. Ideally, NBM of the components should be constructed with data recorded during the period when the likelihood of failure is low. Mathematical modelling methods like artificial neural networks (ANNs) have been frequently utilized to analyse wind turbine data [30]. ANNs can be applied to monitor any component without an in-depth prior knowledge of its working principles. In addition to being general, the ANN method is easily scalable for application on a large set of turbines. ANN models can handle noisy and missing data, and, once trained, they can aid in prediction and generalization. Furthermore, they are highly efficient at modelling non-linear complex systems. These advantages of the ANN-based approaches have made them of interest for monitoring critical components [55].

One of the earliest works using ANN normal behaviour models for condition monitoring for wind turbine applications was presented by Garcia *et al.* [41]. A software tool called Intelligent System for Predictive Maintenance was proposed. It was divided into six modules responsible for normal behaviour modelling, anomaly detection, health condition assessment, failure diagnosis, preventive maintenance scheduling, and maintenance effectiveness assessment. The study concluded that artificial intelligence and modelling techniques (such as genetic algorithms, fuzzy logic) are sufficient for developing a predictive maintenance strategy.

Zaher *et al.* [37] employed a similar auto-regressive ANN model in case studies with real wind turbine gearbox bearing data. The results showed that the anomalies in the gearbox operation could be tracked 6 months before the eventual replacement took place. The paper proposed a multi-agent system which would only extract useful features about all WT components combined from the data and present them to the operator. The anomaly detection was achieved by observing an increase in the magnitude of error between the predicted and measured output parameters. However, such a detection method based on static thresholds from one turbine can be difficult with many different turbines in a wind farm, and automatic threshold values are desirable to generate alarms from the CMS. Kusiak and Verma [56] investigated the ANN normal behaviour models with different numbers of neurons in the hidden layer and then selected the best performing model for predicting faults. The results showed a prediction timeline of 1.5 h before the eventual failure, however, detection of an incipient fault so close to the actual failure does not allow any kind of maintenance planning. Bangalore *et al.* [57] improved the ANN model using a non-linear auto-regressive exogenous model (NARX) to build a monitoring system based on SCADA data and focused on data pre-processing and post-processing techniques relevant to SCADA data sets. The anomaly detection method proposed in the paper used Mahalanobis distance (MHD), which improved the anomaly detection by considering the correlation between ANN model error and operating conditions. The research also demonstrated that to formulate a NARX model to exhibit NBM, continuous data with no missing inputs is required. This is because the NARX model uses a feedback loop where the output from the previous time instant is required to estimate the output at a given time. Nevertheless, when working with data from a real WT, it is almost impossible to have a dataset with continuous inputs.

Turnbull *et al.* [58] developed an NBM using an auto-regressive feed-forward ANN model, which was executed separately for SCADA and CMS vibration data for the same turbine to establish a more robust method to identify gear faults and predict its remaining useful lifetime (RUL). In the first case, SCADA features such as wind speed, power output, high-speed shaft rotational speed, ambient temperature, and bearing temperature, was selected to model gearbox oil temperature. Various residual error features were then used to describe the error distribution over a chosen time period and selected to train a SVM classifier model to detect anomalies in the measured gearbox oil temperature. The second

case study used a combination of SCADA and vibration features to model NBM and predict anomalies using the SVM model. The benefit of using the SVM classifier was two-fold: (1) SVM anomaly detection model was able to combine different error features into a single threshold, and (2) SVM was capable to learning the complex decision boundary between the normal operation and faulty operation points, therefore, providing accurate predictions of failure 14 weeks before the failure.

Table 2.3 compares some important aspects of methods proposed in the literature that employ ANN models for wind turbine NBM.

Table 2.3: Summary of SCADA based monitoring methods utilizing ANN-based NBM for wind turbines

	Garcia <i>et al.</i> [41]	Zaher <i>et al.</i> [37]	Kusiak <i>et al.</i> [56]	Bangalore <i>et al.</i> [57]	Turnbull <i>et al.</i> [58]
Method used	Multilayer feed-forward auto-regressive ANN	Multilayer feed-forward auto-regressive ANN	Multilayer feed-forward ANN	NARX ANN	Multilayer feed-forward auto-regressive ANN
Input selection	Domain knowledge	Domain knowledge	Data mining algorithms	Domain knowledge	Domain knowledge
Anomaly detection	Confidence based on ANN model training data	Observation of increase in frequency of error	Observation of errors	Based on a threshold derived from statistical distance measure (Mahalanobis distance)	SVM classification based on different residual error features
Failure prediction interval	26h	4-6 months	1.5 h	2-3 months	14 weeks

The performance of regression-based NBMs can be expressed through several measures based on predicted output $\hat{Y}(i)$ and actual output $Y(i)$. Some of the most common measures for evaluating regression models are described below in Equations 2.7 - 2.10. The higher the difference between the actual output $Y(i)$ and the predicted outcome $\hat{Y}(i)$, the more inaccurate the model is in representing the phenomenon; the closer the values and the better the performance of the model is.

Mean absolute error (MAE) calculates the average of the absolute values of the error. It does not provide any information on the direction in which the actual and predicted values differ but only the absolute distance. Also, it does not penalise larger errors more than the smaller ones.

$$MAE = \frac{1}{N} \sum_{i=1}^N |\hat{Y}(i) - Y(i)| \quad (2.7)$$

Mean absolute percentage error (MAPE) measures the error between actual and predicted outcomes as a percentage. It achieves so by calculating it similar to MAE but also dividing by the actual value, as shown in Equation 2.8. By computing the error as a percentage, it provides a better understanding how the model predictions deviate from the actual values in relative terms.

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{Y}(i) - Y(i)|}{\hat{Y}(i)} * 100 \quad (2.8)$$

Root mean squared error (RMSE) calculates the average of squared errors across all samples and takes the square root of the results. It is represented in the Equation 2.9. Because the errors are squared, it penalises the prediction values farther away from the actual values.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{Y}(i) - Y(i))^2} \quad (2.9)$$

R squared or the coefficient of determination represents the proportion of the dependent variable y that can be explained by the independent variables x . For instance, if R^2 of a model is 0.75, then approximately 75% of the observed variation can be explained by the model features. It is calculated by taking one minus the sum of squares for residuals divided by the sum of squares, as shown in Equation 2.10.

$$R^2 = 1 - \frac{\sum_{i=1}^N (\hat{Y}(i) - Y(i))^2}{\sum_{i=1}^N (Y(i) - \bar{Y})^2} \quad (2.10)$$

Overall, it is usually important to report both measures, R^2 and RMSE. This is because the R^2 expresses the relation between features x in the model and the target variable y . RMSE, instead, expresses how the data points are spread out with regards to the regression fit.

2.5 Discussion

This chapter highlighted that even though the scientific community has shown increasing interest to study in condition monitoring of WTs, the research on fault prognosis is still in nascent stages. Keeping in mind that the gearbox was identified as one of the most critical WT components, this study is focused on developing a framework for performing gearbox fault prognostics and health management. From the possible WT PHM approaches discussed in this chapter, a data-driven approach was selected. Physics-based approaches were not explored due to their limitation in capturing the dynamic working operation of a WT and specificity to a failure mode.

Initially, in this project, stochastic approaches utilizing distance-based PCA and Wiener degradation process were explored. Both methods make use of the statistical trends present in relevant signals [50] [51]. However, these approaches did not result in success. Models based on statistical approaches have poor generalization capabilities meaning they are specific to the dataset. The Wiener degradation model requires a specific trend in the target signal which is not always present within a dataset. Distance-based PCA is applicable when the abnormal data is of significant distance from the normal operational data when represented in principle component space. In absence of specific statistical characteristics in the target signals, the stochastic approaches have limited application. Furthermore, to capture complex non-linear relationships that exist within the data increased mathematical complexity of the model is required, which is difficult to design.

Owing to the failed attempts of adopting a stochastic approach, a ML-based approach was employed. Classification models perform poorly when the dataset has highly imbalanced classes (i.e. healthy - faulty, normal - abnormal). Because of the specifics of the dataset utilized in this project classification model was determined as not a good option. Therefore, finally, a regression-based NBM approach was employed and a framework to realise it, was developed. This is further discussed in Chapter 4.

Additionally, most of the research done in the field of wind turbine prognosis utilizes run-to-failure data to perform RUL predictions. However, obtaining such a dataset in a real-world scenario would be

a costly endeavour. To make use of a dataset without run-to-failure data, a new method for predicting failure could be proposed which could also set automatic thresholds and ring an alarm to perform maintenance of the WT component.

Lastly, in the literature reviewed the potential of using high-frequency SCADA data for condition monitoring has not been fully explored. In the current standard industry practice, SCADA data aggregated over 10min intervals is used. Such aggregation is done by averaging which results in loss of information about the condition of the wind turbine component, limiting the suitability of this data system for WT CM purposes. To this end, an investigation on the effect of averaging for the aim of condition monitoring, a comprehensive sensitivity analysis with varying SCADA sampling frequencies is proposed necessary.

3

Theoretical background

To understand and implement the framework proposed in this research, comprehension of machine learning algorithms and related data-driven techniques is necessary. This chapter aims to provide a theoretical background for such understanding and explains related terminology and concepts. One of the major parts of the proposed framework is an artificial neural network (ANN). Architecture, training and implementation principles of ANN are explained in Section 3.1. To build an ANN model suitable dataset is necessary - it should adequately represent different operational regimes. However, real-world data is usually not ideal, and to that end, pre-processing is necessary to balance the dataset. The issue of imbalanced domains together with its solution approaches are introduced and explained in Section 3.2. Another important part of the framework is a support vector machine. Its purpose and cognition, when concerned with classification problems, is given in Section 3.3. Lastly, this research proposes an online WT monitoring scheme that utilises linear regression and bootstrapping. These concepts are explained in Section 3.4.

3.1 Artificial neural networks (ANN)

Artificial neural networks (ANN) are computational systems inspired by biological neural networks; ANN is designed to simulate the way human brain analyse and process signals. When used in combination with powerful computational hardware, ANN can solve complex problems that would be unsolvable otherwise. ANN being a simulation of human brain activity, has a similar architecture to the human neural system, where multiple neurons are interconnected with each other into a neural network. These neurons are activated based on the information they receive from other neurons they are connected to [59]. An ANN is a *black-box* approach as there is barely any control of the model except by the external hyper-parameters. The basics and relevant working of an ANN are discussed in this section. Practical application of these concepts is presented in Section 5.2 and results of ANN implementation are shown in Section 6.1.

Single perceptron

The simplest form of an ANN is known as the *perceptron*, consisting of a single node or neuron. The computation process within the neurons consists of addition and multiplication procedures. Figure 3.1 shows an overview of how this perceptron works. The perceptron considers a certain number of inputs p_i from R input nodes. Each of these nodes is connected to the perceptron and each connection has a weight $w_{1,i}$. At the node (summation box in Figure 3.1), a linear addition n of all input nodes and their weighted connections is performed, using Equation 3.1.

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \dots + w_{1,R}p_R = \sum_{i=1}^R w_{1,i} * p_i \quad (3.1)$$

The other half of the node transforms this linear algebra operation into a non-linear system of equations by means of activation function f , which is defined by the user. This function processes the inputs and provides for an output. An example of a sigmoid logistic function is provided in Equation 3.2.

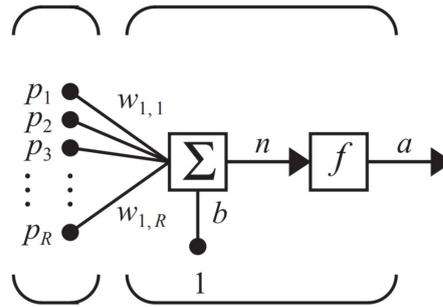


Figure 3.1: Calculation process within a neuron (adapted from [60])

$$f(n) = \begin{cases} 0, & \text{if } n < 0 \\ \frac{1}{1 + e^{-n}}, & \text{if } n \geq 0 \end{cases} \quad (3.2)$$

Additionally, the node itself has a bias (b), which provides additional weight to the node's output. Consequently, the node with a higher bias has a higher contribution to the final output. It allows for shifting the activation function and directly influences $f(n)$. Finally, the output of the activation function is passed as the output a of the node; Equation 3.3

$$a = f\left(\sum_{i=1}^R w_{1,i} * p_i + b\right) \quad (3.3)$$

ANN architecture

The above description is for a single perceptron but this yields in a poor model. To increase the flexibility and complexity of ANN model several perceptrons and layers can be used, which yields in a multi-layer perceptron (MLP). This is a widely-used architecture, where each node in a given layer is connected to every node in the previous and next layer, if present. Similar to the single perceptron, each connection has a weight w_i . Figure 3.2 gives an overview of this architecture. The network typically consists of three different types of layers - input layer, hidden layers and output layer.

The input layer carries the data stream input features. Much like in a biological neuron system, the information propagated through the neurons in the hidden layer, which send and receive a stream of information from the previous layers to the output layer. Such an architecture is known as a *feed-forward* neural network; the information is only fed forward in the network. More complex architectures are also possible and may include, skipping connection layers, cycles within the network itself or input fed in a layer. Nevertheless, these are difficult to design and are not considered for this thesis; only feed-forward neural networks are considered.

Input layer

The input layer is the first layer in the ANN architecture which gets raw input parameters as input. Each node carries an input parameter and is connected to each node in the next layer with connection weights. This layer does not have any activation function or bias.

Output layer

Another base layer in the ANN architecture is the output layer placed at the end of the network. The output of this layer is read as the model output and further used to evaluate the model's performance (using performance metrics as described in Sub-section 2.4.2). The output layer provides results respective to the specified task. For instance, for supervised learning the output layer outputs the classification label or value from the regression problem. In the case of a classification problem,

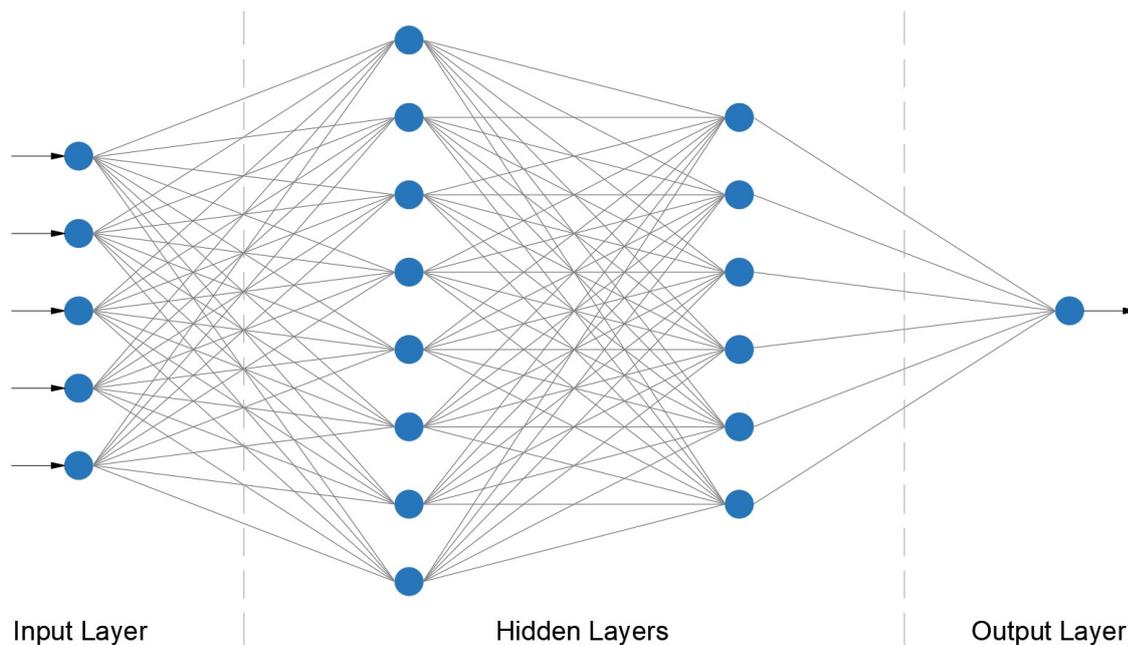


Figure 3.2: The architecture of an artificial neural network (ANN)

the output layer is made of the same number of neurons as the necessary classes to predict. On the other hand, for a regression problem, the output layer consists of only one neuron, containing the target variable to be predicted when provided with some input variables.

Hidden layers

Between the input and output layers, there can be an almost unlimited number of hidden layers, consisting of a pre-defined number of neurons. As the name suggests, these layers are hidden and do not interact directly with the user, except through setting the hyper-parameters. A high number of nodes and layers is not always advantageous as it increases the required computational power and does not necessarily result in better model performance. Furthermore, it also increases the number of unknowns that need to be solved by the model during training. The number of hidden layers is a crucial parameter for the architecture of ANN. For regression problems, it has been found that it is better to use one hidden layer and work on changing the number of neurons and/or training data set until the best performance is achieved [61]. The reasons for not using more than one hidden layer are as follows [62]:

1. Due to the addition of more hidden layers, the network performance becomes unstable (due to more unknowns) and is subjected to more noise as there are more neurons and connections between the layers.
2. The fitting of the model becomes more complex and very specialised to predict the training cases, therefore compromising the generalization capabilities of the model due to overfitting to the training data.
3. There is more potential to reach a local optimization solution when the network has two or more hidden layers because more neurons are optimized at different layers. Consequently, this might lead to increased model prediction errors.
4. In the case of employing certain activation functions such as sigmoid, tanh for the hidden layers, as more layers are added, the problem of 'vanishing gradient' becomes relevant. This is because some activation functions compress a large input space into a small input space, which in the case of sigmoid function would be 0 and 1. As the inputs of the activation function become larger, the gradient becomes too small for training to be done effectively.

Another important parameter for the ANN architecture is the number of neurons within the hidden layer. If the chosen number of neurons in the hidden layer is too large, it could lead to overfitting, meaning that the model fit will be very good for the data from which it learned, but it will not be able when presented with new data to perform as well. This means that the generalisation capability of the model would be low [60]. To obtain the optimal number of neurons in the hidden layer, an iterative process is executed. As a starting point, as a rule of thumb, the number of hidden neurons (N_h) can be calculated by equation 3.4, where α is an arbitrary scaling factor ranging (usually) from 2-10 and indicates how generalized we want the model to be [60].

$$N_h = \frac{N_s}{\alpha * (N_i + N_o)} \quad (3.4)$$

where N_s is the number of samples in the training data set, N_i is the number of input neurons, and N_o is the number of output neurons.

Activation function

An activation function in a neural network defines how the weighted sum of the inputs (net input n) is transformed into an output from a neuron in a network layer. The purpose of the activation function is to introduce non-linearity in the ANN model, enabling it to capture complex patterns. If a linear activation function is selected, the ANN model will imitate a linear operation, equating to a linear regression problem.

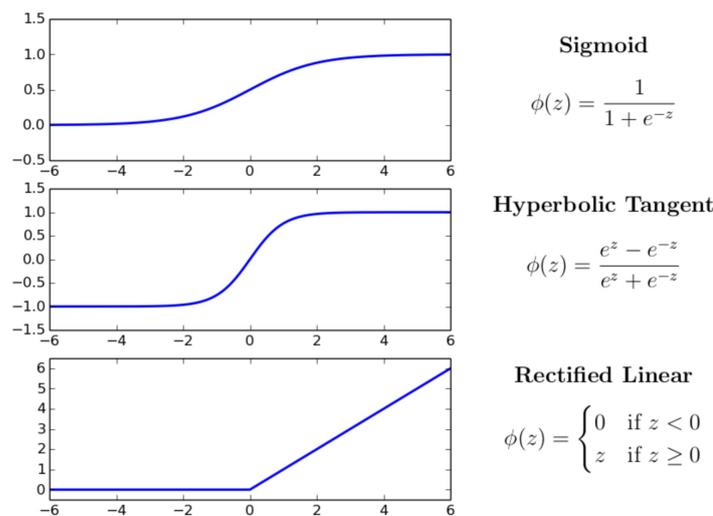


Figure 3.3: Common activation functions used in ANN models (ϕ - activation function, z - net input) [63]

Non-linear functions are important to discover relevant information in the layers and are chosen based on the problem the neuron is solving. Some of the common activation functions are shown in Figure 3.3. The sigmoid function transforms the input domain $(-\infty, \infty)$ into the range domain $(0, 1)$, whereas the *tanh* function maps the same input domain onto a slightly different domain $(-1, 1)$. All hidden layers typically use the same activation function, while the output layer usually uses a different activation function depending on the prediction required by the model [64]. For example, the hidden layer may use a logistic (sigmoid) activation function to understand non-linear relations between the inputs and output for a regression problem. However, for the output layer, a typical choice of activation function would be an identity ($f(n) = n$) function.

Optimization

Optimization of ANN refers to training the model using input data for which the target output feature is already known. Optimization of a neural network is done by updating the matrix of weights W_i , subjected to an objective function. The cost function that needs to be minimised is the mean square error (MSE) - the difference between the neural network predicted outputs and known target variable values. The algorithm used to minimize this objective function is known as back-propagation. Equation 3.5 demonstrates the optimization of a general ANN [62].

$$\begin{aligned} & \text{Optimize : } W_i \in R^N \\ & \text{Minimize : } \sum_{i=1}^N (T_i - Y_i)^2 \end{aligned} \quad (3.5)$$

where, N represents the number of hidden neurons, and W_i is the i^{th} weight value. T_i and Y_i are the target and predicted values, respectively. In essence, once the output is computed through the feed-forward pass algorithm, where the training input values are propagated to the output layer using randomly initialized weights, the inverse process starts. The weights are updated by comparing the target value from the training dataset with the prediction and then calculating their respective derivatives at every layer, starting from the output layer, going through all the hidden layers. Consecutively, ANN computes a new output based on the actualization of these weights, repeating the process until the cost function is minimized, i.e. the weight matrix is associated with the minimum loss or MSE. The backpropagation algorithm is a way of learning from the training data set, as progressive iterations tend to reduce the training error close to zero [61].

The optimization algorithm describing the feed-forward pass and backward propagation flow is summarized as follows [65]:

- Step 1 - Initialisation: random initialization of weight matrix.
- Step 2 - Feed-forward pass: propagate the input values through the ANN to the output layer; compute errors between the training values of the target variable and predicted values.
- Step 3 - Backward propagation: Compute the gradient of the optimization function; update weight matrix; repeat step 2 until optimization is achieved as per equation 3.5.

A detailed explanation for the optimization of neural networks along with mathematical derivations is presented in Appendix A.

Other various optimization algorithms have been proposed to be integrated with the backpropagation algorithm. Following the classical gradient descent technique, the stochastic gradient descent (SGD) technique randomly selects a subset of the training data set to calculate the partial derivatives. Compared to its other variants, such as batch gradient descent and mini-batch gradient descent techniques, SGD significantly reduces the computational time as fewer calculations are required. However, in cases where the steepness of surfaces between different axes is not equally distributed, SGD has trouble finding local optimal points [66]. In this regard, a new method named Momentum [67] was formulated to accelerate SGD in the relevant direction and damp oscillation by adding a fraction γ of the update vector of the past time step to the current update vector. Consequently, an adaptive subgradient (Adagrad) algorithm [68] was proposed, which adapts the learning rate to the parameters, performing larger updates for infrequent and smaller updates for frequent parameters. As an extension to Adagrad method, adaptive moment estimation (Adam) [69] was articulated. This method computes the adaptive learning rates for each parameter. In addition to storing an exponentially decaying average of past squared gradients, it also keeps an exponentially decaying average of past gradients.

In summary, building an efficient ANN for system prediction requires careful study of some issues related to the network, including (1) a universal function approximation capability (i.e. the ability to generalize the problem successfully); (2) resistance to noise or missing data (i.e. feature extraction

and noise filtration) and (3) accommodation of multiple non-linear variables for unknown interaction (i.e. selection of appropriate activation function).

3.2 Pre-processing approach for imbalanced regression

Imbalanced domains are a relevant problem when working with real-world data and refers to a dataset that is not able to properly represent a target variable under all relevant cases. The main problems pertinent to imbalanced domains arise due to the limited representation of events of concern in the data and non-uniformity in the bias of the user across the range of the target variable [70]. This issue has been studied mostly in the context of classification tasks [71] [72]. One of the approaches for handling imbalanced domains is intervening in the learning process of any predictive model. This could be done in the data pre-processing stage, which tends to change the original distribution of the data before the learning algorithm is applied, to change the target variable distribution to force the learning algorithm to also focus on rare and interesting cases. One of the reasons for applying a special-purpose algorithm in the pre-processing stage is because of its flexibility regarding the use of any learning algorithm later on. A case of imbalanced domain with regards to wind turbine operational data was encountered in this project and is further discusses in Section 5.1.4.

Several solutions have been proposed in the literature for dealing with imbalanced classification tasks. However, imbalanced domains also occur in other predictive contexts such as regression tasks, data streams or time series forecasting [73] [74]. Compared to classification tasks, the problem of imbalanced domains in regression tasks is more complex due to the continuous nature of the target variable. Also, the definition of the importance of values of the target variable is not straightforward. As mentioned before, a solution to the imbalanced domain problem is changing the data distribution: increasing the density when the case is relevant and decreasing when it is not. The relevance is circumstantial, and hence, it must be determined by the user providing a relevance function and threshold. This results in determining rare and relevant cases (D_R) and the set of normal and uninteresting cases (D_N).

There are three most commonly applied strategies for dealing with imbalanced domains: Random under-sampling [75], SMOTER [76] and introduction of Gaussian noise [77]. Random under-sampling is a straightforward strategy that, by randomly removing samples belonging to the normal and less interesting cases for the target variable, achieves a better balance between the interesting / rare and uninteresting / normal cases. SMOTER is an adaptation for regression of the well known SMOTE (synthetic minority oversampling technique) algorithm [78]. This algorithm applies a random under-sampling strategy to the normal cases and generates new synthetic examples from the rare cases through an interpolation strategy. This interpolation is carried out using two rare cases (one is a seed case and the other is randomly selected from the k-nearest neighbours of the seed). The new target variable value is determined as a weighted average of the target variable values of the two rare cases used. The introduction of Gaussian noise employs the same random under-sampling strategy for normal cases. However, it generates new synthetic examples for rare cases using the addition of normally distributed noise to the rare examples [79]. These three approaches have been designed for classification problems and adapted to regression tasks.

To overcome the issue of imbalanced domains in regression tasks, Branco *et al.* [70] proposed a new pre-processing approach named SMOGN (synthetic minority over-sampling technique with the introduction of Gaussian noise), which incorporates the two existing approaches for under-sampling and oversampling, trying to solve problems associated with both of them. It combines a random under-sampling strategy with two over-sampling techniques: SMOTER and introduction of Gaussian noise. SMOGN algorithm is a pre-processing approach for tackling imbalanced domains which acts before the learning process stage. The motivation behind SMOGN is described as follows [70]:

- To limit the risks incurred when using SMOTER as it would not use the most distant examples in the interpolation process.
- To increase the generalization capability by allowing expansion of the decision boundaries of rare

cases, which is more difficult to achieve with the introduction of Gaussian noise, as it is a more conservative approach.

Application and effects of the SMOGN algorithm are discussed and illustrated in Sub-section 5.1.4.

The SMOGN algorithm [70] begins by building data partitions containing consecutive examples considering the target variable, which are clustered into two types: $Bins_R$ - the rare and important partitions, and $Bins_N$ - the normal and less important partitions. This means the $Bins_R$ contains the higher relevance examples (examples with relevance above a pre-defined threshold). Random under-sampling strategy is applied to the examples in the partition belonging to $Bins_N$. On the other hand, the examples in $Bins_R$ are targeted with an oversampling procedure. For each case (the seed example) in a partition belonging to $Bins_R$, synthetic samples are generated using either SMOTER or the introduction of Gaussian noise. The main idea is that if a selected nearest neighbour is at a 'safe' distance, it is suitable to perform interpolation through the SMOTER strategy. However, if the selected neighbour is not at a safe distance, then it is better to generate a new synthetic sample by introducing Gaussian Noise on the seed case. The threshold used to decide the "safe" distance for a selected neighbour depends on the distance of the seed example from all the remaining cases in the partition under consideration. It is selected as half of the median of the distances between the seed example and other examples belonging to the same partition.

Figure 3.4 shows a synthetic sample with 5-nearest neighbours of a seed case, where some are within the safe distance range (3 blue points marked in the dark grey area) and others at an unsafe distance (2 blue points marked in the light grey area). The examples belonging to the relevant bin are marked with blue bullets and the examples marked with crosses belong to a normal bin. It is also worth mentioning that the examples belonging to the normal bin (Bin_N) are more likely to overlap with the examples from the relevant bin (Bin_R) at an unsafe distance.

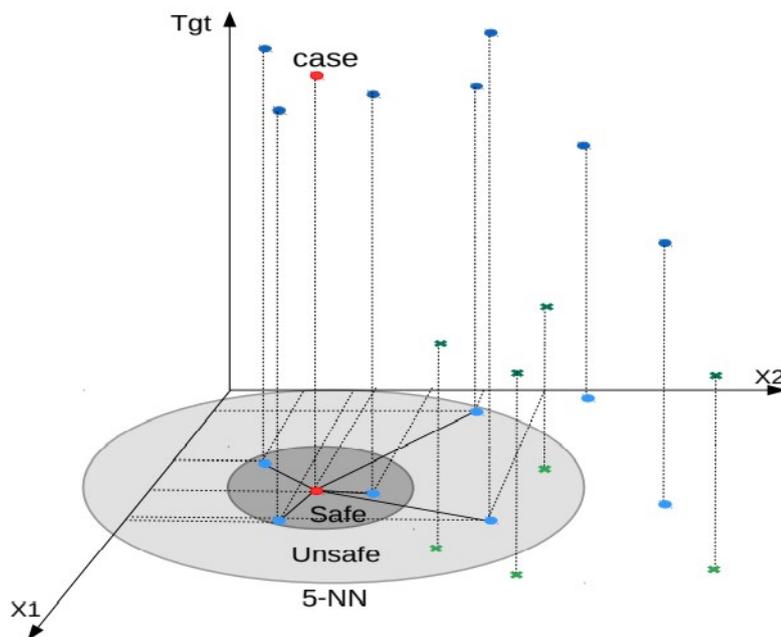


Figure 3.4: Synthetic sample generation using the SMOGN algorithm [70]

3.3 Support vector machines (SVM)

Support vector machine (SVM) [80] has been extensively used for both classification and regression purposes. SVM is characterized as a binary classification algorithm, which differentiates the data set between two classes by finding an optimal hyperplane that separates them. The optimal hyperplane

is constructed to maximise the margin between the positive and negative training examples as shown in Figure 3.5. Through a set of *support vectors* (the subset of the training data which is nearest to the separating hyperplane) the max-margin hyperplane is determined. Multiple SVM classifiers can be built, each of which is capable of classifying between different classes (*multi-class*), or by classifying between pairs of classes (*one-vs-one*), also known as one-class support vector machine (OC-SVM). In this regards, SVMs also provide the flexibility of differentiating two classes based on multiple features and hence, can capture complex separable boundaries between the two classes. One-class classification is quite distinct from the multi-class classification setting, where the classifier is only required to differentiate between a target class and one specific non-target class (opposing class) and not all non-target classes. With regards to anomaly detection setting, a one-class classifier distinguishes between the normal data (target class) and any sort of anomaly (non-target class). This behaviour of one-class classifiers makes them an ideal choice for data-driven anomaly detection problems.

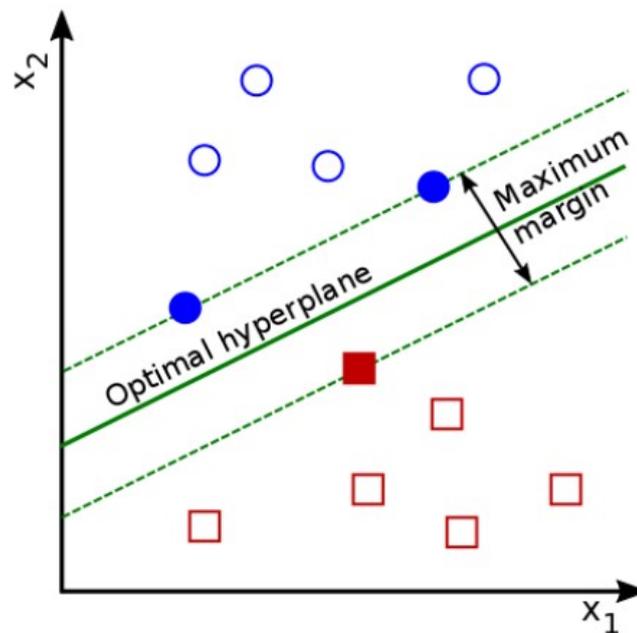


Figure 3.5: Support vectors and max-margin hyperplane [63]

One class support vector machines (OC-SVM) is a variant of SVM, which, as the name suggests is a one-class classifier and classifies between two classes by producing a hyperplane around the training data, which is then used to decide whether the future data is similar to this class, or an anomaly [81]. This module is especially advantageous in settings where there are a lot of available 'normal' data and not many cases of anomalous instances. To this end, the Anomaly Detection Module in this project was developed based on OC-SVM model which was trained using a dataset comprising of all or mostly normal operations. The model recognises the boundary of normal behaviour and then aids in deciding if the future data also belongs to the normal class. The implementation of OC-SVM is discussed in Section 5.3 and its results are presented in Section 6.3.

Like SVM, OC-SVM uses support vectors to decide on the boundary, which are just data points closer to the boundary. In the conventional SVM, this would be the boundary between two or more classes. Being a boundary-based method, the model is sensitive to outliers in the training data that can degrade the model performance [82]. In order to offset this disadvantage, OC-SVM uses ν as a hyperparameter to define what portion of the data should be classified as outliers. The parameter ν is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors relative to the total number of training samples. For instance, a value of ν set to 0.01 would find at most 1% of the training samples being misclassified at the cost of a small margin, while would be beneficial in preventing any misclassified anomalies in the later stage.

The training of OC-SVM involves an intensive computational complexity as it aims to solve a quadratic programming problem. An OC-SVM solution is obtained by estimating a probability distribution function which makes most of the training data more likely than the others, and a decision rule that separates these by the maximum possible margin. Let $\mathcal{Z} = z_1, z_2, \dots, z_n$ be the n -dimensional training data. Let $\Phi : \mathcal{Z} \rightarrow \mathcal{G}$ be a kernel map which transforms the training data to a higher dimensional space. OC-SVM separates the data by trying to solve the following quadratic programming problem [83]:

$$\min_{w \in \mathcal{G}, \epsilon_i \in \mathbb{R}} \left\{ \frac{1}{2} \|w\|^2 + \frac{1}{\nu N} \sum_{i=1}^N \epsilon_i - b \right\} \quad (3.6)$$

subject to

$$\nu \in (0, 1], \epsilon_i \geq 0, \forall i = 1, \dots, N, \quad (3.7)$$

$$(w * \Phi(z_i)) \geq b - \epsilon_i, \forall i = 1, \dots, N, \quad (3.8)$$

where w is the weights of the function, ν is the ν hyperparameter, ϵ are the non-zero slack variables in the margin to produce a soft margin that would help prevent overfitting, b is an offset parameterising a hyperplane in the feature space associated with the kernel, z_i are the input values and $\Phi(z_i)$ maps these inputs from input space to feature space allowing for a hyperplane splitting the data classes to be formed.

Using Lagrange techniques and a kernel function $K(z, z_i) = \Phi(z)^T \Phi(z_i)$, for the dot-product calculations, the decision function $f(z)$ becomes:

$$f(z) = \text{sign}\{(w * \Phi(z)) - b\} = \text{sign}\left\{ \sum_{i=1}^N \alpha_i K(z, z_i) - b \right\} \quad (3.9)$$

Through this method, we thus create a hyperplane characterized by w and b which has maximal distance from the origin in the feature space \mathcal{G} and separates data points. Here, α_i are the Lagrange multipliers; every α_i is weighted in the decision function and therefore "supports" the machine. In Equation 3.6, non-zero ϵ_i are penalised in such a way that the decision function as shown in equation 3.9 should be positive for most input examples, i.e., they will be considered normal, while the regularization term ($\|w\|$), shown in Equation 3.6 should remain small. This is determined by the variable ν , which is the contamination hyperparameter. The model should then provide a predicted output either being positive 1 for data in the centre, representing clustered observed data and negative 1 for data lying outside the boundary, meaning abnormal or anomaly. Figure 3.6 shows OC-SVM facilitating a boundary around the observed data during training, and is able to identify anomalies, i.e. the data points away from the learned decision boundary.

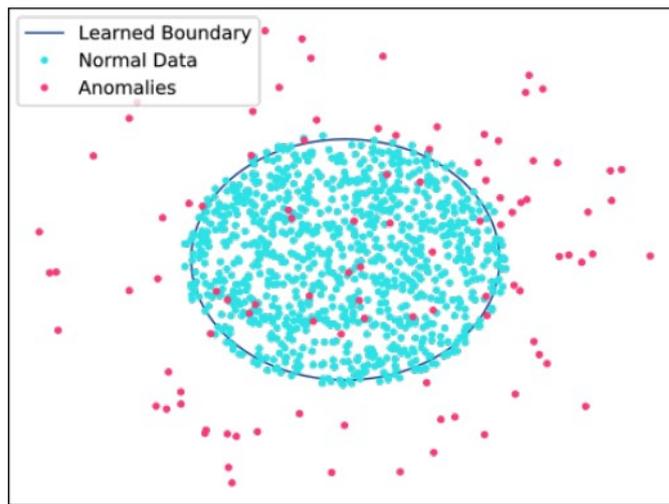


Figure 3.6: OC-SVM learned boundary from training data and anomaly identification [84]

3.4 Linear Regression and Bootstrapped Confidence Intervals

Regression analysis can be defined as the collection of statistical tools that are used to model and explore relationships between variables in a non-deterministic manner. Of all the tools available, linear regression is a widely implemented technique to model the association between a dependent variable and one or more independent variables [85]. In the simplest case, which is what was explored during this project, there is one independent (response variable), usually represented by y and a dependent variable (predictor variable) x . Such a relation can be explained with a simple linear regression model of the form $Y = \beta_1 x + \beta_0$, with a slope β_1 and an intercept term β_0 . The most straightforward and common approach to finding the optimal parameters (β_1 and β_0) is through the setting of ordinary least squares (OLS). OLS is a statistical method that ensures that the difference between the actual observed data and the fitted linear regression model is as small as possible [86].

Given a set of n pairs of observations $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ with a linear relationship between them, the n samples in observation can be described by a simple regression model and can be represented by equation 3.10 [85].

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, 2, \dots, n \quad (3.10)$$

where, the data x_i is the value of the predictor variable and y_i is the response variable in the i^{th} trial, β coefficients are the unknown parameters of the linear fit and ϵ_i is the random error term with mean 0 and variance σ^2 . The simple linear regression model relies on a number of assumptions being satisfied for it to provide a reliable approximation to a linear association between two variables. The assumptions are related to the probability distributions of the random error term (ϵ) in the model. The four main assumptions about these random errors are [87]:

1. The distribution of ϵ at each x -value has a **mean of zero**: $E\{\epsilon_i\} = 0$. In other words, the data points are scattered along both sides of the regression fit in a balanced way such that the random errors average is 0 as we move across the plot from left to right.
2. The distribution of ϵ at each x -value has a **constant variance**, also known as *homoscedasticity*: $var\{\epsilon_i\} = \sigma^2$. This means that the data points scatter evenly around the regression fit such that the variation of the random errors is constant as we progress through the plot from left to right.
3. The distribution of ϵ at each x -value is **normal**. That is, the data points are more likely to be scattered closer to the regression fit than farther away and have a gradually decreasing likelihood of being distant from the regression fit.
4. The value of ϵ for a certain observation is **independent** of the value of error for any other observation.

OLS finds the estimates of β_1 and β_0 that best fit the data by minimizing the sum of the squared distances (errors) from the fit of each response [88]. The sum of the square of error of the observations from the linear regression line is given by equation 3.11 [85].

$$L = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_i)^2 \quad (3.11)$$

The OLS estimators of β_0 and β_1 , symbolized by $\hat{\beta}_0$ and $\hat{\beta}_1$, must satisfy conditions in equations 3.12 and 3.13 [85].

$$\left. \frac{\partial L}{\partial \beta_0} \right|_{\hat{\beta}_0, \hat{\beta}_1} = -2 \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) = 0 \quad (3.12)$$

$$\left. \frac{\partial L}{\partial \beta_1} \right|_{\hat{\beta}_0, \hat{\beta}_1} = -2 \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) x_i = 0 \quad (3.13)$$

The OLS estimates of the intercept $\hat{\beta}_0$ and the slope $\hat{\beta}_1$ in the simple regression model are the solution to the above equation and are presented in the equations 3.14 and 3.15 [85].

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \quad (3.14)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n y_i x_i - \frac{(\sum_{i=1}^n y_i)(\sum_{i=1}^n x_i)}{n}}{\sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n}} \quad (3.15)$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$ and $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$.

In statistics, a confidence interval is a measure of how accurate, or how good, an estimate of a certain parameter is. Confidence interval comprises a range of acceptable values that are defined as good estimates of the unknown population parameter. The necessity of confidence intervals stems from the fact that a point estimate, being a single value, is insufficient to express the statistical variation or random error, that the estimate might have [89]. The confidence interval can be defined as a range of values constructed from the sample data such that the population parameter is likely to occur within that range at a predefined probability. This predefined probability is known as the level of confidence. Standard errors are often used to estimate approximate confidence intervals for parameter θ of interest. For a given estimate $\hat{\theta}$ and an estimated standard error se , the $100(1 - \alpha)\%$ confidence interval for θ is given by the equation 3.16. In the case of a normal distribution, 95% confidence intervals can be obtained using 1.96 as the critical value [90].

$$\hat{\theta} \pm \text{criticalvalue} * se \quad (3.16)$$

This equation can also be used in reference to regression problems following the four assumptions related to the residuals as mentioned above. However, this means that its application is limited as if these assumptions are violated the estimation of the regression coefficients obtained by OLS may not be appropriate [91]. In such a case, bootstrapping is an advantageous method. The bootstrap is an important non-parametric approach to statistical inference that can provide valid standard errors, confidence intervals and hypothesis tests without making the assumption of a normal distribution of the parameters. It is a data-based simulation method for statistical inference, which involves repeatedly drawing random samples from the original data, with replacement. 'With replacement' means that any observation can be sampled more than once. Although for most problems, it is hard to discern the true confidence interval, bootstrapping method is asymptotically more accurate than the standard intervals obtained using sample variance and assumptions of normality [92]. Another advantage of this method is that it avoids the cost of repeating a certain experiment to get more groups of sample data [89]. This is a very important benefit when analysing failure data for wind turbines as the real-world data is often skewed with a significantly low number of data samples representing the faulty state of the turbine. The bootstrap procedure can be implemented to get more accurate statistical inferences in the following situations [93]:

- When the theoretical distribution of a parameter of interest is complex or unknown. Since the bootstrapping procedure is distribution-independent, it provides an effective method to understand the underlying distribution of the sample and parameters of interest derived from it.
- When the sample size is insufficient to draw a straightforward statistical inference. Even when the underlying distribution is well known, bootstrapping provides a good way to account for the distortions caused by the specific sample.

In this project, owing to the unknown distribution of the sample data and insufficient samples in the studies data set, the bootstrap method was used to estimate the confidence intervals around the linear regression fit and its implementation is further discussed in Section 6.4. The main idea behind the bootstrapping regression is to construct bootstrap standard errors and confidence intervals for regression coefficients. The procedure to do so for a simple regression model is discussed here, specifically called bootstrapping the residuals. The topic of bootstrapping regression is discussed in detail in the book by Efron and Tibshirani [89].

1. Estimate the regression coefficients $(\hat{\beta}_0, \hat{\beta}_1)$ for the original sample and calculate the fitted value (\hat{y}_i) and residual for each observation. $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ and $\epsilon_i = y_i - \hat{y}_i$ for $i = 1, 2, 3, \dots, n$
2. Keeping these residuals as the original sample, generate M bootstrap samples and calculate $y_{m,i}^*$ values for each observation in the bootstrap sample. $(y_{m,i}^* = \hat{y}_i + \epsilon_{m,i})$ where, $m = 1, 2, 3, \dots, M$
3. Regress the bootstrapped $y_{m,i}^*$ values on the fixed x values to compute the estimated bootstrap coefficients. Estimates are calculated using the OLS method. These estimators $\{(\hat{\beta}_0^*, \hat{\beta}_1^*)_1, (\hat{\beta}_0^*, \hat{\beta}_1^*)_2, \dots, (\hat{\beta}_0^*, \hat{\beta}_1^*)_m\}$ can then be used to construct the bootstrap standard error and confidence intervals for the regression coefficients. The estimated coefficients can be ordered in an increasing manner, and a simple 95% bootstrap confidence interval would be from the $0.025M^{th}$ to $0.975M^{th}$ largest values.

4

Methodology

One of the goals of this research project is to develop and implement a framework for prognostics of wind turbine gearbox. Based on the literature reviewed and presented in Chapter 2 as well as the theoretical background explained in Chapter 3, a methodological approach was established and prognostics model constructed. The methodology of this research is explained in this chapter. Section 4.1 describes the framework on which this research and its results are based upon. This framework can be segmented into three different modules, each of them is explained in more detail in the sub-sections. Firstly, as real-world datasets are not ideal, the need for data pre-processing and how it is carried out are discussed. After that, the principles behind the normal behaviour model are described. Lastly, the process of detecting anomalies in the turbine's operation is given together with the explanation of a real-time monitoring scheme that sets robust thresholds for activating warnings and alarms for maintenance. Another aim for this research is to conduct a sensitivity study to investigate the effects of different data sampling periods and determine an optimal sampling period for SCADA data. The way the sensitivity study was conducted is explained in Section 6.5.

4.1 Framework for gearbox failure prognostics

The aim of this thesis is to develop a framework for wind turbine gearbox prognostics using SCADA data. As mentioned before in Section 2.3, owing to the extensive information stored and applicability of SCADA systems, SCADA data has attracted great research interest to perform condition monitoring of wind turbines. Various signal analysis methods and model-based approaches have been developed to analyze trends of typical WT SCADA parameters. Such analyses have shown that significant changes and anomalies in the WT behaviour can be detected at an early stage [30]. One such approach has been adapted for this framework, the idea behind which is as follows:

- Normal behaviour of the WT gearbox is emulated to predict the behaviour in the case when no fault progression is happening.
- The predicted signal representing normal behaviour is compared to the real measurement taken in the machine.
- If the deviation of the measured signal from the modelled normal behaviour signal is observed an indication of fault initiation and progression is detected and quantified.
- The quantified deviation is then tracked through time and a maintenance alarm is given when the deviation starts increasing substantially indicating that the fault is bound to happen.

To realize the idea explained, a regression-based data-driven approach is implemented using a *normal behaviour model (NBM)*. This can be identified as the foundation of the developed framework.

The established framework is shown in Figure 4.1. The entire framework can be segmented into three modules, namely, data pre-processing, normal behaviour model and anomaly detection and prognosis. The data pre-processing module entails operations necessary to treat data that is used for building and employing the NBM, particularly, data cleaning and feature selection and engineering. The NBM module is an artificial neural network (ANN) model, which is trained to learn the mapping between selected input and target features and is able to imitate the healthy state operation of the turbine. In the Anomaly detection and prognosis module, the NBM predictions are compared with the

field measurements to track anomalies. Here, OC-SVM is employed to analyse the error and set a complex, continuous boundary to distinguish between periods of healthy and anomalous operation of the turbine. Furthermore, to understand the inception of failure in the gearbox, a real-time monitoring scheme is developed which sets off an alarm for maintenance when the fault progression exceeds a certain threshold.

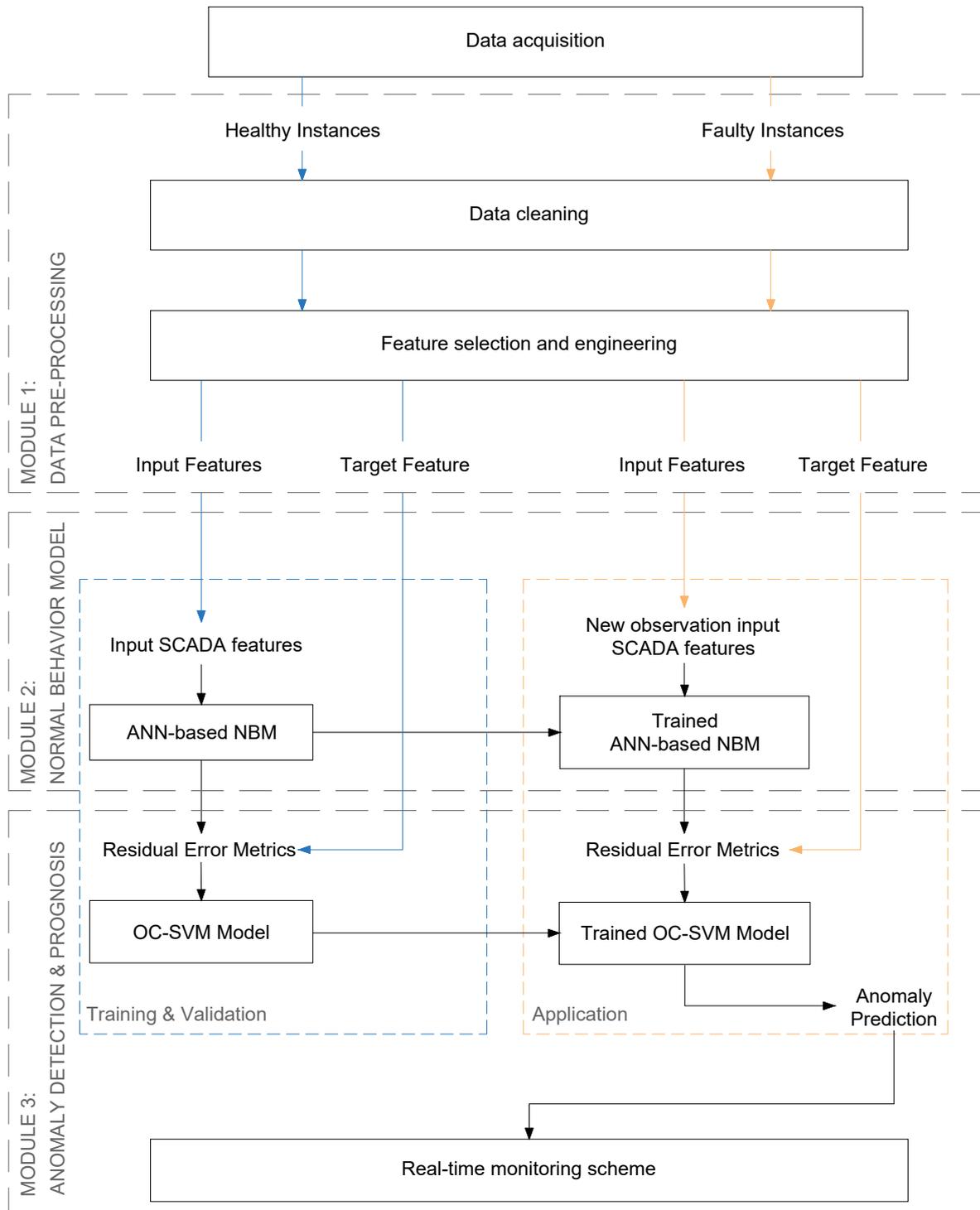


Figure 4.1: Model architecture for gearbox failure prognostics

The framework is executed in two phases, the first being *Training & Validation*, which encompasses training, testing and validating the employed machine learning models i.e. ANN and SVM. This is done using the healthy instances which are represented in blue. Once the models are trained and validated, they are used to detect anomalies and this characterizes the second phase, *Application*. This phase is executed using faulty instances from the wind turbine operational data and is shown in orange. The *Training & Validation* phase is a one-time process, during which the normal behaviour model is trained to emulate normal operating conditions and the extracted residual features are then used to train the classification model. The *Application* phase represents the continuous application process of anomaly detection and condition monitoring.

Additionally, another goal of the thesis is to determine the optimal SCADA data sampling frequency which can be utilised for wind turbine condition monitoring purposes. Once the framework is developed, a sensitivity study to investigate the effects of different sampling periods is carried out. This was done by analysing the model performance and amount of anomalies detected while varying the sampling period of the data and is discussed in Section 6.5.

4.1.1 Module 1: Data pre-processing

The framework, presented in Figure 4.1, has been developed to employ a data-driven approach for WT gearbox prognostics using SCADA data. In most modern WTs, SCADA systems record over 200 variables in intervals of 10 min, generating rich historical data. By using appropriate data treatment solutions, the recorded signals could be converted into useful information for condition monitoring. The SCADA dataset utilised in this project contains one failure mode of the gearbox which occurred due to gear teeth misalignment in the planetary stage and resulted in a turbine shutdown and maintenance [15]. A detailed overview of the data is provided in Sub-section 5.1.1.

As explained before, the developed framework is based on the NBM approach which emulates the healthy behaviour of the WT component. NBM is built using the healthy data representing the normal operation. Hence, one of the primary tasks after acquiring the data is to identify healthy and faulty periods of wind turbine operation. This is done by first identifying the date of failure and selecting an optimum time frame in which the first signs of incipient failure could be apparent, referred to as 'Faulty data'. On the other hand, 'Healthy data' comprises data depicting no signs of anomalous or faulty turbine behaviour and usually belongs to the early operational period of the machine component. One of the considerations while distinguishing these two data sets is to provide a good balance in terms of volume of data for the specific purposes healthy and faulty data have: training and validation of NBM and appropriate time span to analyse the progression of failure respectively.

Usually, the raw SCADA data acquired contains erroneous or abnormal data points - further referred to as 'outliers'. These might be caused by malfunctions in the SCADA communication system, sensor or signal processing error. In the context of SCADA signals, outliers can be identified as measurement data far away from a much larger and densely clustered set of normal measured data. Data cleaning is the process of correctly identifying and processing such outliers as they could have a strong negative impact while training the model which might even lead to false alarms. Some of the most commonly used filtration techniques are quantile filter, the Hampel identifier and the extreme studentized deviate test. The effects of removing outliers for wind turbine fault diagnostics and condition monitoring has been examined by Marti-Puig et al [94]. They identified that the best filtering strategy is to define operation ranges for each variable by a human expert or the manufacturer and filter these ranges separately. This is because systematic filtering of outliers might result in the removal of failure states of the wind turbine which are misidentified as bad / abnormal data. In this case, the data vectors corresponding to periods of the turbine's emergency shutdowns and anomalous power output were filtered out. Additionally, any abnormal signal values recorded, i.e. values above the threshold defined by the manufacturer, were removed. The filtration strategy employed within the framework of this project and all filtration steps are discussed in detail in Sub-section 5.1.4.

Once the data has been cleaned, the next step is feature selection and engineering. Feature selection is the process of selecting variables, herein time-series signals, that relate to the outcome that is

wished to study, understand and predict. Firstly, a target feature for the NBM should be identified such that it is sensitive to the analyzed failure mode of the gearbox and its deviation from predicted values carries information of fault inception. Various data mining approaches can be performed for such a selection; for example, Kusiak and Verma [42] proposed three data mining algorithms to establish a cause and effect relationship between different measurements available in the wind turbine SCADA system. However, such an approach may lead to a selection of a large number of input features and domain knowledge has to be applied to keep the number of parameters to a reasonable value as suggested by Schlechtingen *et al.* [95]. Furthermore, the selection of input features based on domain knowledge has been demonstrated successfully in other reviewed literature [41] [37] [58]. Hence, in this research, an understanding of the physics and the domain knowledge was considered to be the best method to decide suitable input parameters for modelling. The gearbox lubrication oil temperature was selected as the target feature as it is the most sensitive to the condition of the gearbox. Additional signals - wind speed, power output, high-speed shaft rotational speed, nacelle and ambient temperature were selected as the input features. This is also demonstrated in Table 2.2.

Furthermore, a commonly occurring issue when working with real-world datasets is the imbalance in data distribution across different operational regimes of the machine. Such imbalance domains within the data result in over-fitting for operational domains with large volumes of data and under-fitting for domains with relatively less data. This then means that the ML model learns more about some operational regimes than others, which deters its generalization capabilities. With regards to wind turbine, this problem is exacerbated by the fluctuations in the wind speed and consequently, the imbalance in its operational domain. In order to tackle this issue, a pre-processing technique, synthetic minority over-sampling technique with introduction of Gaussian noise (SMOEN), was implemented within the framework to reduce the difference between data volumes in different regimes. An extensive explanation of the working principles of this technique is given in Section 3.2 and its application is discussed in Section 5.1.4.

4.1.2 Module 2: Normal behaviour model

After the procedures within module 1 of the framework are carried out, the NBM can be built. The NBM works by empirically modelling the target feature/variable based on the selected input features. The process is summarized in Figure 4.2, where $u(t)$ are the input variables at time step t , $\hat{G}(t)$ represents the data-driven NBM to predict target variable $\hat{y}(t)$ while $G(t)$ constitutes the process of obtaining the measured target variable $y(t)$ through the required in-field sensor. Finally, $e(t)$ represents the residual error between the predicted and the measured value.

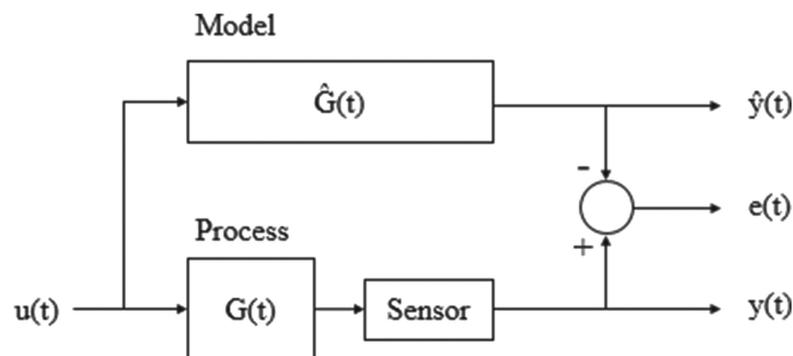


Figure 4.2: Normal behaviour model-based monitoring [96]

In this framework, the normal behaviour model (indicated as *Model* in Figure 4.2) is realised by the use of an ANN. The first and foremost task is to design an optimal architecture of the ANN which can be done experimentally or through algorithms. This task is concerned with configuring various parameters of the ANN such as the number of hidden layers and neurons, optimization algorithm, learning rate etc. For a given specific problem to be solved and the data in hand, an optimal ANN architecture results in reduced complexity and computational power. Furthermore, with regards to model performance, a simple architecture could lead to over-fitting issues, while complex architecture - to the under-fitting

problem; in both cases causing poor generalization capabilities and trap in local solution [97]. In this project, the model configuration was accomplished through experimentation utilizing knowledge about the data and the problem to be solved.

The normal behaviour model is developed in periods where the turbine components can be considered healthy (normal operations). Out of the samples in the healthy data set, 70% are used for training, 15% for testing with the remaining 15% for validation. As discussed in Sub-section 2.4.2, various metrics exist to verify the validity of the ANN model such as mean absolute error (MAE), root mean squared error (RMSE) etc., and they were computed for all modelling phases to evaluate the model's generalizing capabilities. This trained model is then, used to predict the selected target feature, where the prediction error gives an indication of changes in signal behaviour and thus incipient faults [37].

4.1.3 Module 3: Anomaly detection

Once the ANN-based NBM model is trained and validated, the next stage of the process is to evaluate the error between the model predictions and the actual measurements and determine a threshold that can distinguish between normal and anomalous behaviour of the wind turbine. While there are different error metrics proposed in the literature, such as root mean squared error (RMSE), health degree based on probability or Mahalanobis distance [57], there is no definite solution on how to best measure the NBM error. For instance, in case of modelling gearbox lubrication oil temperature, simple thresholds can be set based on the training RMSE to determine anomaly rates up to failure [41]. However, the interpretation of these results is limited, since only a single statistical metric fails to capture all the information about fault that can be gathered and, in some cases, can even lead to false or missed alarms. In this regard, Turnbull *et al.* [58] proposed a robust method to track and analyze the error by using a combination of different residual error features and detect anomalies through support vector machines. A similar approach was implemented in this project.

In this work, four residual error (difference between actual and predicted signal values) features - the root mean squared error, the minimum error, the maximum error and the error standard deviation - are computed for a selected time period. These statistical features were selected to best understand the distribution of errors and capture maximum information about them. These residual features from healthy data can then be used as input to train the OC-SVM model and recognize 1% of data as anomalies in the training period; therefore, a similar percentile would be expected if no fault is present in the system. This percentage is obtained through data analysis and is an iterative process. It is important to define such a percentage of anticipated outliers in the data to develop the min-max decision boundary for OC-SVM and prevent it from misclassifying any anomalous operation as normal. If the percentage of anomalies exceeds the predefined value of 1%, this would be an indication of incipient gearbox fault in the WT. The SVM is able to learn the complex, continuous boundary formed by these features and sets it as a threshold for a data point to be recognised as an anomaly. Once both the models - ANN-based NBM and OC-SVM - are trained, they can then be used to detect anomalies in the dataset representing a faulty condition of the turbine. This is done by first using the NBM to predict the target feature based on the same input features as the ones used in the training phase and then assessing the error between predicted and measured values. Finally, the calculated error metrics are fed into the SVM classifier to identify if the new data point represents normal or anomalous behaviour.

After identifying the anomalies in the faulty dataset, for each week of turbine operation before the failure, the percentage of anomalies observed is recorded. The percentage of anomalies is the ratio of total anomalous data points and the total number of data points logged in that week. These anomalies are then tracked and analysed to determine an alarm threshold, which would indicate an alert for maintenance and provide enough lead time to plan required activities. The alarm threshold can be set in several ways. One approach is to set confidence interval bounds based on the observed distribution of the percentage of anomalies. However, the applicability of such a static threshold would be inadequate as it could be limited to a specific data set and, if implemented, could result in false alarms. To tackle this issue, a real-time monitoring scheme based on setting an adaptive threshold is developed and integrated with the framework.

As stated above, the cause of the gearbox failure with the turbine in this study was gear-teeth misalignment. Misalignment faults in the gearbox can occur due to various reasons such as abrasive wear, surface fatigue caused due to moisture ingress in the lubrication oil or gradual wear by debris in the form of ferrous and non-ferrous particles accumulated in the oil [98]. Such wear-out processes progress through a period of time and occur as a result of the ageing of the machine. The failure rate of a machine component throughout its lifetime can be explained with the 'bathtub' curve shown in Figure 4.3 (top). During the useful life, the machine degrades at a constant rate of failure, which when seen in the cumulative failure rate plot (Figure 4.3 - bottom) translates to a linear degradation path [99]. However, as the machine approaches failure i.e. in the wear-out period, the degradation accelerates and deviates from the previous linear behaviour. This point of deviation can be related to the Figure 2.5, where the point in time when failure can be detected is followed by an accelerated degradation path finally leading to a functional failure. Based on this concept, a real-time monitoring scheme is developed.

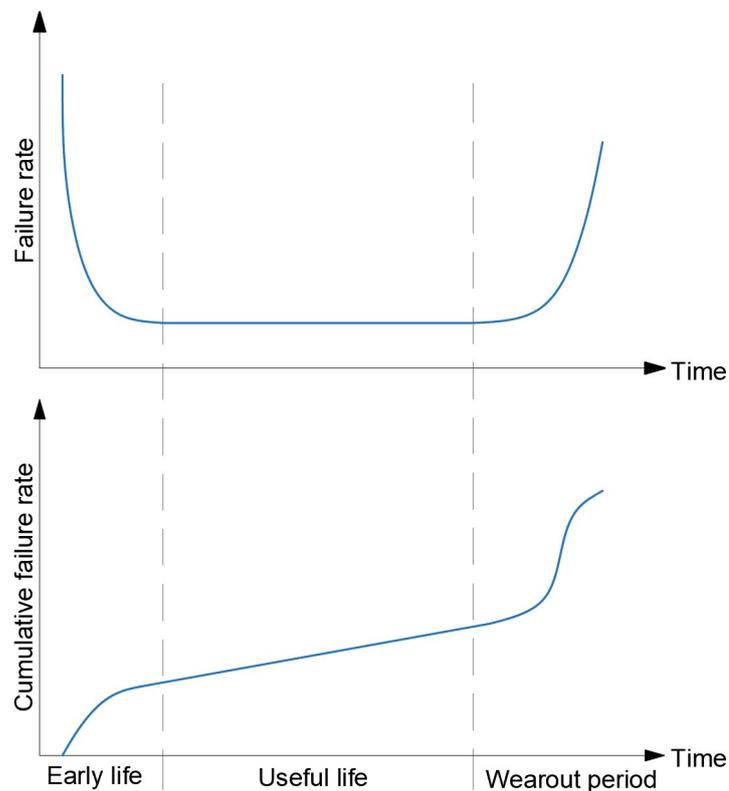


Figure 4.3: Bathtub curve showing instantaneous failure rate over the lifetime of a WT component (above) and cumulative failure distribution (below) [99]

The developed monitoring scheme uses a linear regression model to track the progression of detected anomalies in time and any positive deviation from this behaviour can be interpreted as the transition from constant to accelerated degradation rate. The scheme alarms a maintenance alert, which would allow for early detection of fault and provide enough lead time to plan and execute maintenance activities. For each incoming week the linear regression model predicts a range of acceptable values (95% confidence bounds) for the percentage of anomalies utilizing information from *a priori* data. With each progressing week, the parameters of the model are iteratively updated making the threshold adaptive and robust. The 95% confidence interval (CI) bounds are estimated using the bootstrapping method. The bootstrapping method is advantageous when working with a relatively small amount of data as it estimates the CI bounds without assuming that the data is distributed normally and bases its estimation on the distribution observed from re-sampling the data [100]. Such a bound is required to take into account the uncertainties in anomaly detection due to ML model performance or even the outliers in the data, thus, increasing the reliability of the alarm threshold. Another benefit of such an alarm threshold technique is that it could be used for turbines with no failure data. The implementation

of this scheme is further discussed in Section 6.4.

4.2 Sensitivity study

One of the key features of this research stems from the fact that it utilises high-frequency SCADA measurements instead of 10 min averaged data. To investigate the potential of such high-frequency data, a sensitivity study is undertaken to verify the effect of different sampling periods of SCADA data for the purpose of wind turbine condition monitoring. This is carried out by implementing all the modules of the developed framework as mentioned above while varying the sampling periods of the SCADA data. For each sampling period, the healthy instances of the turbine are pre-processed using the same filtration steps as mentioned in Sub-section 4.1.1, which are then used to train, test and validate the ANN-based NBM model using the 70%-15%-15% split as mentioned in Sub-section 4.1.2. The model performance is evaluated for all phases of model development using appropriate metrics (R^2 and $RMSE$). The next step is to assess the percentage of anomalies detected considering the faulty instances of the wind turbine operation as described above for respective sampling periods. The results derived from the sensitivity study are further elaborated and discussed in Section 6.5.

5

Data analysis & Model development

This chapter discusses the implementation of the methodological approach followed during this research project as introduced in Chapter 4. As the framework is divided into three modules, the application of each of them is explained in respective sections. Module 1 entailing data pre-processing is presented in Section 5.1 which provides an overview of the data used in this project and illustrates each step of Module 1 as they are executed with this dataset. Section 5.2 discusses the development of the ANN model (Module 2) and its hyper-parameter tuning to ensure good performance and generalization. Section 5.3 explains the implementation of the OC-SVM model (Module 3) for anomaly detection.

5.1 Module 1: Data pre-processing

5.1.1 Data Overview

The dataset used in this research was collected and shared by the National Renewable Energy Laboratory (NREL). NREL is a federally funded research and development centre sponsored by the USA Department of Energy. NREL's Flatirons campus is the home of the National Wind Technology Center (NWTC) which is a research facility situated in Colorado, USA and focuses on wind energy technology research. At the NWTC, a control advanced research turbine (CART2) was installed with the main focus on conducting state-of-the-art wind turbine control research. Although testing of fault detection and prognostic techniques was not necessarily the objective, a wide range of sensors was placed in the turbine. The dataset record produced by these sensors can now be used for other purposes as well, which, in this case, is wind turbine gearbox prognostics.

The CART2 turbine configuration has a rotor diameter of 43.3 m and reaches a rated generator power of 600 kW at a wind speed of 11 m/s. The maximum aerodynamic torque generated by the rotor is 162 kNm at a rated rotor angular velocity of 41.7 rpm, commanding 3.524 kNm of applied generator torque for rated operation [102]. Additional turbine characteristics are summarized in the Table 5.1.

The CART2 is a modified Westinghouse WTG-600 wind turbine that was originally installed on the island of Hawaii and was operated for about 10 years. In its original configuration, the turbine used a synchronous generator, fluid coupling and hydraulic collective pitch actuation. To enable advanced control research for variable speed wind turbines, the CART was retrofitted with an electro-mechanical pitch actuator system hosting a servo motor coupled to a gearing system to allow for high bandwidth individual pitch control. Additionally, the generator and power electronics were upgraded for a 650-kW squirrel cage, induction speed generator with a variable speed mode [103].

CART2, being a research turbine, is outfitted with a substantially greater number of sensors than normally would be installed in a commercial turbine. The sensor system includes, but is not limited to, pressure transducers, strain gauges, thermometers, position encodes, accelerometers, anemometers, wind vanes and power current and voltage meters. A total of 88 measurements are recorded including pitch angles, shaft torque and rotational speed, oil temperature and pressure, power output, yaw positions, wind speed, tower acceleration, generator power / current / voltage and other control signals [101]. The sensors installed in the CART2 turbine are shown in the Figure 5.1.

The dataset shared by NREL for this research comprised CART2 operational data from field tests

Table 5.1: Key characteristics of CART2 turbine [101]

Parameter	Value	Unit
Number of Blade	2	(-)
Cut-in wind speed	4	m/s
Cut-out wind speed	25	m/s
Rotor Diameter	43.3	m
Tower Height	34.87	m
Rated Generator Electrical Power	600	kW
Rated Rotor Speed	41.7	rpm
Maximum Rotor Speed	58	rpm
Rated Generator Speed	1800	rpm

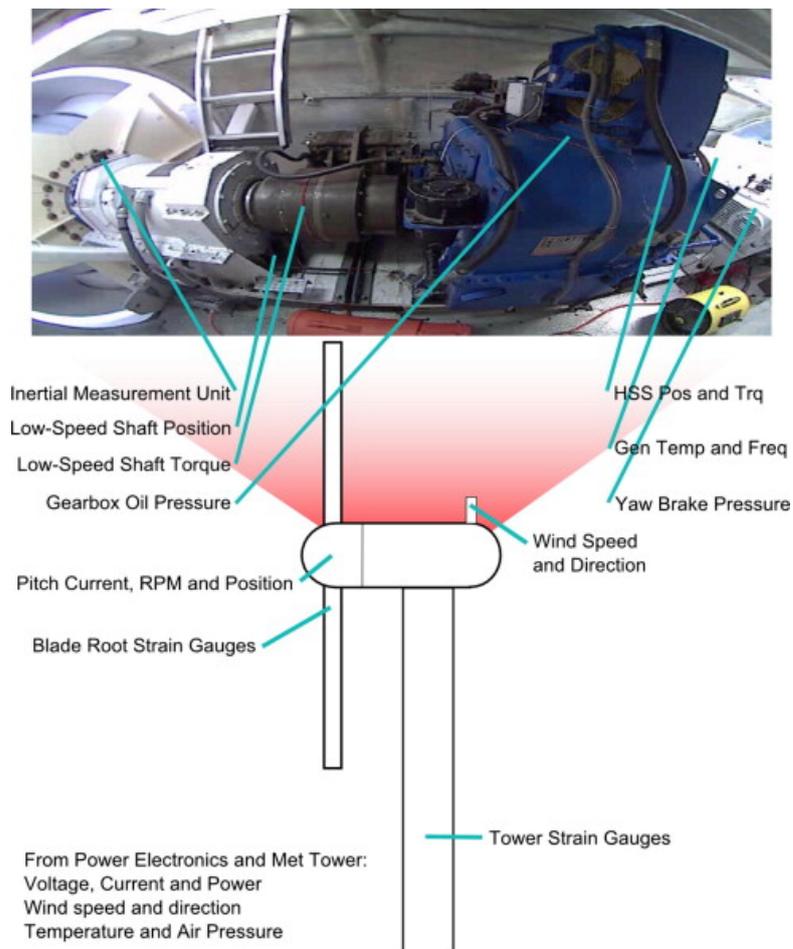


Figure 5.1: CART2 sensor location [101]

recorded in the years 2002, 2008, 2009 and 2010. The CART2 experienced a gearbox failure in the Spring of 2009 (7th April'09), which led to a turbine shutdown. The root cause analysis suggested that the failure was due to the misalignment of gear teeth in the first stage of the gearbox (planetary gear failure). The gearbox was then replaced in the Summer of 2009 and the field tests resumed later in the fall of 2009.

The dataset was received in folders segregated by each operational year, which was further segmented by days, comprising SCADA data collected during the CART2 field tests. The operational log of the turbine for each day was stored in a generic data file (*.dat format), wherein each *dat* file encompassed 600 secs of operation for the CART2 turbine. Table 5.2 summarizes the number of raw data points (sampled at 100 Hz) available for this case study contained in each folder. The reason why 2002 data is not shown in the table is discussed in Sub-section 5.1.3.

Table 5.2: Data overview as collected for each operational year and associated number of raw data points

Operational Year - 2008		Operational Year - 2009		Operational Year - 2010	
Folder name	Raw data (100Hz)	Folder name	Raw data (100 Hz)	Folder name	Raw data (100Hz)
01232008	180,000	01062009	1,500,000	03302010	1,454,615
01282008	60,000	01072009	180,000	04232010	840,000
03252008	420,000	01082009	1,200,000		
06132008	180,000	01132009	420,000		
06272008	120,000	01292009	1,980,000		
10082008	120,000	02262009	1,920,000		
10162008	120,000	03202009	1,620,000		
10242008	960,000	03312009	451,411		
12242008	420,000	04072009	52,445		

In comparison with the datasets used in the literature for the analysis of gearbox failure detection and prognostics, the CART2 dataset is unique because:

- Usually the SCADA system used for wind turbine monitoring stores 10 min averages of signals measured, whereas, in this dataset the output of each sensor is recorded at a control rate of 100 Hz.
- The CART2 turbine was installed to perform advanced wind turbine control research and therefore, the data was recorded only for the days when the field tests were conducted. Consequently, in contrast to datasets used in WT PHM research, the collected sensor data is discontinuous in time series which poses a problem in understanding the degradation path of the gearbox before failure. Additionally, the main idea behind the field tests organised for CART2 was to test the performance of the turbine when adjustments were made to the control algorithm. This could pose difficulty in understanding the turbine dynamics and its effect on sensor outputs.

5.1.2 Categorization of healthy and faulty instances

One of the foremost tasks, in order to apply the normal behaviour modelling (NBM) approach, is to distinctly categorize the data corresponding to the turbine's healthy and faulty operation. '*Healthy data*' consists of data representing the normal behaviour of the turbine and the information extracted from this data is then used to model its healthy state. On the other hand, '*faulty data*' comprises data indicating deterioration of the machine component before the fault occurs, hence, containing information of the fault initiation and development. The developed NBM output is compared with the faulty data to identify faults based on the deviation from normal behaviour.

To that end, firstly the exact failure date was determined. Based on the available data, the final day of operation for the CART2 turbine in Spring 2009 was assumed as the date of failure i.e. 7 April 2009. Once the failure date was determined, SCADA data for 4 months up to failure was stored separately and termed as '*faulty data*'. A 4 month time period was selected for this study to understand the progression of the fault and detect the earliest signs of failure inception to have a longer maintenance window, if possible. The data from 2008, for all periods up to 6 months before the failure occurred were identified as the representation of the turbine's healthy operation and termed as '*healthy data*'. The data from 2002 is too far behind in the timeline from the date of failure and hence was not used. Additionally, keeping in mind turbine's degradation through time, a normal behaviour model based on data from 2002 would not be ideal to compare turbine component's health close to failure and could lead to false/missed alarms. Another period of data collection that was rejected for the use of framework implementation was from October 2009 to April 2010 - after the gearbox failure and replacement took place. The reason for not considering this data as the 'healthy data' is twofold: (1) as the failed planetary gear was replaced, the sensor data would reflect the turbine's operation with the new component and would not be apt to analyse the failure of the replaced component; (2) the field tests conducted after the gearbox replacement were performed to test a new control algorithm and its fine-tuning would cause SCADA data errors.

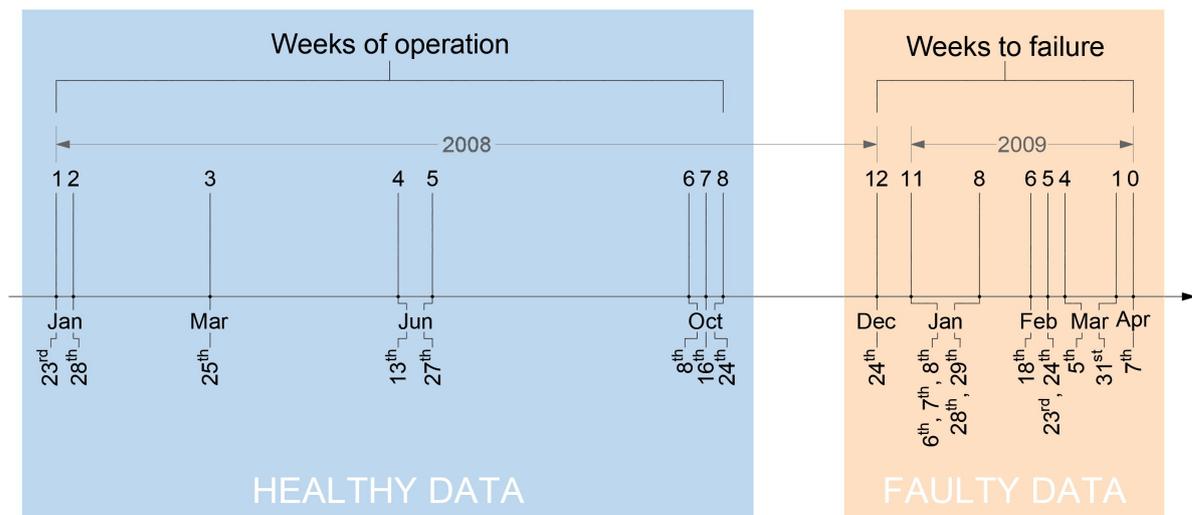


Figure 5.2: Weeks of operation and corresponding dates in which the data was available for healthy data (blue) and weeks to failure and corresponding dates for faulty data (orange)

As mentioned above, the healthy dataset refers to data from January to October 2008 (i.e. all available samples in 2008 up to 6 months before the gearbox failure occurred). Furthermore, this dataset is segmented with respect to the weeks of operation. Figure 5.2 summarizes the 8 available weeks of CART2 operation (ranging from 1 to 8) and the corresponding days of field tests performed in respective weeks. On the other hand, the faulty data for each day belonging to the SCADA data log 4 months prior to the turbine failure that took place in April was analysed in terms of 'Weeks to failure'. This segmentation was done based on the known failure date (7 April 2009), which is also shown as '0 weeks to failure' or the 'week of failure' and counting the number of weeks leading up to this 'week of failure'. Figure 5.2 shows the weeks to failure, wherein the first day considered in the faulty dataset

was 24 Dec 2008, which is 12 weeks before the gearbox failure occurred and the turbine was shut down for maintenance. The gaps within the weeks of operation in healthy data and weeks to failure in the faulty data as shown in Figure 5.2 demonstrate the discontinuous operation of the turbine in the time domain.

5.1.3 Data cleaning

In any data-driven modelling approach, the ML algorithm learns the input/output mapping based, solely, on the data provided during the training stage. Hence, it is essential that the training data is free from outliers / bad data. In the real world, however, there seldom exists a perfect dataset, and often, the SCADA data are found to contain inconsistencies attributable to the noise in the data. These inconsistencies lead to inaccuracies in the ML models and hence need to be dealt with in an appropriate manner.

In this regard, the first challenge that the CART2 dataset presented was the size of the raw 100 Hz data and the recorded signal quality at this sampling frequency. In machine learning terminology such a case could be seen as a big data problem. For each hour of turbine operation, considering all the signal features, 31,680,000 data points are recorded. Hence, in order to reduce the dimensionality of the data, it was decided to down-sample the original 100 Hz data to 1 Hz by considering the median values for every second of operation. This simple strategy provided two advantages: (1) the data was reduced to 10% of its original size; (2) the number of outliers in the original data were significantly reduced by selecting the median values, which is a robust statistical parameter that is not influenced by data outliers [104]. Furthermore, the failure characteristics would still be evident in the down-sampled data. Figure 5.3 shows the wind turbine power curve for the healthy data utilizing the raw 100 Hz data on the left and the re-sampled 1 Hz on the right. It can be observed that the raw data has considerably more outliers as even the characteristic wind turbine power curve is not visible.

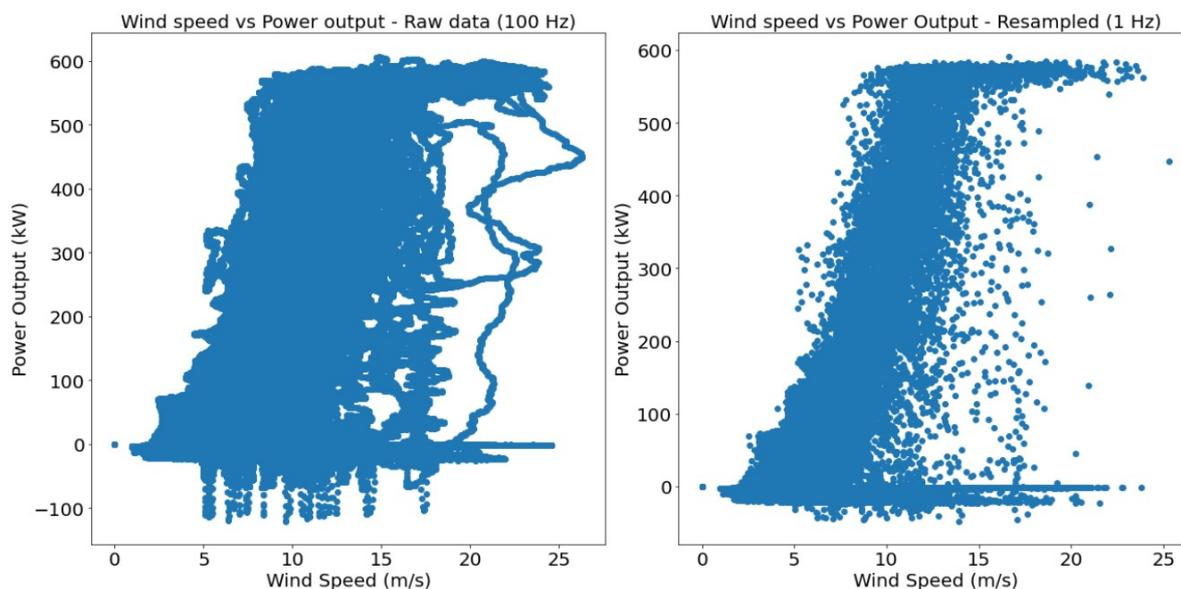


Figure 5.3: Unfiltered wind turbine power curve for healthy data with raw data sampled at 100 Hz data and re-sampled at 1 Hz data

Furthermore, the data provided still had to be filtered of 'bad' data points due to malfunctions of the SCADA system or sensors, periods of emergency shutdowns, or even erroneous recorded sensor values. The following rules were utilized for filtering the outliers/garbage data out:

1. Filter out all signal data vectors where a turbine emergency shutdown was initiated due to [101]:
 - a safety-critical sensor being damaged and not providing trustworthy data. An example of this is an incorrect blade pitch encoder signal interfering with the pitch control.

- a safety-critical component or subsystem being damaged or not operating properly. For instance, the torque or rotational speeds for the high-speed or low-speed shaft might show large deviations from the expected values for a certain power output of the turbine, indicating a fault in the gearbox components or sub-components.
 - the turbine being in an unsafe state because of external factors or behaviour of the controller. Examples include over-speed operation and large nacelle accelerations. Also, in this case, a number of shutdowns were initiated during the field tests.
2. Filter out all data vectors where one or more parameters have a value higher than a predefined threshold. In this report, the threshold values are decided based on the specification sheet for sensor values shared by NREL. For instance, all negative wind speed measurements, negative power values, measurements with a gearbox oil temperature greater than 200°C etc. are filtered out.
 3. Filter out all data vectors that correspond to a situation where the wind turbine is not producing any power. This can be done by analysing the power curve of the turbine and filtering out data points based on the following criteria:
 - All data vectors corresponding to wind speeds less than the cut-in wind speed and power values greater than 0 are filtered out.
 - All data vectors corresponding to wind speed greater than the cut-in wind speed and power values close to 0 are filtered out.
 - All data vectors corresponding to blade pitch anomalies, curtailment losses are filtered out. This is done by investigating the blade pitch vs rotational speed of the low-speed shaft. In the constant power region, the pitch control algorithm is designed to regulate the rotor speed to 41.7 rpm (rated rotor speed as presented in Table 5.1) and therefore, the blades start pitching as the turbine operates above rated wind speed until the rated rotor speed is achieved. If the blades pitch before the turbine achieves this predefined rotor speed, it would lead to anomalous power values visible in the turbine's power curve.

In Figure 5.4, on left-hand side, anomalous blade pitch values in both healthy and faulty datasets can be observed. These anomalies are represented by the scattered data points with high positive values of blade pitch before the rated rotational speed is achieved. In order to filter out these outliers, the quartile algorithm is employed for wind turbine operations below the rated rotational speed of 41.7 rpm. In statistics, IQR (Inter-quartile range) is defined as $IQR = Q_3 - Q_1$, where Q_1 and Q_3 are the 25th and 75th percentiles respectively. The quartile algorithm for filtering outliers demonstrate that any data point outside the range $[Q_1 - k \times IQR, Q_3 + k \times IQR]$, may be an outlier, where particularly, $k = 1.5$ indicates outliers and $k = 3$ indicates data that are 'far out' [104].

All the three filtration steps mentioned above were applied sequentially to both, healthy and faulty, datasets separately and this is presented in Figure 5.5. On the left, the unfiltered power output values with respect to wind speed, for both datasets are shown. The outliers in this scatter plot are evident such as, power values greater than 0 for wind speeds below the cut-in wind speed (the point at which the blades start rotating and turbine starts generating power), low values of output power at wind speeds greater than the cut-in wind speed, anomalous power values due to pitch anomalies (these can be observed as low power output values for wind speeds greater than 15 m/s) etc. Table 5.3 summarizes the number of data points in the original dataset and the cleaned dataset used for further analysis.

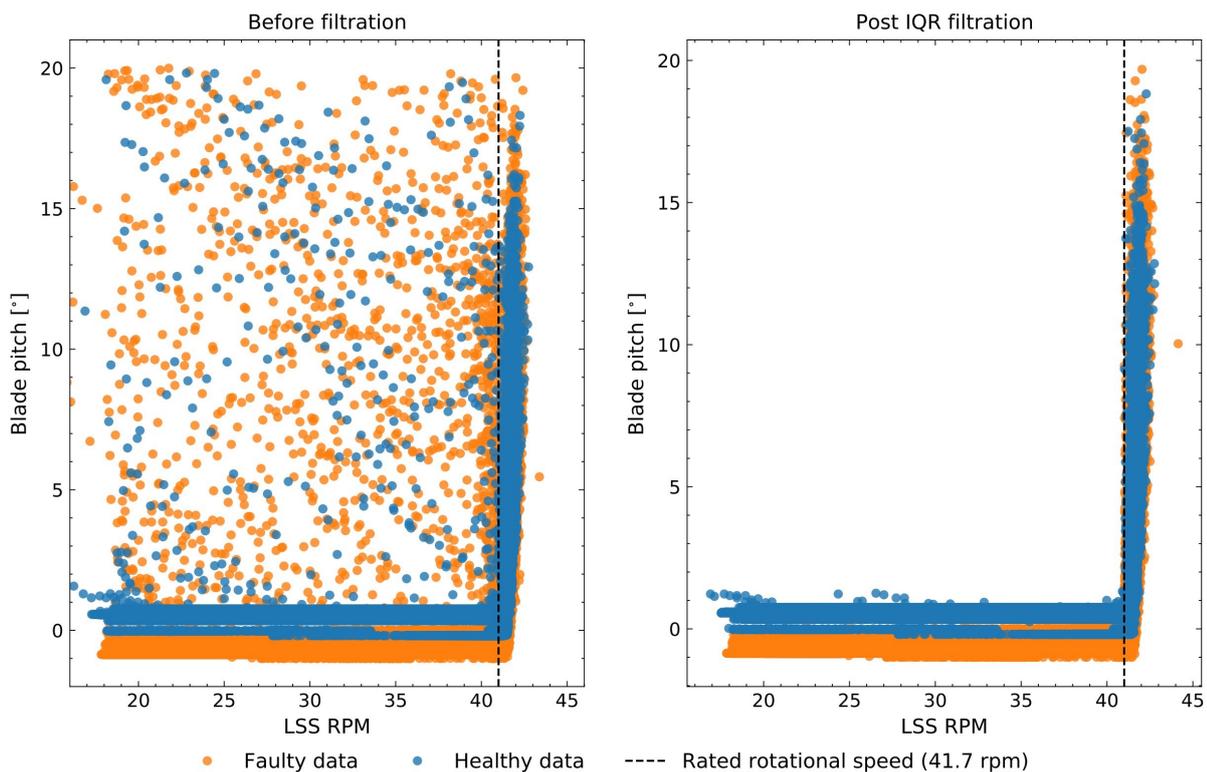


Figure 5.4: Blade pitch vs LSS rotational speed for healthy and faulty data (re-sampled at 1 Hz): (left) in original data showing pitch anomalies; (right) after filtration

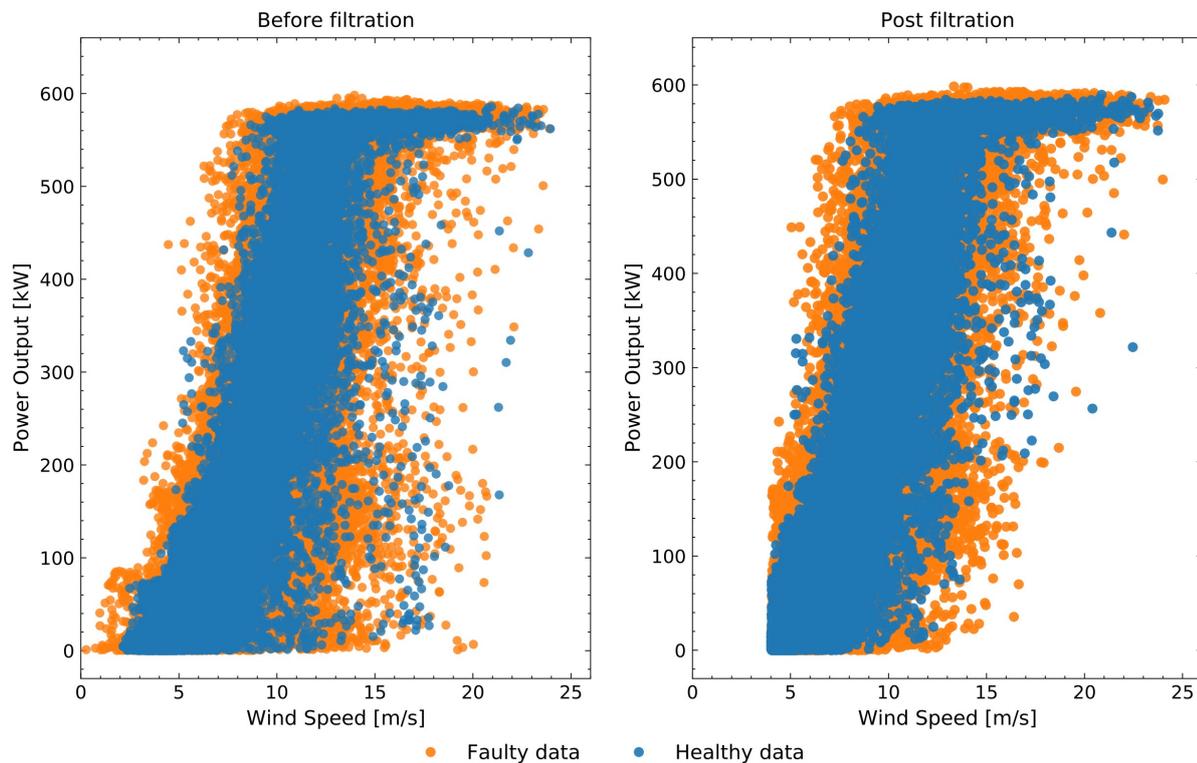


Figure 5.5: Wind speed vs power output for healthy and faulty datasets (re-sampled at 1 Hz): (left) before filtration and (right) after filtration

Table 5.3: Summary of the number of data samples in healthy and faulty datasets

Category	Raw data (100 Hz)	Resampled data (1 Hz)	Filtered data
Healthy data	3,240,000	32,400	18,399
Faulty data	9,743,856	97,434	64,766

5.1.4 Feature selection and Pre-processing

The next step in building a normal behaviour model is the selection of input features for modelling a particular output / target feature. As mentioned in the Sub-section 4.1.1, an understanding of the physics and domain knowledge was used to decide suitable features for modelling. From a condition monitoring perspective, gearbox lubrication oil temperature values are highly important, as the most common failure modes in the gearbox, would, potentially, manifest themselves into a deviation in these measurements. Hence, a normal behaviour model for the gearbox lubrication oil temperatures is utilized to perform gearbox condition monitoring. Once the target feature was determined, the next step was to identify the input features for modelling the gearbox lubrication oil temperature.

The gearbox lubrication oil temperature is directly connected to the nacelle and ambient temperature, and there exists a state of thermal equilibrium between these temperatures under normal operating conditions [105]. The ANN-based NBM can be used to emulate this thermal equilibrium condition, and any disturbance in the equilibrium may then indicate an anomalous operation in the gearbox. Consequently, the ambient and nacelle temperature measurements are utilized as input features for the ANN NBM. Furthermore, the temperatures inside the nacelle are directly related to the wind speed and power being produced by the wind turbine, as the electrical and mechanical losses are proportional to the power produced; this concept was also explored in Feng *et al.* [38] for the monitoring of wind turbine gearboxes. Hence, the wind speed and power output from the turbine were also included as input features to the NBM. Additionally, the lubrication oil is in constant contact with the gearbox components and its flow inside the gearbox would also depend on the rotational speed of the shafts. Therefore, the rotational speed for the high-speed shaft was also included in the list of input features. In summary, the ANN-based NBM utilises five input features to predict one target feature as summarized in Table 5.4.

Table 5.4: Model features for the ANN-based NBM

Feature No.	Feature	Description	Layer
1	U_{wind}	Wind speed	Input layer
2	P_{out}	Power output	Input layer
3	U_{Hss}	High-speed shaft rotation speed	Input layer
4	T_{nac}	Nacelle temperature	Input layer
5	T_{amb}	Ambient temperature	Input layer
6	T_{gb}	Gearbox lubrication oil temperature	Output layer

Data Pre-processing

Before training the ANN with the selected input and output features, it was important to ensure that the training data is balanced across all power domains of the wind turbine. Figure 5.6 shows the density distribution for the healthy and the faulty datasets across all operation power domains (0 - 600 kW). It can be observed that the density plot for the healthy data is left-skewed which means that it consists of data points mainly referring to low power output values, i.e. between 0 - 100 kW. On the other side, the faulty data shows an even distribution across all power regions with a slight peak at high power values. Training with such imbalanced healthy data, would lead to over-fitting the model to predict accurately in the turbine low-power operation regions but perform poorly for the other power regions. Therefore, to combat this issue, the SMOGN pre-processing approach, described in Section 3.2, was applied to the healthy data.

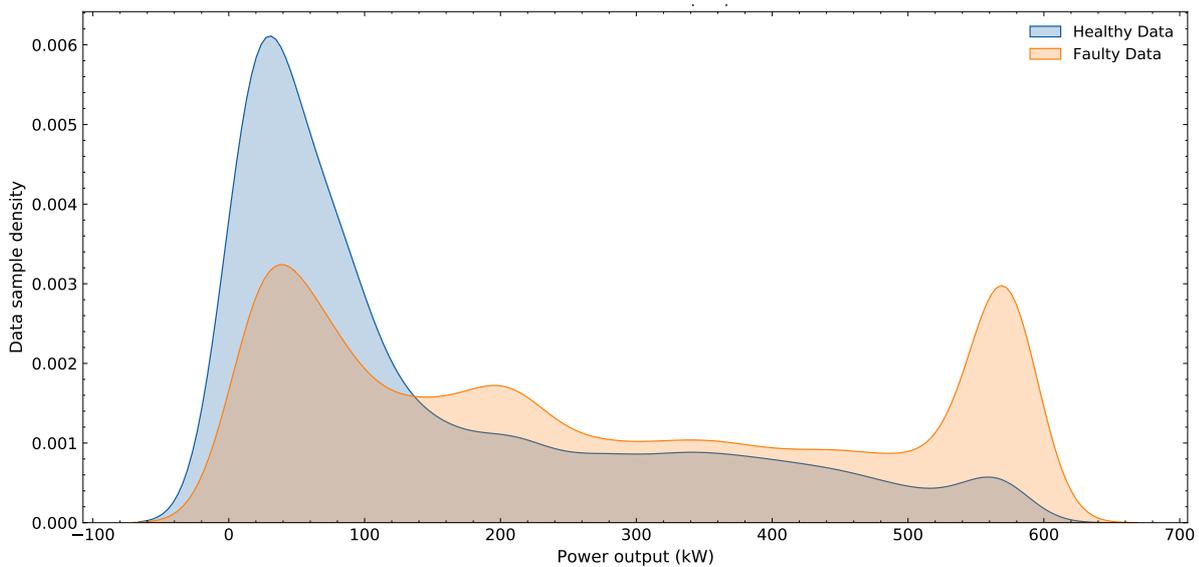


Figure 5.6: Data density plot for healthy and faulty data across all power domains

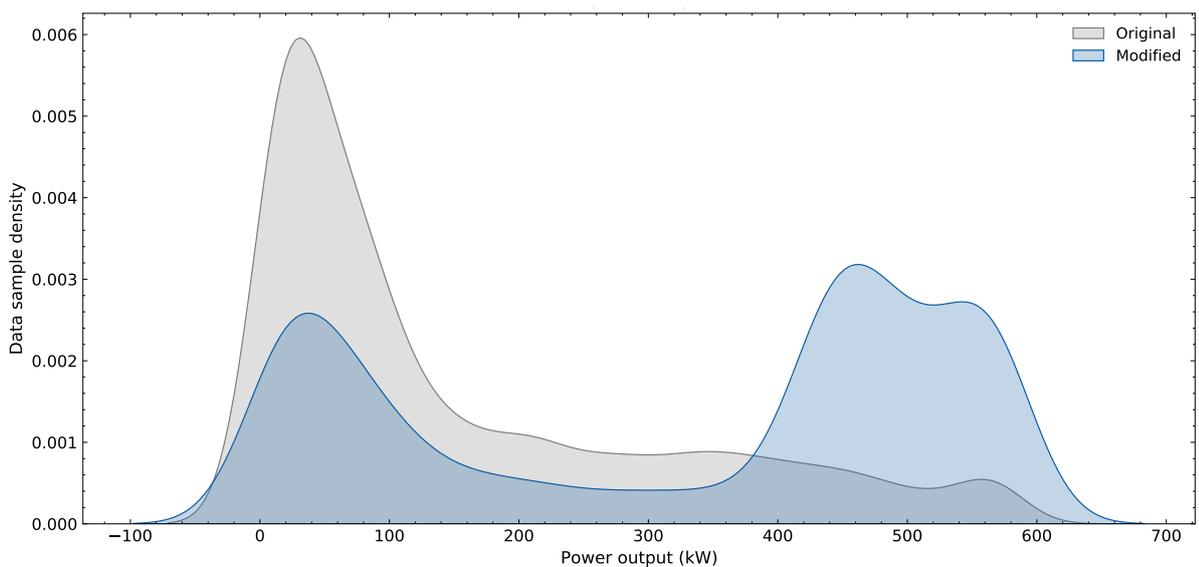


Figure 5.7: Data density for original and modified training data after SMOGN application

The algorithm first builds a data partition which is based on the box plot, where, the data points which

are closer to the median are classified as normal and less important and the data points closer to the box plot extremes are classified as the rare and important partitions. The samples belonging to the former class are undersampled using the random sampling strategy, whereas, the oversampling for the latter class is done using k-nearest neighbours. An additional feature of this algorithm is that if the selected nearest neighbours are too far from the data points, the random samples are synthetically generated by the introducing Gaussian noise, thus making the oversampling more robust. It is worth mentioning that this feature engineering approach is only applied to the training data (i.e. 70% of the healthy dataset) as it is important to verify the model accuracy with actual data and not with synthetically generated samples. This additional step is taken into consideration to improve the generalisation capabilities of the model and prevent any case of false alarms. Figure 5.7 shows the application of SMOGN to the training data. In the modified data, the data density is improved for both the high and low power regions. The low-power region (0 - 100 kW) has undergone undersampling whereas synthetic samples were generated in the high-power region. It is also worth noting, that there was no reduction in the number of data samples when the training data was modified.

Feature Scaling

Additionally, since the collected SCADA data has different features of varying scales and units, training a machine learning model with such data degrades the predictive performance as well as slow down the prediction. Hence, all the training data is normalised using a min-max scaler in the range of 0 to 1. The normalisation of the data is done using the equation 5.1 shown below [106]:

$$W' = \frac{w - \min(w)}{\max(w) - \min(w)} \quad (5.1)$$

where W' is the rescaled value and w is the original value, $\max(w)$ and $\min(w)$ are the feature maximum and the minimum values, respectively.

5.2 Module 2: Normal behaviour model

From the SCADA data log gathered 16 months prior to failure, as shown in Figure 5.2, the operation of the wind turbine for the initial 10 months (from Jan - Oct 2008 when it was known that no serious faults had occurred) was utilised for training and validating the NBM, while the SCADA data 4 months prior to failure was utilized to track error leading up to failure. Out of the 18,399 samples from the healthy dataset, as shown in Table 5.6, 70% are selected for training and 15% for validation, with the remaining 15% used to test the model independently. Such a split provides a good balance between the number of samples required for training, validating and testing the model [107].

During the training phase, the training samples were chosen at random through an algorithm and then fed into the neural network, which was then utilized to adjust the error between predicted and known values of the target variable. It was then validated with the validation samples, and the mean squared error (MSE) was calculated for the new data points. This process was repeated until the MSE no longer increased for the validation dataset, indicating that the model generalizes well and was no longer over-fitting to the training dataset. Once this was achieved, the model was then independently tested with the testing dataset (the 15% of data remaining after random selection of training and validation data) to ensure generalisation.

In order to model the NBM, a three-layer feed-forward neural network was utilised, which had an input layer consisting of five wind turbine operational parameters to predict a single output: the gearbox lubrication oil temperature - T_{gb} , as described in Table 5.5. The input features include parameters to describe the environmental conditions (wind speed - U_{wind} and ambient temperature - T_{amb}), the WT operating conditions (power output - P_{out} and high-speed shaft rotation speed - U_{hss}). Additional model parameters for the neural network such as number of hidden neurons, optimisation algorithm, activation function etc. were chosen to reflect the number of input and output neurons and the number of training samples, which in this case utilised a single hidden layer.

The number of neurons for the hidden layer was computed using Equation 3.4, by varying the parameter α , to optimize the balance between the accuracy of prediction and computational time for training. The value of α was set to 5, and based on the number of training samples and neurons in the input and output layer, the number of neurons in the hidden layer was set to 343. The neural network model used 'Adam' - adaptive momentum estimation algorithm - as the solver for optimizing weights. When compared to its counterparts namely - limited memory-BFGS and stochastic gradient descent algorithms, Adam performs better on relatively larger datasets (usually on the scale of more than 1000 training samples) [108]. Figure 5.8 shows the schematic diagram for the feed-forward neural network utilized for modelling the NBM in this case study and all its parameters are listed in Table 5.5.

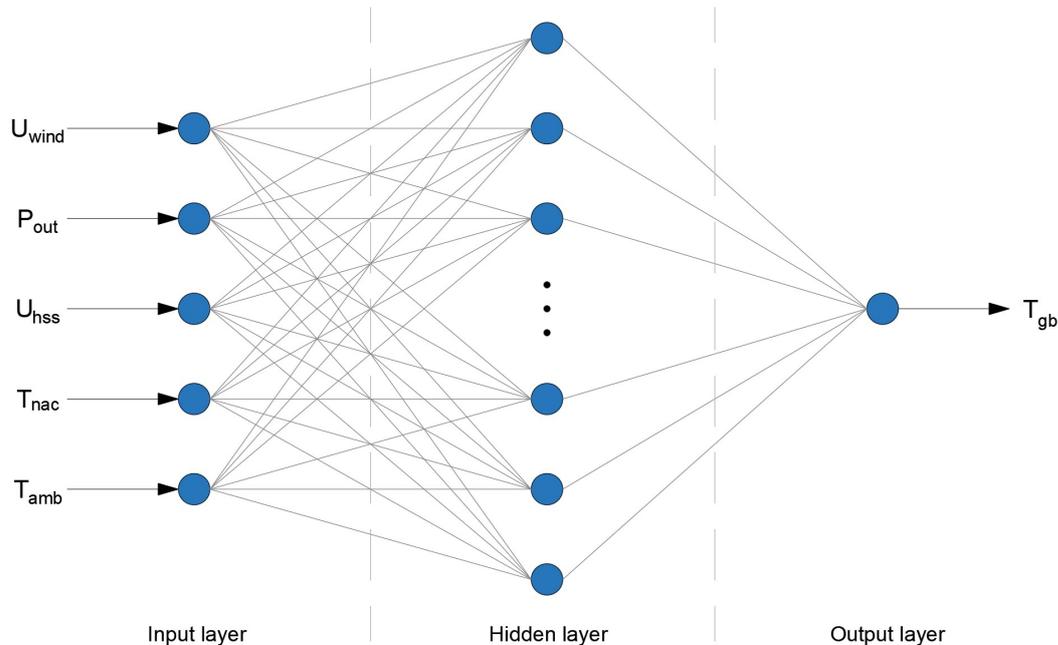


Figure 5.8: Schematic of three-layer feed-forward neural network

Table 5.5: Key characteristics of ANN model

Attributes	Value
Number of input neurons	5
Number of hidden neurons	343
Number of output neurons	2 (1 hidden layer and 1 output layer)
Loss function	Adaptive moment estimation (Adam)
Learning rate	0.001 (constant)

Once the neural network was trained and optimized using the healthy data, the same input features as mentioned in Table 5.4 from the faulty dataset were then fed into the ANN model to predict the gearbox lubrication oil temperature. The fault in the gearbox then manifests through analysis of the residual error between the predicted and the actual temperature values.

In order to predict the gearbox failure using the faulty data, based on analysing the lubrication oil temperature, it was ensured that there was no ambiguity in the interpretability of the anomaly detection

Table 5.6: Data summary for training, validation, testing and implementation of ANN model

Model phase	Number of samples
Model development (Jan - Oct 2008)	18,399
Model development-Training (70%)	12,873
Model development-Validation (15%)	2,758
Model development-Testing (15%)	2,759
Model implementation (Dec - Apr 2009)	36,309

results. As mentioned before, the operational days of the CART2 turbine were grouped into respective weeks to represent the 'Weeks of operation' in case of healthy data and 'Weeks to failure' for the faulty data. However, when analysed carefully, it was observed that the healthy dataset does not comprise any week with continuous operation of the turbine for two or more consecutive days. On the other hand, the faulty data includes 2-3 continuous turbine operational days in 11, 8 and 5 weeks to failure. If all these consecutive days of operation in the faulty data are considered for failure analysis, it could raise an important question while interpreting the results: *'Is the observed temperature rise in the lubrication oil due to the turbine's continuous operation or because of the incipient failure in the gearbox?'*. The NBM model would not be able to differentiate between the temperature rise caused due to these two different reasons and could flag incorrect anomalies. Hence, to take care of such ambiguity, only the first days of operation in all 'weeks to failure' were considered. This ensures a fair comparison with the data considered for the healthy operation of the turbine. Therefore, 36,309 data samples were used to describe the turbine's operation in all weeks, 4 months before failure. Table 5.5 summarizes the number of data samples available for each phase of the model development including training, validation, testing and implementation.

5.3 Module 3: Anomaly detection and Prognosis

After training and optimizing the ANN, the next step was to establish a baseline residual error on the basis of the residuals between the actual and predicted temperature values as observed in the healthy dataset. Once this baseline error is determined, there are several ways to then compare residuals from the faulty data. A typical approach in this regard is to compare the daily or weekly RMSE with the RMSE of the training period to give an indication of whether the temperature (or any other chosen parameter) aligns with the predicted values. Such an approach does come with limitations, stemming mainly from the fact that only one parameter is used to describe an error which, over any particular time period is multifaceted and has a unique error distribution associated with it. Another approach could be looking at more than one parameter in isolation, or even by analysing the entire distribution and tracking them in time, however, this introduces a different problem: how to robustly set thresholds that will indicate the fault.

To address the limitations with the existing approaches, a OC-SVM model was utilised, firstly by extracting multiple parameters (residual error features) that can effectively describe the distribution of error over a chosen time period and secondly, to set more complex boundaries that can more precisely describe the threshold to indicate a fault.

A OC-SVM model was developed to evaluate the error (difference between actual and predicted values) distribution for each minute based on the NBM output. A one minute time period is chosen to take into consideration the inertia in recording a change in the oil temperature. For each per minute error distribution, the parameters stated in Table 5.7 were calculated and used as inputs to the OC-SVM

model. In order to train the OC-SVM model, the error features from the healthy data were selected, giving a total of 370 samples. In the SVM, the predicted values would be a result of binary classification, i.e. either +1 or -1, where a negative score would constitute an outlier or anomaly, which lies outside the decision boundary whereas a positive score would be interpreted as the turbine operating in a healthy state. The model was trained to recognise 1% of data as anomalies in the training period by defining the ν hyperparameter as 0.01; therefore, a similar percentile would be expected moving forward if no fault was present in the system.

Table 5.7: Residual error metrics

Feature No.	Feature	Description
1	e_{rms}	Root Mean Square Error (RMSE)
2	e_{min}	Minimum Error
3	e_{max}	Maximum Error
4	e_{std}	Standard deviation of error distribution

5.4 Discussion

This chapter firstly provided with an overview of the CART2 dataset and the gearbox failure data present within it and highlighted its unique characteristics. Based on the known date of failure of the turbine, the data was categorized into healthy and faulty data. Subsequently, to filter out outliers in the data, a series of filtration steps were implemented. After the data filtration process, the features for the ANN-based NBM model were selected using domain knowledge. The gearbox lubrication oil temperature was found to be the most sensitive to the condition of the gearbox and was determined as the target feature. Additional features - wind speed, power output, high speed shaft rotational speed, ambient and nacelle temperature - were chosen as the input features to the model.

Before training the ANN model, a SMOGN pre-processing approach was implemented to ensure a balanced dataset across all power regimes of the turbine operation. Owing to the dynamic nature of wind availability, real-world turbine operational data is often riddled with issue of imbalanced domains i.e. low number of data points for some power domains and extremely high number of data points for some. This might lead to poor generalization capabilities of the model due to overfitting of the model for power domains with high density of data and underfitting for power domain with low density. Although this is a relevant problem when working with real turbine operational data, an effective way to combat it has not been fully appreciated in the literature. The SMOGN approach has been proposed by Branco *et al.* [70]. However, the application of it has not yet been presented in existing literature, to that end Sub-section 5.1.4 shows the efficacy of implementing the SMOGN approach.

Furthermore, this chapter discussed the selected hyper-parameters of ANN model to obtain a good model performance. The model was trained using the turbine's healthy operational data and a 70-15-15 split for training, validating and testing was used to ensure a good balance between the number of samples available for each model development phase. Lastly, to analyse the residual error between the measured and predicted values and correctly identify instance of anomalous turbine operation, a OC-SVM was implemented. Four residual error feature - RMSE, minimum error, maximum error and the standard deviation of error distribution - were used as input to the OC-SVM model.

6

Results and Discussion

This chapter presents and discusses the results of the developed framework introduced in Chapter 5. Firstly, in Section 6.1 performance of the trained ANN-based NBM is discussed by evaluating error metrics for each phase of the model development. Then the model implementation and normal behaviour predictions are shown in Section 6.2. Deviations of the gearbox lubrication oil temperature from the predicted normal behaviour are then considered as anomalies; hence, sign of incipient fault. To quantify the error progression, anomalies are detected using a OC-SVM model. Its anomaly detection results are discussed and compared to other methodologies commonly used in literature in Section 6.3. The implementation of a real-time monitoring scheme based on linear regression and bootstrapping CI is presented in Section 6.4. Lastly, the sensitivity study and its results are discussed in Section 6.5 where conclusions are drawn on the effects of using data sampled at different frequencies on model performance and anomaly detection.

6.1 ANN model performance

A 3-layer feed-forward neural network presented in Section 5.2 model was used for modelling normal behaviour and was trained, validated and tested using the 1 Hz re-sampled healthy SCADA data. The model utilized 70% of the healthy data for training, 15% for validation and 15% for testing. The data samples for each phase were selected randomly and fed into the neural network. Two metrics, namely root mean squared error (RMSE) and R-squared values (R^2), were used to evaluate the model performance in each phase of the model development. Table 6.1 summarizes these evaluation metrics for each development phase of the model to verify its performance and generalization capabilities. As mentioned before in Section 5.2, MSE was opted as the loss function to optimize the ANN model, and the low values of RMSE demonstrate that the predicted values of the target feature are indeed close to the actual values. The R^2 values close to 1 and deviating 1.7% between datasets show that the model fits well to the actual data and generalises relatively well, which sets a good foundation when trying to detect anomalies leading up to failure. In order to get a better sense of the error distribution of this particular ANN model, its error (*actual – predicted*) for all the above-mentioned phases is shown in the histogram shown in Figure 6.2.

Table 6.1: ANN model performance metrics for each model development phase

Model phase	RMSE	R^2
Model development-Training (70% data)	0.04	0.97
Model development-Validation (15% data)	0.02	0.95
Model development-Testing (15% data)	0.06	0.95

As the training data (70%), was modified using the SMOGN algorithm and the network was trained on synthetically generated samples, it was necessary to test the model performance with the actual healthy data. Therefore, after the neural network was optimised, the input features (wind speed, power output, high-speed shaft rotational speed, nacelle and ambient temperature) from the entire healthy

data were fed to the ANN model and the comparison between the predicted and observed values can be seen in Figure 6.1. The RMSE between observed and predicted values for healthy data obtained was 1.3. It can be observed that the model performs equally well for periods of low and high operational power. The maximum deviation of model-predicted temperature is observed when there is a sudden change in output power of the turbine, due to the variability in wind speed.

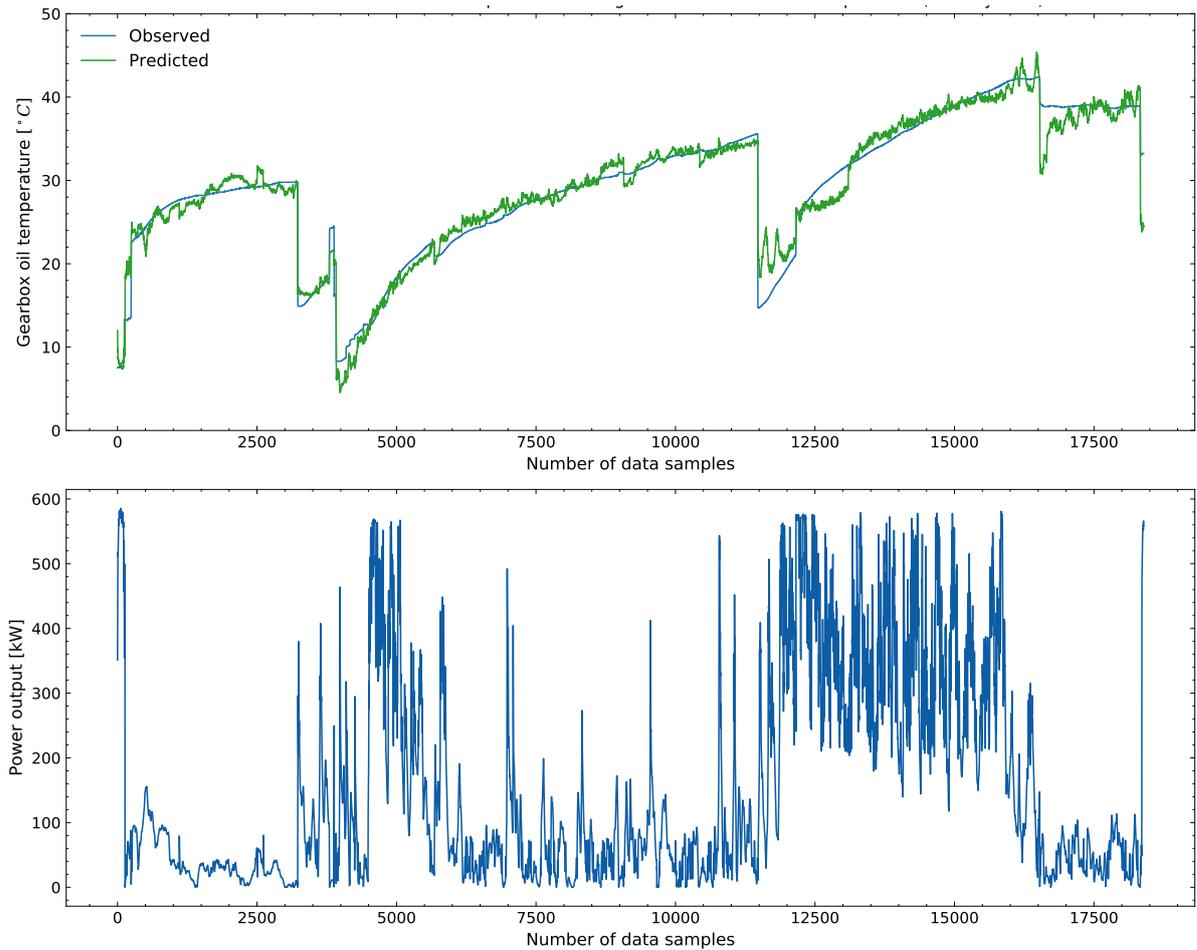


Figure 6.1: Observed and predicted gearbox oil temperature values for healthy data (top) and power output (bottom)

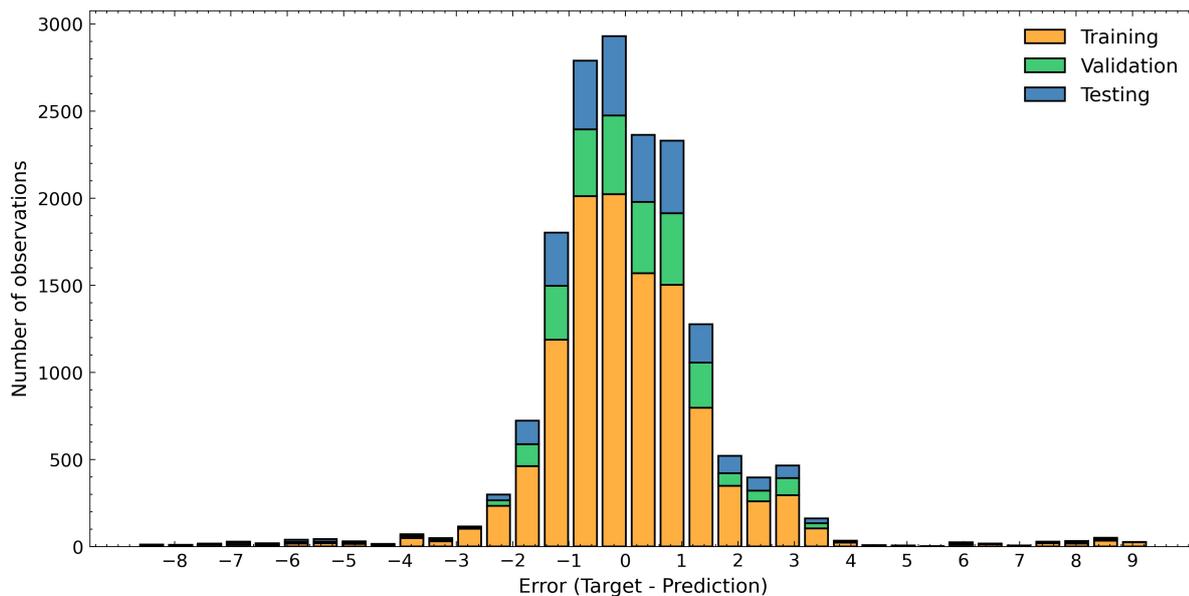


Figure 6.2: Distribution of error for all modelling phases of ANN

6.2 Model Implementation

Once a satisfactory ANN model performance was achieved, the next step was to implement the model using the faulty data for each week starting from 4 months prior to failure. In order to do so, the same input features - wind speed, power output, high-speed shaft rotational speed, ambient and nacelle temperatures - were used to predict the gearbox lubrication oil temperature. It was assumed that the incipient gearbox fault would manifest itself in an increase of deviation between the observed and predicted values. The comparison between the model predicted values and the actual temperature values for each operational week before failure can be seen in Figures 6.3 and 6.4.

From the figures, it can be observed that the deviation between the NBM predictions and the actual values of the gearbox lubrication oil temperature increases as the machine approaches failure (from 12 to 0 weeks to failure). From 12 to 6 weeks before failure (Figure 6.3), the neural network predictions show minor variations from the actual recordings, which however increase further from 5 weeks before failure to the week when the failure actually occurred (Figure 6.4). This can be seen as an increasing error between the 'observed' and 'predicted' values, which can be interpreted as a sign of fault progression in the wind turbine gearbox. It is worth mentioning that observed oil temperature values have lower values as the turbine approaches failure. Identifying a fault through the analysis of monotonically increasing signal trends, such as one done by Hu et al. [50], would misinterpret this behaviour as a sign of no-fault. This demonstrates the superiority of an ML-based data-driven approach when compared to statistical approaches such as moving average, signal trending etc.

Although the increase in deviation between the NBM predicted healthy behaviour and the actual observations of the turbine is evident, signifying signs of fault in the gearbox, it is unclear as to how this can be analysed to identify the first signs of incipient gearbox fault. Furthermore, it can be seen that ANN predictions demonstrate deviations around the actual measured values which should be taken into consideration when analysing the residual error between the two curves. When compared, the deviations were smaller for the ANN model prediction with the healthy data as shown in Figure 6.1 as it is trained on that dataset. If a simple threshold based on only training RMSE is used to analyse the model error and identify anomalies in the faulty data, even the minor deviations (any value greater than training RMSE) might get misclassified which in result would lead to false alarms. To combat this problem, a one-class SVM model was developed which could establish a complex boundary and diminish the effect of model performance while identifying anomalous behaviour of turbine distinctly caused by the incipient gearbox fault.

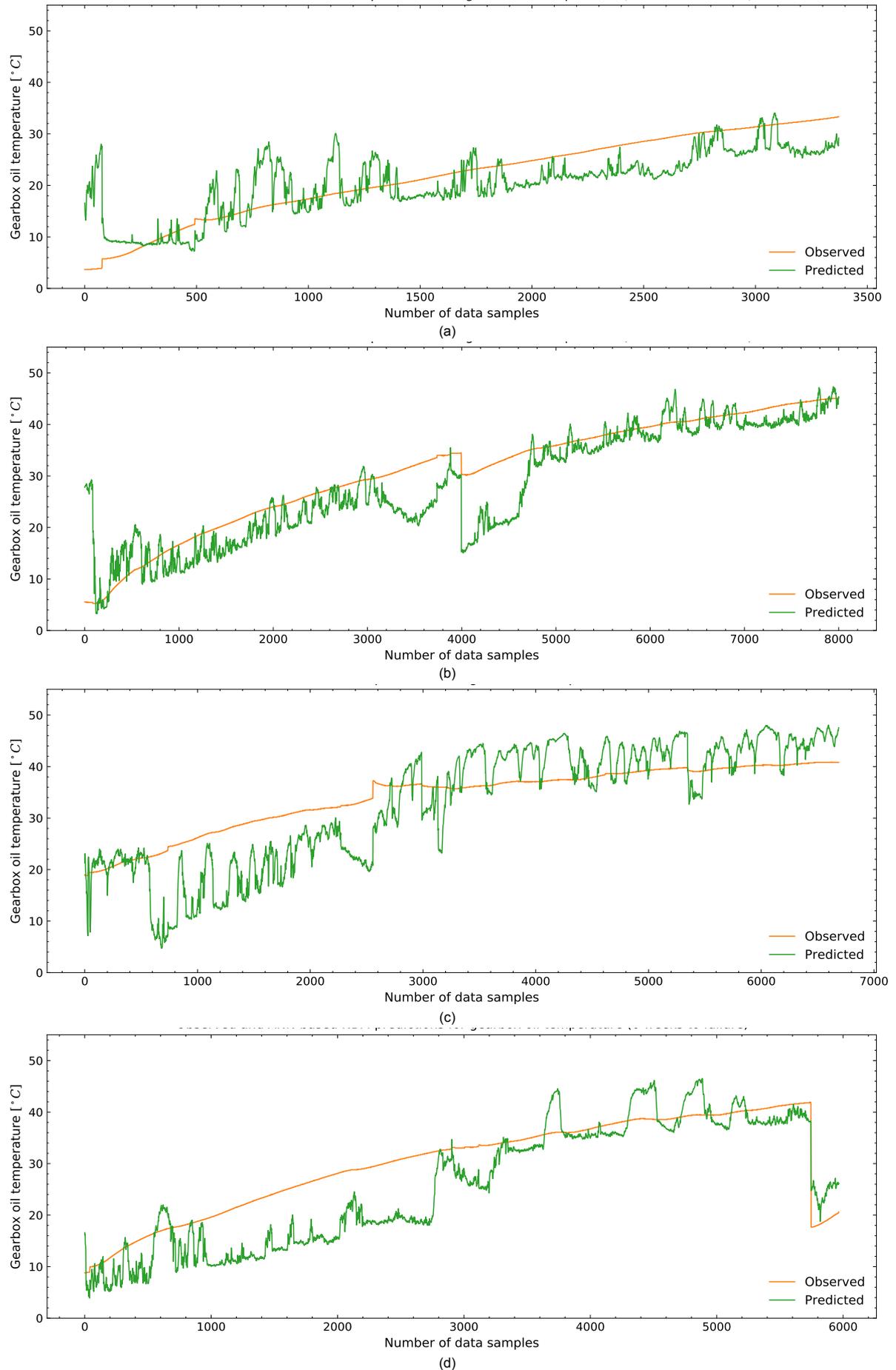


Figure 6.3: Observed and ANN-based NBM predictions for gearbox lubrication oil temperature for (a) 12 weeks to failure (b) 11 weeks to failure (c) 8 weeks to failure and (d) 6 weeks to failure

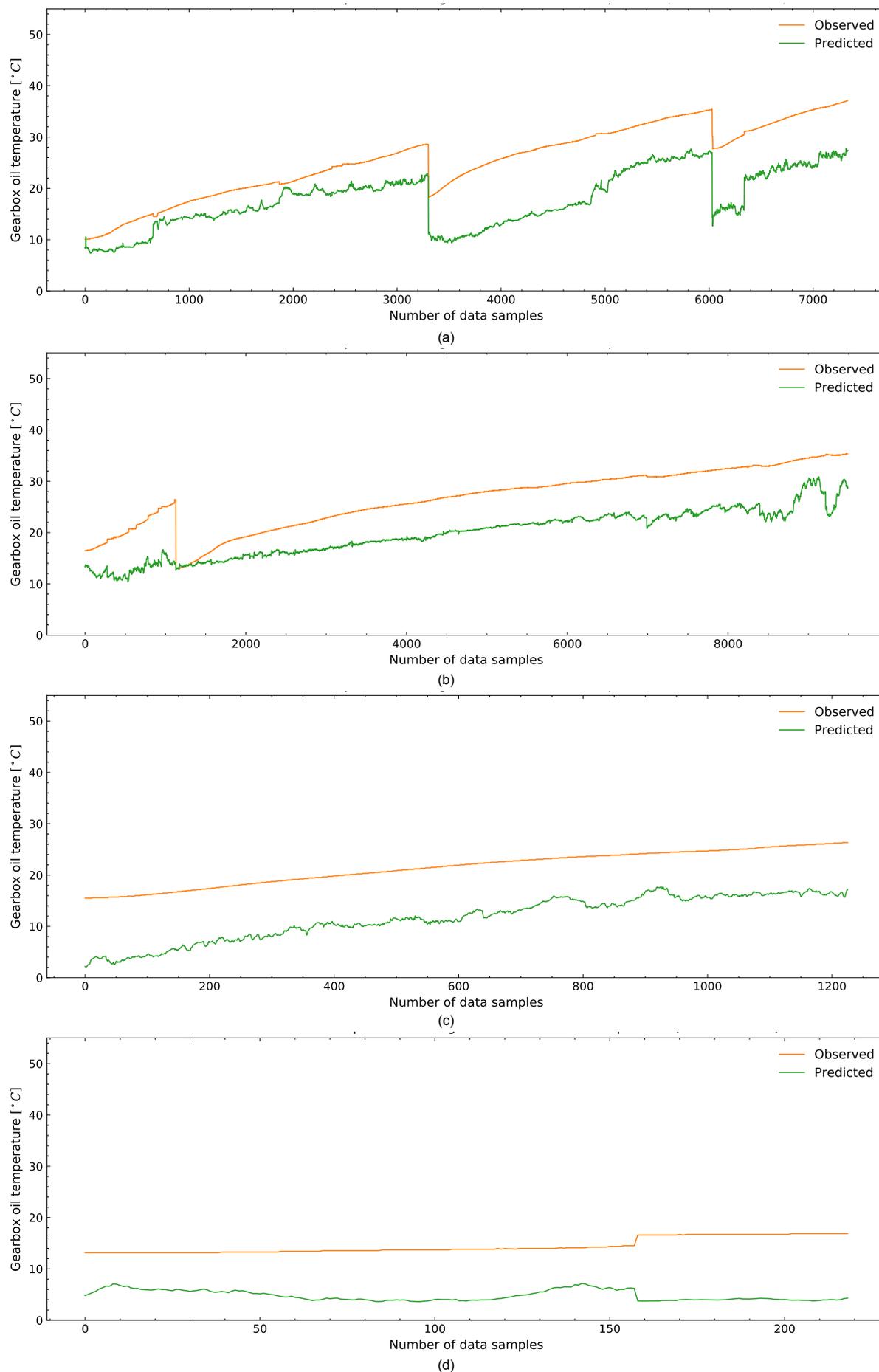


Figure 6.4: Observed and ANN-based NBM predictions for gearbox lubrication oil temperature for (a) 5 weeks to failure (b) 4 weeks to failure (c) 1 week to failure and (d) 0 weeks to failure (day of failure)

6.3 Anomaly detection results

In order to examine the error between the predicted and observed values, a one-class SVM model was developed as described in detail in Section 5.3. For every 1-min step, the error output (*actual – predicted*) was determined and the residual error features as shown in Table 5.7 were computed. The SVM model was trained using the error features calculated for the healthy data. The data samples provided during training were utilized to generate a complex decision boundary to represent the healthy state of the turbine. The SVM was trained to identify 1% of data as anomalies, therefore, when the faulty data is fed into the model, any increase in this percentage would represent a fault initiating in the turbine gearbox.

The Figure 6.5 shows a scatter plot of two of the computed features namely, Max error (e_{max}) and RMSE (e_{rms}) for both healthy and faulty data. The SVM model should be able to learn the boundary defined by the error features calculated with healthy data and detect anomalies if any data point falls outside this learnt boundary. Figure 6.6 shows a distinction between the data points classified as 'normal operation' or an 'anomaly' for the faulty data set. SVM performs well in identifying data points that fall inside the developed boundary, classifying them as healthy turbine operation by giving a +1 prediction value. On the other hand, SVM predicts a value of -1 for anomalous turbine operation, represented by data outside the defined boundary.

For each available turbine operational week, in data 4 months leading up to failure, the percentage of detected anomalies by the OC-SVM was calculated and is shown in Figure 6.7. In general, the results show an increase in the anomalies as the turbine approaches the failure, with a sharp increase in a month before failure ('4 weeks to failure') and, thereafter, an increasing trend showing 100% anomalies detected on the day of failure. These results are in line with the observed deviations between the actual and predicted values of the gearbox lubrication oil temperature as shown in Figures 6.3 and 6.4. From 12 to 5 weeks to failure, the percentage of anomalies detected increases on average by 8% with a substantial increase by 32% between 5 and 4 weeks before the failure. Once the progression of anomalies is obtained, it is imperative to understand how to set a robust threshold that could provide information about the first warning i.e. when the failure could be first detected and flag a maintenance alert allowing enough lead time to plan and execute required maintenance activities. This is done using a real-time monitoring scheme, which is discussed in the Section 6.4.

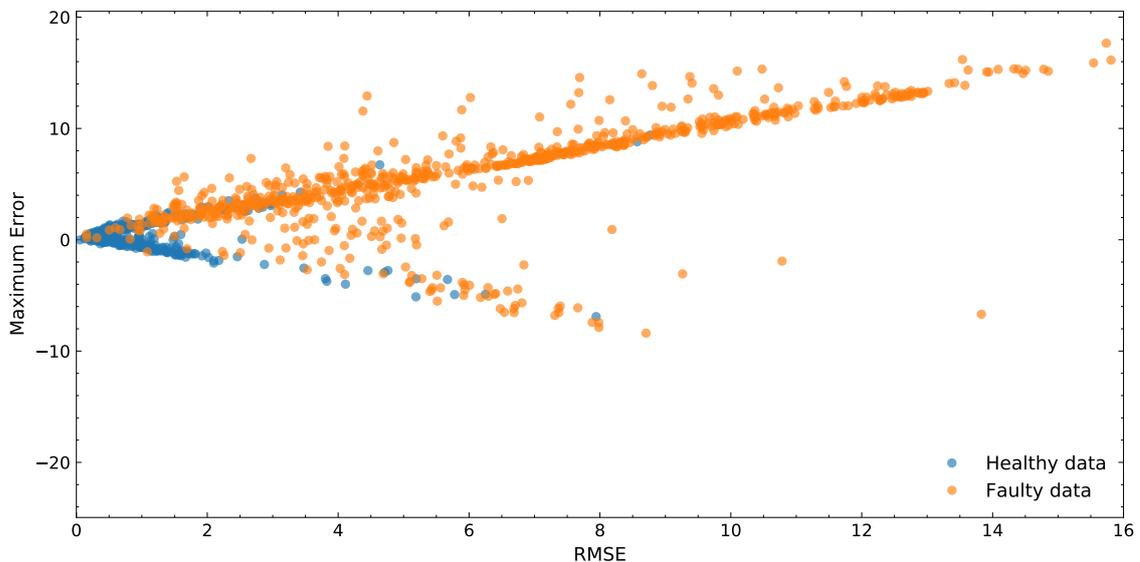


Figure 6.5: Maximum error vs RMSE - for both healthy and faulty data

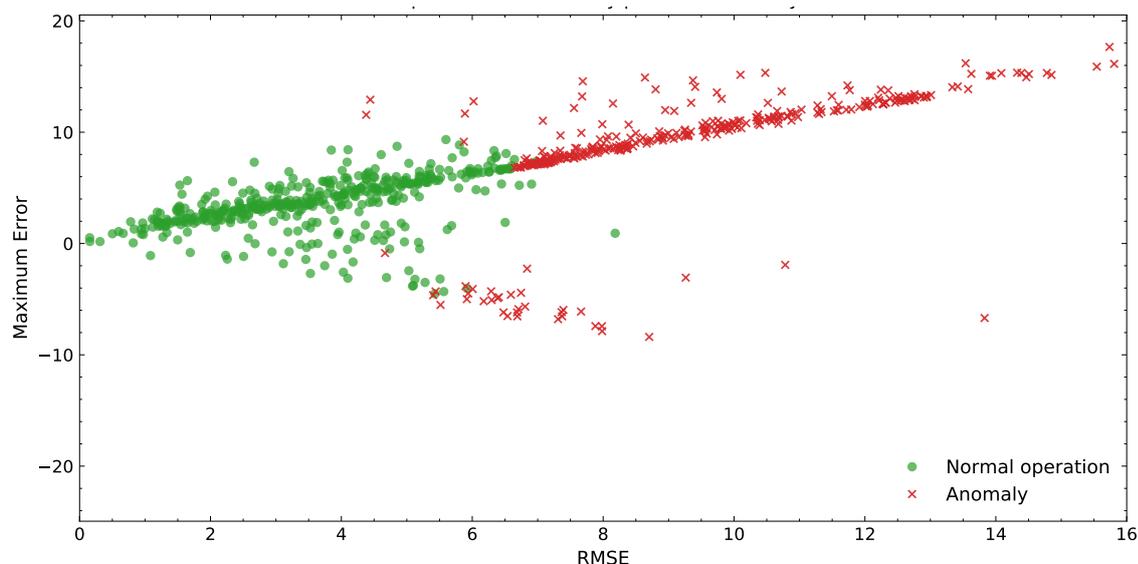


Figure 6.6: Anomalous and normal operation points detected by OC-SVM model considering faulty data

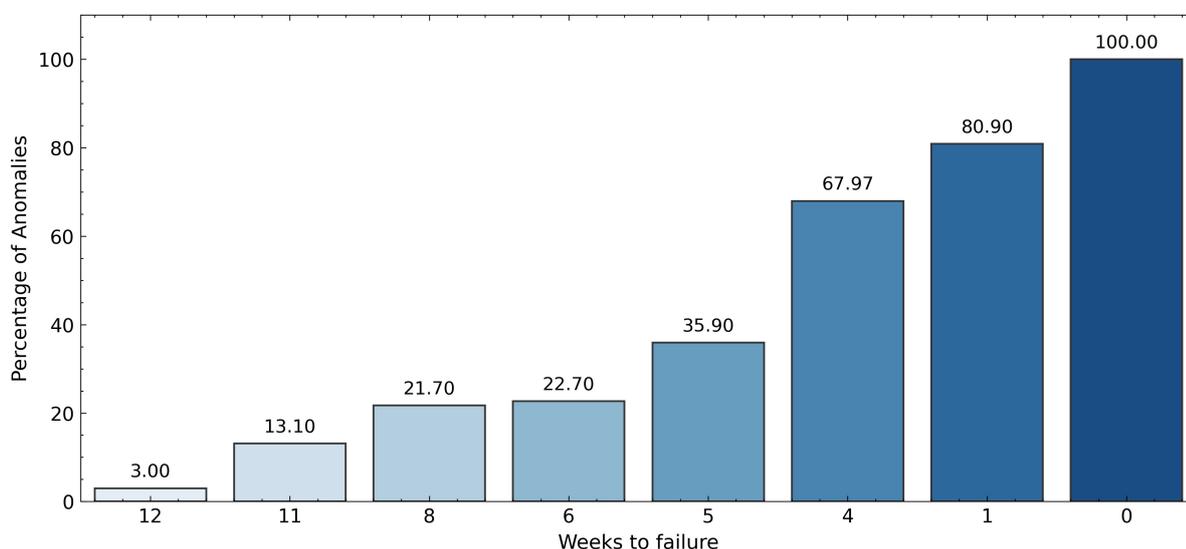


Figure 6.7: Percentage of anomalies detected for each week leading up to failure using OC-SVM

6.3.1 Comparison of OC-SVM result with other methods

As discussed in the Section 5.3, the OC-SVM combines different statistical features to understand the complex boundary between the healthy and anomalous operation of the turbine. However, it is interesting to compare the results obtained from the OC-SVM model with the existing standard methodology based on single statistical feature used in literature [96] [9] [109]. A common practice is to compare the RMSE values for any chosen parameter computed during the training period to set thresholds for identifying anomalies. The limitations associated with such an approach stems from the fact that only one parameter is used to describe the error between the actual and predicted values over a duration of time, which is multifarious and usually has a unique distribution associated with it.

To this end, thresholds based on the average RMSE and error standard deviation computed during the training period are used, which could then be compared to the RMSE for the data 3 months prior to failure. Additionally, the results obtained from these thresholds are also compared to the results from OC-SVM. Three different thresholds were used to compare the RMSE calculated for the faulty

data; threshold 1 was simply the RMSE, threshold 2 was one standard deviation added to the RMSE, and threshold 3 was 2 standard deviations above the RMSE. Figure 6.8 shows the comparison of the percentage of anomalies in weeks leading up to failure detected by the SVM model and the anomalies computed using the 3 simple thresholds based on *RMSE* and error standard deviation (*Std*) of the healthy data.

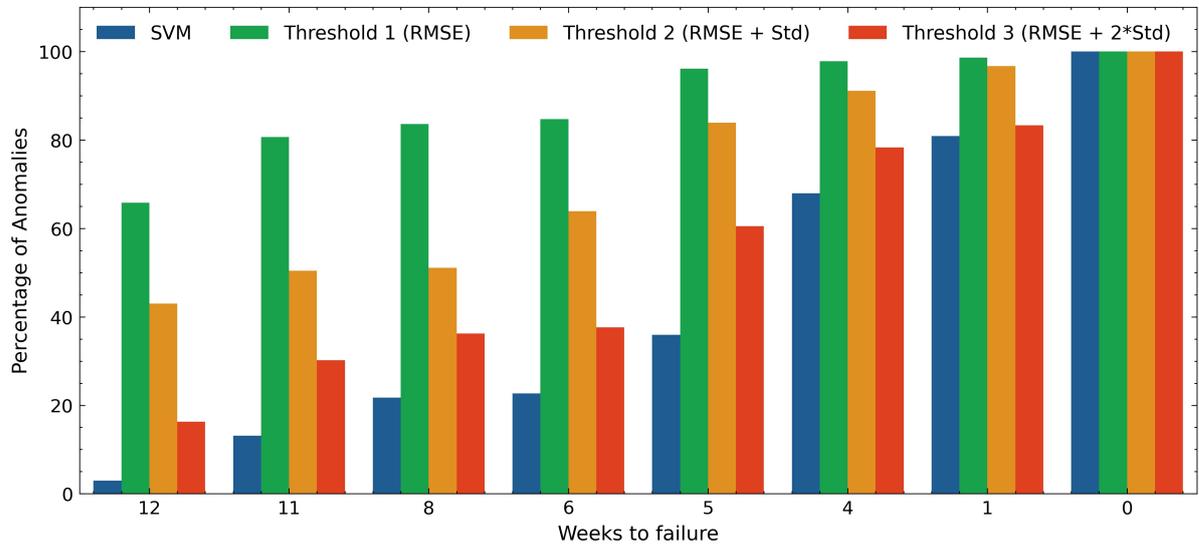


Figure 6.8: Comparison of percentage of anomalies detected by OC-SVM and standard statistical thresholds based on RMSE and standard deviation (Std)

In general, the percentage of anomalies detected with each threshold shows an increasing trend in the weeks leading up to turbine failure. However, it is worth mentioning that the large proportion of detected anomalies throughout the 3 months of turbine operation could lead to a large number of false alarms. As an example, the average percentage of anomalies for the first two months with each threshold are 73.7%, 50.2% and 36.8%, whereas, the OC-SVM model detects an average of 14.4% anomalies for the same period. Such a high percentage of anomalies detected by the RMSE based thresholds suggest that these approaches capture the small deviations between the observed and predicted gearbox oil temperature values, caused by the ANN model and not the progression of the actual fault. This means that such standard thresholds are more sensitive to the performance of the model. This can be seen in the percentage of anomalies detected from '12 weeks to failure' to '6 weeks to failure', which can be attributed to the deviation in ANN model's predictions and are not an indication of a gearbox failure. On the other hand, the SVM model detects a relatively low percentage of anomalies, giving confidence that the increase in anomalies is indeed related to the fault. This comparative study demonstrates the efficacy of using an SVM model over simple thresholding techniques. The OC-SVM combines different error features to describe the deviations between the NBM predictions and actual output over a duration of time and is less sensitive to the model performance.

6.4 Real-time monitoring scheme

Once the anomalies have been detected by the OC-SVM, the next task was to set a threshold that, when exceeded, would be capable of determining the first clear sign of fault initiation before the functional failure of the machine occurs. To this end, a real-time monitoring scheme was developed. As explained in Sub-section 4.1.3, this monitoring scheme is based on the concept of the bathtub curve and cumulative failure rate which increases linearly until the degradation becomes significant and is then characterized by a deviation from linear behaviour. Such a linear progression can also be observed for the percentage of anomalies recorded for each week before failure as shown in Figure 6.7. This can be intuitively explained from the observed trend of progression of anomalies as it captures the degradation accumulating over a period of turbine operation as it approaches failure.

The monitoring scheme developed in this project employs a simple linear regression model and estimates confidence interval limits around the model fit using bootstrapping method as described in Section 3.4. Such a scheme can be implemented in an online setting as, for each monitored week, the regression model uses the previously recorded observation or *a priori* data to estimate its parameters and develops a 95% confidence interval (CI) bound which provides an expected range of values for the percentage of anomalies for the respective week. If the recorded percentage of anomalies for that week falls within the expected CI bounds, the machine is considered to be in an operational state without the urgent need for maintenance i.e the machine still has enough remaining useful lifetime. However, if the recorded percentage of anomalies is above the upper threshold limit established by the CI bounds, it would result in a maintenance alert and would be marked as the period when the first signs of impending machine failure can be detected. As time progresses, the linear regression model takes into account the last recorded values and updates its parameters and new CI bounds are established, making the thresholds adaptive and robust. Furthermore, the 95% CI bounds account for the uncertainties in anomaly detection pertinent to ML model performance and the unfiltered outliers in the data.

The implementation of the developed real-time monitoring scheme is shown in Figure 6.9. In Figure 6.9(a), the week in monitoring ('5 weeks to failure') is represented by the green marker and the previously recorded observations (from 12 to 6 weeks to failure) are shown as blue markers. It can be seen that the linear regression model fits well to the previously recorded data and the shaded region depicts the 95% CI bounds estimated through bootstrapping. The reason behind the observed shape of this shaded region is explained in Section 3.4. The percentage of anomalies logged in this week falls within the expected CI bounds and therefore, the gearbox is expected to be working well and no urgent maintenance is required. Figure 6.9(b) refers to the '4 weeks to failure' as the week being monitored. It can be seen that the model updates its parameters and its CI bounds as the anomaly data from '5 weeks to failure' is added to the past observations and is now represented in blue. The percentage of anomalies logged 4 weeks before failure is above the upper threshold limit of the 95% CI bound. This sets an alarm for maintenance and is identified as the first clear sign of incipient fault indicating a significant change in degradation rate.

Realistically, once this alarm is set off, there would be no need for recording any more data as it would potentially result in a turbine shutdown followed by further maintenance activities. However, due to the availability of run-to-failure data for the CART2 turbine, the monitoring scheme can be further evaluated and can be seen in Figure 6.9(c) and (d). Figure 6.9(c) shows the updated linear regression model taking into account the log observed in 4 weeks before failure and it can be seen that the upper limit of the established 95% CI reaches a value of 100% for the week of failure, hence, signifying an expected potential breakdown of the gearbox in a month. It is worth mentioning that the observed percentage of anomalies for 1 week before failure is found to be within the acceptable CI bounds which could be classified as a missed alarm, however, it can be attributed to the missing data for weeks between 4 and 1 week before failure. The effect of missing data is also visible as the width of the 95% CI increases significantly from the previous analysis done with data from 4 weeks before the failure. A similar observation can be made for Figure 6.9(d), where the CI bounds again predict a potential breakdown in the coming week (taking into account '1 week to failure' data) with the upper threshold limit depicting 100% anomalies in a period of a week.

The developed real-time monitoring scheme shows good results in identifying the first clear signs of incipient fault in the gearbox and sets off a maintenance alarm a month before the actual failure, which provides enough lead time to plan and execute required maintenance activities. Another advantage of such a scheme is that it can be implemented for a turbine with no run-to-failure data. Additionally, while implementing this scheme in a real-world scenario, it would be beneficial to specify an optimal number of weeks prior to the week in monitoring to be used for modelling the linear regression fit. This could be advantageous in reducing the computational requirement and visualize the trend of anomalies better. Albeit its advantages, this monitoring scheme has some limitations. Firstly, its applicability is restricted to failure modes of the gearbox where degradation accumulates over time and the trend of anomalies can be observed. Secondly, not all failure modes follow the bathtub curve before failure, which means that the linear progression observed in this case might not be the same for all failure cases.

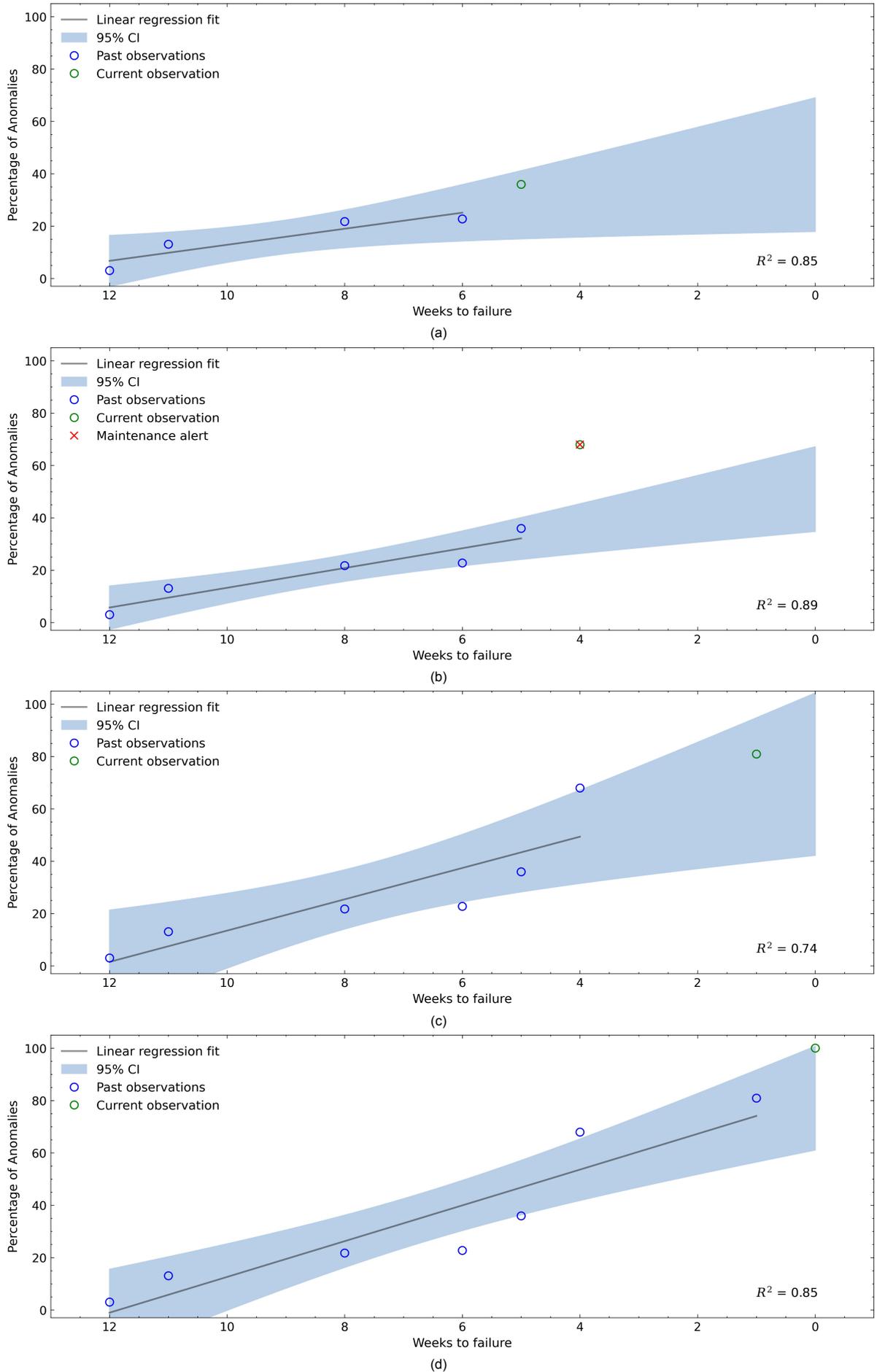


Figure 6.9: Real-time monitoring scheme implemented for monitored weeks (a) 5 weeks to failure (b) 4 weeks to failure (Maintenance alert set off as the percentage of anomalies exceeded the 95% CI bounds) (c) 1 week to failure and (d) 0 weeks to failure (day of failure)

6.5 Sensitivity study

One of the key features of the data used in this study is its high-resolution. Indeed, the CART2 dataset utilized for this research comprises SCADA data sampled at 100 Hz frequency, rather than averaged values over a 10-min period, as is the current industry practice. This provided a unique opportunity to investigate the potential of using high-frequency SCADA data for the purpose of condition monitoring of wind turbines, specifically wind turbine gearbox. When using high-frequency SCADA data, one can evidently think that there would be an effect of noise in such data, which is usually smoothed by the averaging process over conventionally used 10-min periods. However, such averaging effect would also result in a loss of information about the dynamic conditions that the wind turbine is subjected to. In the context of wind turbine condition monitoring, such loss of information might lead to false / missed alarms, therefore there is a clear need to understand the variation in information specific to the gearbox fault which is missed due to the averaging effect. To this end, a sensitivity study with varying SCADA sampling rates was carried out.

In order to conduct this sensitivity study, the developed framework for gearbox prognostics, as discussed in Chapter 4 is implemented for varying sampling rates of SCADA data. As stated in Sub-section 5.1.3, the framework was developed and implemented using data sampled at 1 Hz (one sample per second, i.e. sampling period of 1 s). The sensitivity study was then carried out for both higher and lower sampling periods. The different sampling rates considered in this study are presented in Table 6.2 with the smallest sampling period of 0.01 s (sampling frequency of 100 Hz) and the largest of 10 min. The number of data samples in healthy and faulty instances for each respective sampling period is also shown. The table shows that the number of data samples varies drastically with different sampling periods: approximately 17,000 times fewer observations for low-resolution 10-min data when compared to the 100 Hz dataset. Such a range of sampling periods provides a good basis for the analysis being done. Nevertheless, it is worth mentioning that the number of samples for the high sampling periods of 1 min, 5 min, and 10 min is quite low and this needs to be considered during the analysis. This limitation stems from the CART2 data set itself. The turbine was only operated in periods when testing and research was conducted, resulting in a low number of operational hours and a small amount of data samples when aggregated for higher sampling periods.

The study focuses on two main objectives: firstly to investigate the variation in the performance of the ANN model with varying data sampling periods and secondly - to assess the trend in the percentage of anomalies detected by the OC-SVM model in the operational period before failure. To do so, the healthy and faulty data sets for each sampling period, based on their definitions mentioned in Sub-section 5.1.2, are firstly filtered using all steps described in Sub-section 4.1.1. The healthy data is then split into three sets used for the ANN model development namely, training, validation and testing using the 70-15-15 split ratio for each development phase respectively. The ANN model performance is then evaluated by analysing R^2 and $RMSE$ for each phase of model development corresponding to each sampling period. Afterwards, the error features as discussed in Section 5.3 are computed for the healthy dataset on which the OC-SVM model is trained. This model is then used to identify anomalies in the data from the faulty dataset and its progression for weeks to failure is obtained. It is anticipated that the percentage of anomalies detected for each operational week before the gearbox failure would show a loss of information of its degradation as the sampling period increases, i.e. the averaging effect negatively impacts the failure prediction time.

ANN-based NBM model performance for different sampling rates is computed and analysed. To evaluate the accuracy of the predictions made by the ANN model, two metrics R^2 and $RMSE$ for different training datasets (from healthy data) with corresponding different time resolutions were computed. Figure 6.10 shows these metrics for each phase of model development calculated with different sampling periods. Both metrics in Figure 6.10(a) and (b), show similar but opposite trends as the sampling period increases, with R^2 values showing a decreasing trend and $RMSE$ showing an increasing trend. A good ANN model performance is characterised by high R^2 values (close to 1) and low $RMSE$ values, which can be observed for small sampling periods of 0.01 and 0.02s. For sampling periods between 1 to 30 s, the R^2 values range between 0.9 - 0.95 with negligible variation below 20 s followed by a slight decrease for 25 and 30s, whereas, there is a significant increase in $RMSE$ values as the sampling

Table 6.2: Summary of the number of data samples in healthy and faulty data sets for different sampling periods

Sampling period (s)	Sampling frequency (Hz)	Healthy data	Faulty data
0.01	100	1,839,000	4,603,970
0.02	50	919,500	2,302,000
1	1	18,400	64,770
5	0.2	4,050	9,810
10	0.1	2,080	4,940
15	0.07	1,450	3,210
20	0.05	1,031	2,430
25	0.04	850	2,010
30	0.03	730	1,620
60 [1 min]	0.017	370	841
300 [5 min]	3.3×10^{-3}	234	203
600 [10 min]	1.7×10^{-3}	108	115

period increases from 1 s to 10 s and hardly varies between sampling periods of 10 to 30 s. This is followed by an increasing trend for data sampled at intervals of 60 to 600 s. Similarly, a significant decrease in R^2 values for these sampling intervals can be seen. When analysed from the vantage point of 1 s sampling period, with which the framework was initially developed, there is no significant improvement in model performance when using data with lower sampling periods (0.01 and 0.02 s). In fact, this poses a disadvantage while utilising SCADA data sampled at such high frequencies as storing and operating large datasets would require extensive resources and computational power with no major improvements in the ANN model performance. On the other hand, the ANN still shows good performance for both metrics for data aggregated over periods of 5 - 30 s, which demonstrates the ability of the neural network to learn valuable information from the data even with approximately 25 times fewer data samples when compared with 1 s sampled data.

Another interesting observation that can be drawn from Figure 6.10 is the variation in the metrics calculated for each model development phase. The R^2 and $RMSE$ values for training, validation and testing phases barely change for sampling periods less than 30 s, however, there are significant differences observed when looked at sampling periods of 60 s and above. The metrics for the model training phase being considerably higher (in case of R^2) and low (in case of $RMSE$) than the other two phases is an indication of model overfitting to the training data and its inability to generalize its learning to the testing dataset. This is due to the low number of data samples available for model training within these sampling periods. Furthermore, to understand the distribution of error (Actual - Predicted), the error standard deviation is calculated and presented in Figure 6.11. The observed trend aligns with the ones observed previously with R^2 and $RMSE$, the lowest values for 0.01 s sampled data and the highest for 10-min sampled data. The general increasing trend is marked by a significant change from 30 to 60 s. A high value for standard deviation means that the prediction error is spread out along a wide range of values resulting from higher deviations in the model prediction when compared with the actual values. Once the ANN model is trained for all sampling periods, the error metrics as described in Table 5.7

are calculated and used to train the OC-SVM model to understand the boundary between healthy and anomalous operation.

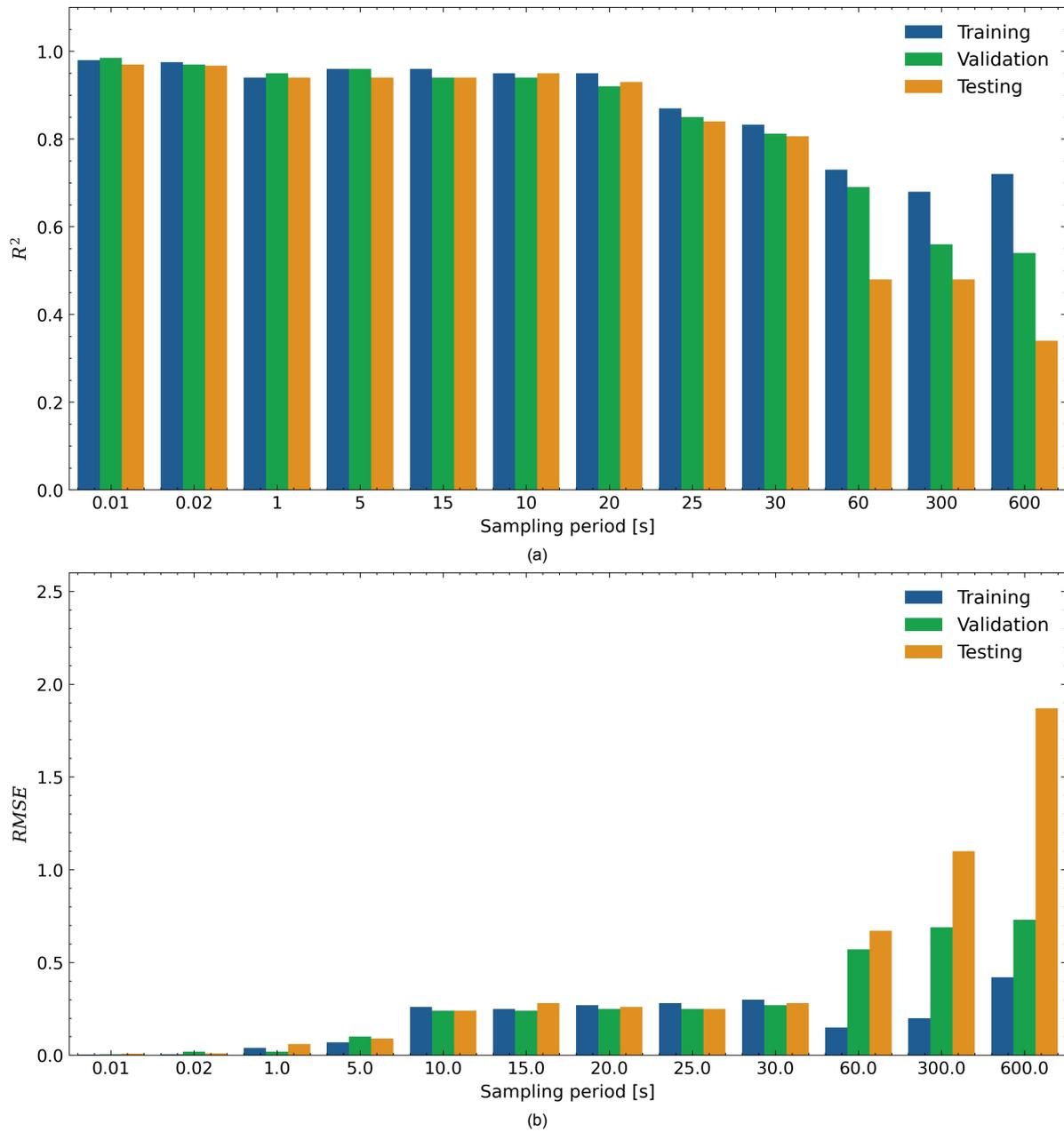


Figure 6.10: ANN model performance: (a) R^2 and (b) RMSE for all model development phases for different SCADA data sampling rates

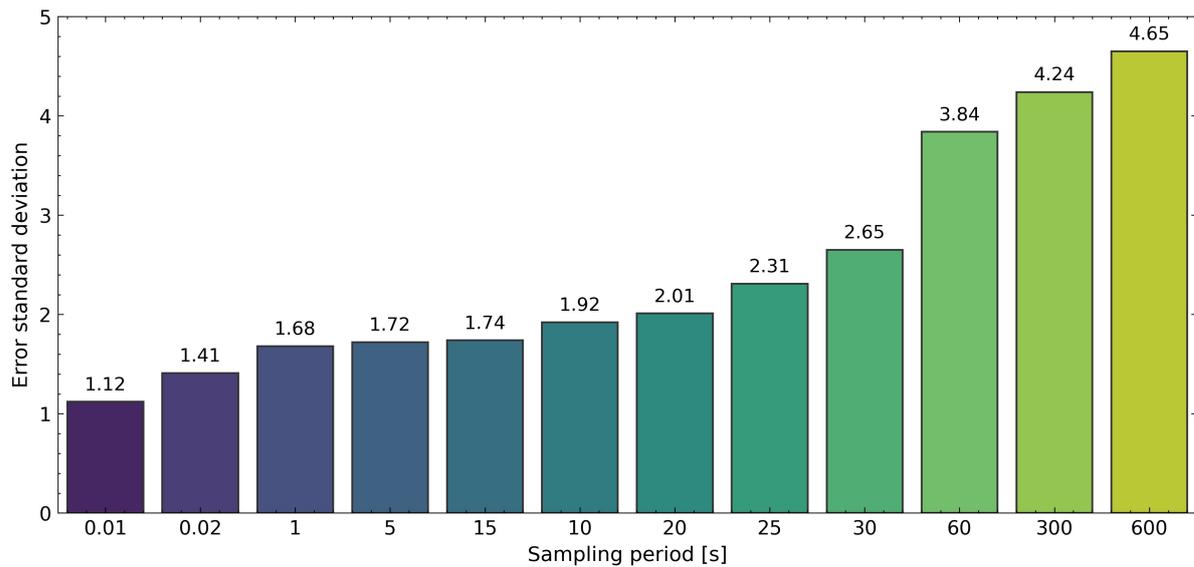


Figure 6.11: ANN model performance: Error standard deviation for different SCADA data sampling periods

Once both the models, ANN-based NBM and OC-SVM were trained with healthy data corresponding to different sampling periods, the percentage of anomalies detected using the faulty dataset was evaluated. Figure 6.12 shows the heatmap for the percentage of anomalies that were detected by OC-SVM for each week before failure corresponding to different sampling periods. As we move from left to right, the percentage of anomalies recorded for each week before failure with different sampling periods can be observed and the trend can be visualised as the colour gradient becomes darker. It can be seen that for all of the different sampling periods, there is an increasing trend in the percentage of anomalies detected as the turbine gearbox approaches failure. The percentage of anomalies progress to 100% for the week of failure for all sampling periods except for high sampling intervals of 60 to 600 s. In fact, the percentage of anomalies detected when using data with these high sampling periods are extremely low for any week before failure. This is the consequence of several factors such as:

- Poor ANN model performance due to a low number of data samples which leads to model overfitting and lack of generalization.
- High range of prediction error values, which could lead the OC-SVM model to misclassify the anomalous operation of the gearbox as healthy.
- Loss of information about the condition of the gearbox because of the data averaging effects.

In contrast, the percentage of anomalies observed when using data sampled at low periods of 0.01 and 0.02 is quite high even 12 weeks before the failure. While this can be accredited to the fact that there is more information available owing to the sampling frequency of the data, knowing that the high-frequency data often entails high noise, such high percentages could also be attributed to the outliers / noise in the data being misclassified as anomalies. In the latter case, this might even lead to false / missed alarms. Lastly, even with lower sampling periods up to 30 s it can be observed that the averaging effect results in a loss of information depicted by the decrease in the percentage of anomalies when the sampling period increases for any specific week before failure. Nevertheless, the data aggregated for 30 s interval still shows the sudden increase in the percentage of anomalies progressing from 5 to 4 weeks before failure, as was seen with the 1Hz data. In fact, the real-time monitoring scheme detects the impending failure (the observed percentage of anomalies for 4 weeks before failure were above the 95% CI bound) four weeks before it actually happened, setting off the alarm for maintenance. The results of the monitoring scheme when implemented with SCADA data sampled at 30 s interval is shown in Appendix B. This means that despite the averaging effect, the data still retains information about the gearbox fault that can be extracted efficiently through the framework and failure can be predicted a month in advance.



Figure 6.12: Heatmap for the percentage of anomalies detected by OC-SVM for different SCADA sampling rates for each week before failure

To conclude, the sensitivity analysis has shown that the ANN model performance suffers as the number of data samples available for training decreases. Poor performance was observed when the sampling period of the dataset used is aggregated over periods of 60 to 600 s. This is due to the overfitting of the model to the training data because of the low number of samples within the dataset. ANN predicts rather well when the sampling period is between 1 to 30 s and going for a higher sampling period (0.01 and 0.02 s) does not provide a good trade-off between the model performance and required resources. As for the percentage of anomalies detected by the OC-SVM, all sampling periods showed a general increasing trend as the gearbox approached failure. While higher frequency data carries more information about the condition of the gearbox, the larger percentage of anomalies observed could be a consequence of outliers within the high-resolution data, which are misclassified as anomalies by the OC-SVM. This could result in false / missed alarms. On the other hand, even considering the loss of information due to the averaging effect, the data sampled at 30 s contained information about the gearbox failure and was adequate to predict the failure a month before its occurrence. This means that SCADA data sampled up to 30 s intervals can be used for the purpose of condition monitoring of wind turbine. Nevertheless, no definitive statements could be made for any sampling interval above 30 s, owing to the specifics of the dataset (i.e. low number of operational hours for the wind turbine) and, consequently, the low number of samples when averaged over 1 to 10 min.

7

Conclusion & Recommendations

Wind energy is an important driver in the energy transition with an ongoing increase in installed wind power capacities across the globe, especially for offshore wind. As the dependence on wind energy increases, the question about its reliability becomes more concerning. Failure of WT components can lead to decreased availability of the turbine and in the case of offshore, would lead to extensive O&M costs. To this end, condition monitoring of wind turbines can be employed to implement an intelligent maintenance strategy so that necessary actions can be planned and downtime of WT is minimised. CMS data systems used specifically to monitor the condition of wind turbine components employ additional sensors and requires substantial investments and expert engineers to interpret the information within the data. On the other hand, almost all commercial wind turbines are installed with SCADA data systems which are mostly used to monitor the performance of wind turbines. Although these systems generate rich historical data and present themselves as a low-cost and reliable data source, their application for condition monitoring is still at an early stage. Hence, the problem statement of this research stressed the need for use of SCADA data for condition monitoring of wind turbines. Of all the different WT sub-assemblies, the gearbox is identified as one of the most critical components, concerning the downtime associated with its failure.

The main objectives of this project were: (1) to develop a framework for detecting gearbox failures using SCADA data that would provide early detection of fault ensuring a reasonable time period for scheduling and executing required maintenance activities and (2) to perform a sensitivity study with varying sampling rates of the SCADA data to determine an optimal sampling frequency which could be used for condition monitoring purposes.

The framework developed in this project explored the implementation of a data-driven methodology based on the NBM approach. The idea behind the approach was to emulate the normal behaviour of the gearbox and analyse the real operation deviations from this behaviour to understand the inception of a fault. To realise this gearbox prognostics framework, a combination of two ML models was used: an artificial neural network (ANN)-based NBM and a one-class support vector machine (OC-SVM). The training, validation and testing of the ML models was carried out using SCADA data collected from the Control Advanced Research Turbine (CART2) located in NREL's Flatirons campus, Colorado, USA, made available by the National Renewable Energy Laboratory (NREL). The dataset contained information of a gearbox failure that took place due to gear-teeth misalignment in the planetary gearbox and comprised of SCADA data sampled at 100 Hz, a higher frequency compared to the industry standard of 10 min averages.

A 3-layer feed-forward ANN was trained using healthy operational data of the turbine so that it could then be used to mimic the normal behaviour of the gearbox. ANN was able to understand the complex non-linear relations between the input and output features. The model comprised of five input features, namely wind speed, power output, high-speed shaft rotational speed, nacelle and ambient temperature, which were utilised to predict one output / target feature - gearbox lubrication oil temperature. The neural network was trained using data collected between Jan - Oct 2008 (initial 10 month period) i.e. when it was known that the turbine was operating in healthy condition and was implemented utilizing data collected 4 months before failure (faulty data comprising of gearbox failure). During the training phase, an additional data pre-processing approach - SMOGN - was implemented to improve the model's generalization capabilities to predict the temperature well in all operational power regions of the turbine. The model performance in all the phases of model development - training, validation

and testing - was evaluated using metrics such as root mean squared error (RMSE) and coefficient of determination (R^2). With the RMSE values for each development phase being lower than 0.1 and R^2 values close to 0.95, the model performance was deemed satisfactory.

In order to detect the anomalous operation of the turbine before failure, an OC-SVM model was used. In order to understand the unique error distribution between the actual and ANN predicted values, four different residual error features - maximum error, minimum error, $RMSE$ and error distribution - were computed. The model was trained on the healthy dataset and was able to learn the complex decision boundary between the normal and anomalous operation of the turbine and was utilised to identify anomalies in the faulty data for weeks leading up to failure. For each week of operation before failure, the percentage of anomalies was calculated. The results showed a general increasing trend in the anomaly rates as the turbine approached failure, with the first significant increase 4 weeks before failure, leading up to 100% for the week of failure. Moreover, the results from the SVM model were compared with standard approaches to evaluate residual errors based on thresholds set by the RMSE and standard deviation values observed during the turbine's healthy state. The comparative analysis demonstrated not only the robustness of the SVM model in identifying anomalies related to the turbine fault but also showed that simple statistical thresholds based solely on one metric could lead to a large number of unwanted / false alarms.

Furthermore, a real-time monitoring scheme was developed to identify the first point of inception of failure in the gearbox and set off a maintenance alarm through analysis of the progression of anomalies observed in time before failure. The monitoring scheme employed a simple linear regression model and bootstrapping to estimate 95% CI threshold bounds for the percentage of anomalies observed for the week in monitoring. These CI bounds were estimated to account for the uncertainties in the ML model performance and outliers in the data. The monitoring scheme demonstrated good results in identifying the first clear signs of incipient fault in the gearbox and predicted a functional failure of the gearbox a month in advance, providing enough lead time to plan and execute maintenance activities. One of the key features of this scheme was that it could update its parameter iteratively by taking into account the percentage of anomalies recorded with each progressing week and set adaptive and robust thresholds. Another important advantage of using such an online monitoring method is that it could be used for turbines with no available run-to-failure data.

One of the most important aspects of the data used in this study was its high-resolution and to understand an optimal SCADA data sampling frequency, a sensitivity study was carried out. This included training the ANN model with 11 training sets corresponding to different sampling periods ranging from low values of 0.01 s (100 Hz) to data aggregated over 600 s (10 min) periods. The analysis for R^2 and $RMSE$ metrics showed that the ANN model performance suffers as the sampling period increases due to the decrease in the number of samples available for training. In the case of an extremely low number of training samples, the ANN model overfits to the training data. It was also found out that SCADA sampled at high frequencies capture more information about the condition of the machine and the averaging effect negatively affects the percentage of anomalies observed in weeks before the turbine failure. Nevertheless, with very high sampling rates (for 0.01 s and 0.02 s), the issue of noise in the data becomes prominent, which might be misclassified as anomalies by the OC-SVM model. Despite the effects of averaging in data sampled at lower frequencies, it was found that with data sampled over 30 s interval, the first clear signs of incipient fault could be observed and a fair prediction period of 1 month could be made. Although the objective of the study was to determine an optimal SCADA data sampling frequency for the purpose of condition monitoring, the results were not exhaustive as the number of data samples available for training and testing the framework was too low for sampling periods over 30 s. Additionally, the sensitivity study demonstrated the robustness of the developed framework in effectively predicting the gearbox failure a month in advance.

Albeit the limitations posed by the dataset (time discontinuity and outliers caused due to testing new control algorithms), the framework developed for the prognostics of WT gearbox failure was effective in identifying fault characteristics a month before the actual failure occurred. However, if the data set comprised of continuous periods of WT operation, there could have been the possibility to detect the failure even before, hence, improving the lead time to conduct maintenance activities prior to the oc-

currence of a catastrophic failure and shutdown of the turbine.

To further realise the potential of the developed framework and taking into consideration, the shortcomings of this study, the recommendations for future work are compiled as follows:

- Scalability of the framework. The framework implemented in this project was developed and testing utilizing data from a wind turbine with a rated power of 650 kW. The scalability of this framework should be assessed with data from higher power rated turbines.
- Analysis with variants of the ANN model. This project employed a simple ANN architecture to understand the complex and non-linear relationship between the input and output features, however, there are available variants of ANN such as recurrent neural networks (RNN), Non-linear autoregressive model with exogenous inputs (NARX) which could provide better model performance, however a trade-off study between increased model complexity and performance improvements must be done.
- Investigation of the hybrid modelling approach. This project investigated data-driven methods using SCADA data for the prognosis of the gearbox. Although the ML models used showed good results in predicting the gearbox failure a month in advance, a clear diagnosis for the failure mode could not be made. This could be further investigated by developing a hybrid model which combines the advantages of machine learning models with physics-based models to identify the root cause of the gearbox failure.
- Inclusion of more sensor signals indicative of the health condition of the turbine. In this study, an understanding of gearbox physics and domain knowledge was used to determine the gearbox lubrication oil temperature to be the most sensitive to gearbox failure. However, a more detailed study using a variety of signals should be done to include more health indicators for the gearbox.
- Sensitivity study with a larger dataset. The sensitivity study with different SCADA data sampling rates carried out in this project could only draw conclusions with data averaged over a 30 s period due to the unique characteristics of this dataset. A fair comparative study should be carried out such that the total number of data samples for each sampling period is sufficient for ANN to perform well and not overfit to the training dataset. Such a study would diminish the effect of ML model performance due to varying number of data samples and the loss of information due to the averaging effect can be studied further, resulting in an optimal SCADA data frequency which could then be implemented in the wind industry.

A

Neural network optimization

In this Appendix, the explanation for the optimization algorithm for the ANN is discussed and derived mathematically [64]. The optimization of ANN involves two passes or processes: forward propagation (feed-forward pass) and backward propagation (backpropagation).

Forward propagation

Let there be a fully connected feed-forward neural network that has L layers and a pre-defined number of neurons (can be any number). The activations of the neurons in a given layer l are stored in a vector $a^{(l)}$, where the superscript denotes the layer. The connections (weights) from neurons in the previous layer $(l - 1)$ to the layer l are stored in weight matrix $W^{(l)}$ and the biases for that layer are stored in the column-vector b^l . Furthermore, $z^{(l)}$ is a vector that represents the input sum of all neurons present in a layer l . Equations A.1 and A.2 show a simple forward pass [60].

$$z^{(l)} = W^{(l-1)} * a^{(l-1)} + b^{(l-1)} \quad (\text{A.1})$$

$$a^{(l)} = f(z^{(l)}) \quad (\text{A.2})$$

Figure A.1 shows the whole neural network and mapping of neurons from input vector x , to the output activation vector $a^{(L)}$. The connections leading into a specific neuron is shown in two colours in two layers.

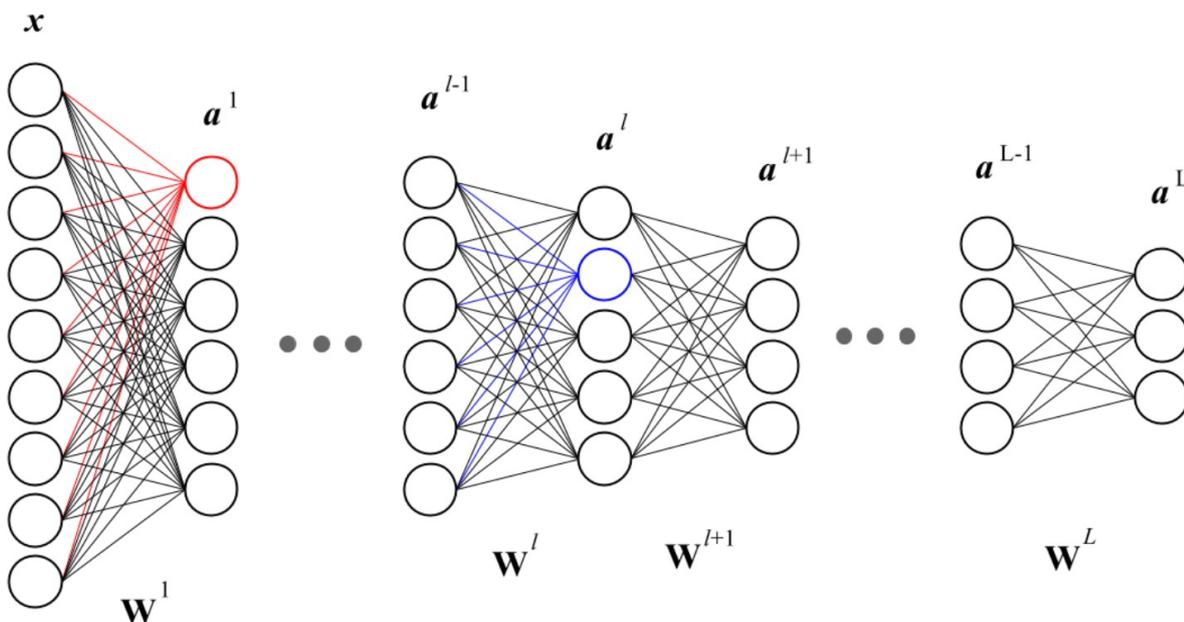


Figure A.1: Entire neural network visualised

To better understand the notations used in the explanation ahead, the network computation is summarised in one mathematical expression. The Equation ref shows the formula for calculation n^{th} element of the output vector in the final layer.

$$a_n^{(L)} = \left[f \left(\sum_m w_{nm}^{(L)} \left[\dots \left[f \left(\sum_j w_{kj}^{(2)} \left[f \left(\sum_i w_{ji}^{(1)} x_i + b_j^{(1)} \right) \right] + b_k^{(2)} \right) \right] \dots \right] + b_n^{(L)} \right) \right]_n \quad (\text{A.3})$$

The notation $w_{uv}^{(l)}$ denotes the connection from v^{th} neuron in layer $(l-1)$ to u^{th} neuron in layer (l) and $b_u^{(l)}$ is the bias of the u^{th} neuron in layer (l) .

The error observed for a neuron i in layer l is denoted by $\delta_i^{(l)}$ and is defined as how much the total error changes when the input sum of the neuron is changed. This is computed by taking the derivative of the cost function (C) with respect to input sum (z) for each neuron and is expressed by the following equation:

$$\delta_i^{(l)} = \frac{\partial C}{\partial z_i^{(l)}} \quad (\text{A.4})$$

Backward propagation

After the feed-forward pass, the error for each neuron is calculated. The next step is optimizing the weights matrix to minimize loss, by the use of a backpropagation algorithm. The backpropagation algorithm works by computing the gradient $\nabla_{ij}^{(l)}$ of the loss function (or objective function C) with respect to each weight $w_{ij}^{(l)}$, propagating from the final layer, iterating backwards, one layer at a time. This is advantageous in avoiding redundant calculations of immediate terms in the chain rule. The derivation utilizes a conventional gradient descent technique to simplify the explanation. Furthermore, a classic activation function is assumed.

The starting point is the derivative and the gradient is defined as:

$$\nabla_{ij}^{(l)} = \frac{\partial C}{\partial w_{ij}^{(l)}} \quad (\text{A.5})$$

This Equation A.5 cannot be solved directly and therefore, need to be modified using two methods to derive an equation that could be computed by the neural network model. This results in the following equation:

$$\nabla_{ij}^{(l)} = W^{(l+1)} \delta^{(l+1)} \cdot \left(a_i^{(l)} (1 - a_i^{(l)}) \right) * a_j^{(l-1)} \quad (\text{A.6})$$

The first method is based on the notion that the gradient can be expressed using $\delta_i^{(l)}$:

$$\nabla_{ij}^{(l)} = \delta_i^{(l)} * a_j^{(l-1)} \quad (\text{A.7})$$

The backpropagation algorithm applies chain rule to communicate the updates and pass information to other nodes. This forms the second method which is based on the relation between adjacent error: $\delta_i^{(l)}$ and $\delta_i^{(l+1)}$. It is expressed as:

$$\delta_i^{(l)} = W^{(l+1)T} \delta^{(l+1)} \cdot \left(a_i^{(l)} (1 - a_i^{(l)}) \right) \quad (\text{A.8})$$

Both these methods result in a general form of backpropagation and can be written as:

$$\nabla_{ij}^{(l)} = W^{(l+1)T} \delta^{(l+1)} \cdot \left(a_i^{(l)} (1 - a_i^{(l)}) \right) * a_j^{(l-1)} \quad (\text{A.9})$$

Demonstration of Equation A.7

Previously, it was defined that:

$$\nabla_{ij}^{(l)} = \frac{\partial C}{\partial w_{ij}^{(l)}} \quad (\text{A.10})$$

Following the chain rule for higher dimensions enables re-writing it to the following expression:

$$\nabla_{ij}^{(l)} = \sum_k \frac{\partial C}{\partial z_k^{(l)}} * \frac{\partial z_k^{(l)}}{\partial w_{ij}^{(l)}} \quad (\text{A.11})$$

On the other hand:

$$z_k^{(l)} = \sum_m w_{km}^{(l)} * a_m^{(l-1)} \quad (\text{A.12})$$

The following expression can be given as:

$$\frac{\partial z_k^{(l)}}{\partial w_{ij}^{(l)}} = \frac{\partial}{\partial w_{ij}^{(l)}} \sum_m w_{km}^{(l)} * a_m^{(l-1)} \quad (\text{A.13})$$

By employing the linearity rule of differentiation [(u + v)' = u' + v'], Equation A.13 can be written as:

$$\frac{\partial z_k^{(l)}}{\partial w_{ij}^{(l)}} = \sum_m \frac{\partial w_{km}^{(l)}}{\partial w_{ij}^{(l)}} * a_m^{(l-1)} \quad (\text{A.14})$$

$$\text{if } k, m \neq i, j, \quad \frac{\partial w_{km}^{(l)}}{\partial w_{ij}^{(l)}} * a_m^{(l-1)} = 0 \quad (\text{A.15})$$

$$\text{if } k, m = i, j, \quad \frac{\partial w_{ij}^{(l)}}{\partial w_{ij}^{(l)}} * a_j^{(l-1)} = a_j^{(l-1)} \quad (\text{A.16})$$

Then for k = i:

$$\frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}} = \frac{\partial w_{im}^{(l)}}{\partial w_{ij}^{(l)}} * a_j^{(l-1)} + \sum_{m \neq j} \frac{\partial w_{im}^{(l)}}{\partial w_{ij}^{(l)}} * a_j^{(l-1)} = a_j^{(l-1)} + 0 \quad (\text{A.17})$$

And, finally:

$$\frac{\partial z_i^{(l)}}{\partial w_{ij}^{(l)}} = a_j^{(l-1)} \quad (\text{A.18})$$

Thus, the first expression of $\nabla_{ij}^{(l)}$ results in:

$$\nabla_{ij}^{(l)} = \frac{\partial C}{\partial z_i^{(l)}} * \frac{z_i^{(l)}}{\partial w_{ij}^{(l)}} \quad (\text{A.19})$$

Equivalent to:

$$\nabla_{ij}^{(l)} = \frac{\partial C}{\partial z_i^{(l)}} * a_j^{(l-1)} \quad (\text{A.20})$$

Using Equation A.6 and A.17:

$$\nabla_{ij}^{(l)} = \delta_i^{(l)} * a_j^{(l-1)} \quad (\text{A.21})$$

B

Real-time monitoring scheme

In this Appendix, the developed real-time monitoring scheme is employed with SCADA data aggregated over a 30 s time period. The purpose of implementing such a scheme within the gearbox prognostics framework was to track the progression of anomalies in time and correctly identify the first indication of failure inception in the gearbox. It is based on a simple linear regression model combined with 95% CI bounds estimated using the bootstrapping method. Section 6.4, discussed in detail how such a scheme could be implemented in an online setting. The results demonstrated its efficacy in identifying the first signs of failure and alerting a maintenance alarm 4 weeks or 1 month before the gearbox failure occurred. This analysis was carried out using SCADA data sampled at a high frequency of 1 Hz, rather than 10-min averaged values, which is the current industry practice.

To further understand the negative effects of averaging in SCADA data and its impact on condition monitoring of the gearbox, a comprehensive sensitivity study was carried out, results of which are discussed in detail in Section 6.5. SCADA data sampled at 30 s showed a similar trend for the percentage of anomalies before failure as was observed with 1 s (1 Hz) data, although there was a decrease in anomalies observed in a certain week. However, to verify the applicability of SCADA data aggregated over a 30 s period for gearbox condition monitoring, the percentage of anomalies recorded before failure were tracked using the real-time monitoring scheme and the first inception of failure was identified. This is shown in Figure B.1. In Figure B.1(a), anomalies recorded for all days before 5 weeks to failure (monitored week) are used to compute the linear fit and the 95% CI bounds. In Figure B.1(b), the anomalies observed for 4 weeks to failure exceed the upper CI bound and a maintenance alert is set off. Similar to the results obtained with SCADA data sampled at 1 s (1 Hz), the first incipient signs of gearbox failure are evident 4 weeks before the actual failure. When Figure B.1(c) and (d) are compared with Figure 6.9(c) and (d), it can be seen that when the percentage of anomalies observed 4 weeks before failure is added for regression fit, the prediction bounds at 0 weeks to failure (day of failure) do not indicate a functional failure (100% anomalies). Nevertheless, in a real-world scenario, no data would be recorded after the maintenance alert is set off i.e. 4 weeks to failure. Therefore, it can be concluded that even with SCADA data sampled over a 30 s period, failure of the gearbox can be anticipated a month before failure.

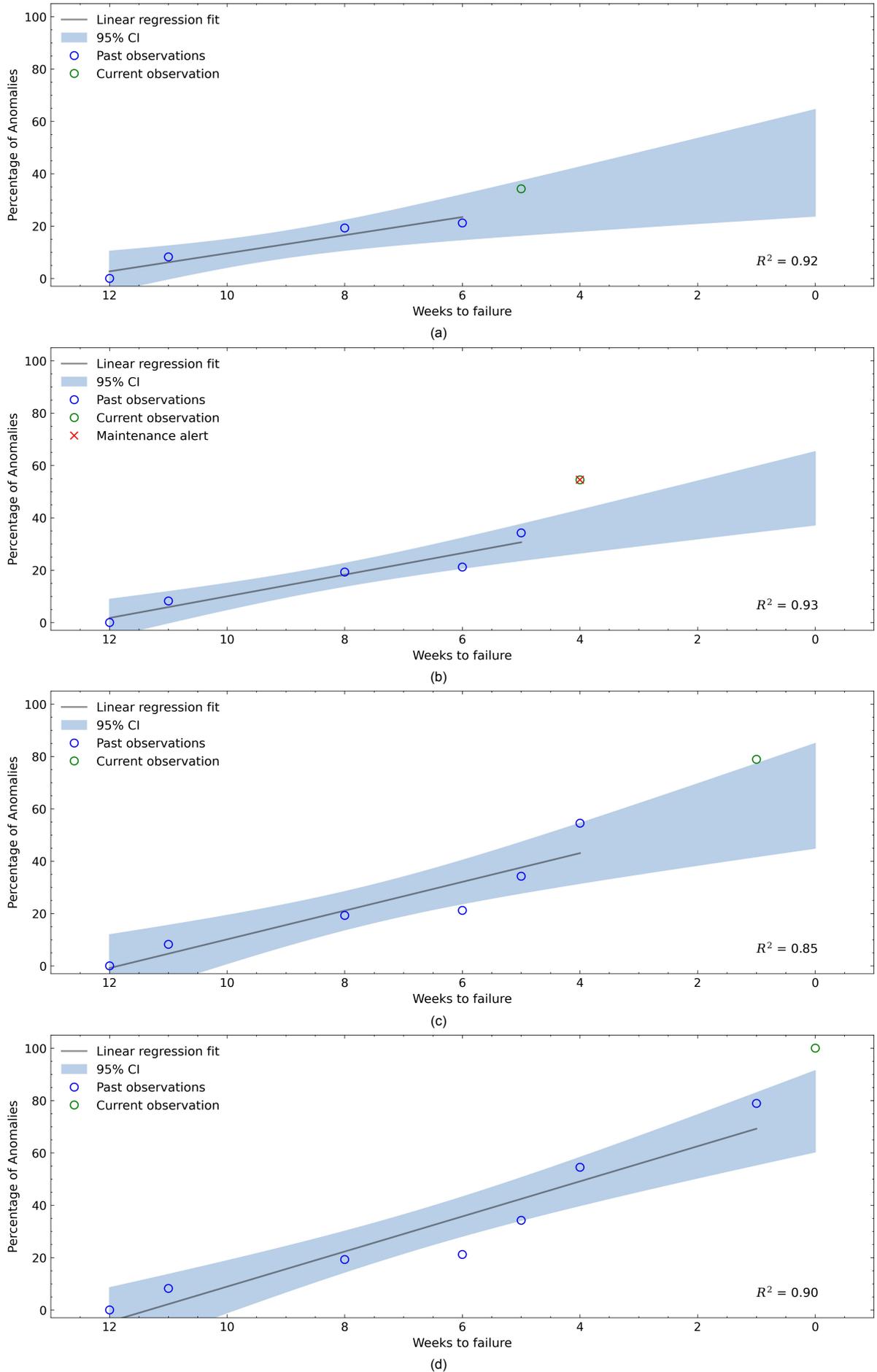


Figure B.1: Real-time monitoring scheme implemented with SCADA data aggregated over 30 s for monitored weeks (a) 5 weeks to failure (b) 4 weeks to failure (Maintenance alert set off as percentage of anomalies exceed the 95% CI bounds) (c) 1 week to failure and (d) 0 weeks to failure (day of failure)

Bibliography

- [1] International Renewable Energy Association. Future of wind: Deployment, investment, technology, grid integration and socio-economic aspects, 2019.
- [2] Wind Europe. It's official: The eu commission wants 30 gw a year of new wind up to 2030. <https://windeurope.org/newsroom/press-releases>, July 2021.
- [3] Wind Europe. Offshore wind in europe – key trends and statistics 2020. <https://windeurope.org/intelligence-platform/product/offshore-wind-in-europe-key-trends-and-statistics-2020/>.
- [4] Maik Dennis Reder, Elena Gonzalez, and Julio J Melero. Wind turbine failures-tackling current problems in failure data analysis. In *Journal of Physics: Conference Series*, volume 753, page 072027. IOP Publishing, 2016.
- [5] Kevin Leahy, Colm Gallagher, Ken Bruton, Peter O'Donovan, and Dominic TJ O'Sullivan. Automatically identifying and predicting unplanned wind turbine stoppages using scada and alarms system data: Case study and results. In *Journal of Physics: Conference Series*, volume 926, page 012011. IOP Publishing, 2017.
- [6] Pierre Tchakoua, René Wamkeue, Mohand Ouhrouche, Fouad Slaoui-Hasnaoui, Tommy Andy Tameghe, and Gabriel Ekemb. Wind turbine condition monitoring: State-of-the-art review, new trends, and future challenges. *Energies*, 7(4):2595–2630, 2014.
- [7] Wenxian Yang, Richard Court, and Jiesheng Jiang. Wind turbine condition monitoring by the approach of scada data analysis. *Renewable Energy*, 53:365–376, 2013.
- [8] DNV. Online scada-based condition monitoring. <https://www.dnv.com/services/online-scada-based-condition-monitoring>.
- [9] Adrian Stetco, Fateme Dinmohammadi, Xingyu Zhao, Valentin Robu, David Flynn, Mike Barnes, John Keane, and Goran Nenadic. Machine learning methods for wind turbine condition monitoring: A review. *Renewable energy*, 133:620–635, 2019.
- [10] Sunil Kr Jha, Jasmin Bilalovic, Anju Jha, Nilesh Patel, and Han Zhang. Renewable energy: Present research and future scope of artificial intelligence. *Renewable and Sustainable Energy Reviews*, 77:297–317, 2017.
- [11] Michele Lucente. Condition monitoring system in wind turbine gearbox. *Master's thesis, KTH in cooperation with NTNU, Stockholm*, 2008.
- [12] Wenbin Dong, Yihan Xing, and Torgeir Moan. Time domain modeling and analysis of dynamic gear contact force in a wind turbine gearbox with respect to fatigue assessment. *Energies*, 5(11):4350–4371, 2012.
- [13] Walt Musial, Sandy Butterfield, and Brian McNiff. Improving wind turbine gearbox reliability. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2007.
- [14] Yanhui Feng, Yingning Qiu, Christopher J Crabtree, Hui Long, and Peter J Tavner. Monitoring wind turbine gearboxes. *Wind Energy*, 16(5):728–740, 2013.
- [15] National Renewable Energy Laboratory. Gearbox reliability database. <https://grd.nrel.gov/#/stats/>, 2017.
- [16] Thomas Bruce. *Analysis of the premature failure of wind turbine gearbox bearings*. PhD thesis, University of Sheffield, 2016.

- [17] Bin Lu, Yaoyu Li, Xin Wu, and Zhongzhou Yang. A review of recent advances in wind turbine condition monitoring and fault diagnosis. In *2009 IEEE power electronics and machines in wind applications*, pages 1–7. IEEE, 2009.
- [18] Amir Rasekhi Nejad, Zhen Gao, and Torgeir Moan. Fatigue reliability-based inspection and maintenance planning of gearbox components in wind turbine drivetrains. *Energy Procedia*, 53:248–257, 2014.
- [19] M Carolin Mabel, R Edwin Raj, and E Fernandez. Analysis on reliability aspects of wind power. *Renewable and Sustainable Energy Reviews*, 15(2):1210–1216, 2011.
- [20] Cuong Dao, Behzad Kazemtabrizi, and Christopher Crabtree. Wind turbine reliability data review and impacts on levelised cost of energy. *Wind Energy*, 22(12):1848–1871, 2019.
- [21] Balbir S Dhillon. *Engineering maintenance: a modern approach*. cRc press, 2002.
- [22] Rolf F Orsagh, Hyungdae Lee, Matthew Watson, Carl S Byington, and Jon Powers. Advanced vibration monitoring for wind turbine health management. *Impact Technologies*, 2006.
- [23] Christopher A Walford. Wind turbine reliability: understanding and minimizing wind turbine operation and maintenance costs. Technical report, Sandia National Laboratories, 2006.
- [24] Johan Ribrant. Reliability performance and maintenance-a survey of failures in wind power systems. *KTH school of Electrical Engineering*, pages 59–72, 2006.
- [25] Luis Barberá, Alejandro Guerrero, Adolfo Crespo, Vicente González-Prida, Antonio Guillén, Juan Gómez, and Antonio Sola. State of the art of maintenance applied to wind turbines. *CHEMICAL ENGINEERING*, 33, 2013.
- [26] Wenxian Yang, Peter J Tavner, Christopher J Crabtree, Yanhui Feng, and Yingning Qiu. Wind turbine condition monitoring: technical and commercial challenges. *Wind Energy*, 17(5):673–693, 2014.
- [27] S Sheng and Wenxian Yang. Wind turbine drivetrain condition monitoring-an overview (presentation). 2013.
- [28] Yanhui Feng, Yingning Qiu, Christopher J Crabtree, Hui Long, and Peter J Tavner. Use of scada and cms signals for failure detection and diagnosis of a wind turbine gearbox. In *European Wind Energy Conference and Exhibition 2011, EWEC 2011*, pages 17–19. Sheffield, 2011.
- [29] Milad Rezamand, Mojtaba Kordestani, Rupp Carriveau, David SK Ting, Marcos E Orchard, and Mehrdad Saif. Critical wind turbine components prognostics: A comprehensive review. *IEEE Transactions on Instrumentation and Measurement*, 2020.
- [30] Jorge Maldonado-Correa, Sergio Martín-Martínez, Estefanía Artigao, and Emilio Gómez-Lázaro. Using scada data for wind turbine condition monitoring: A systematic literature review. *Energies*, 13(12):3132, 2020.
- [31] Georgios Alexandros Skrimpas, Karolina Kleani, Nenad Mijatovic, Christian Walsted Sweeney, Bogi Bech Jensen, and Joachim Holboell. Detection of icing on wind turbine blades by means of vibration and power curve analysis. *Wind Energy*, 19(10):1819–1832, 2016.
- [32] E González and JJ Melero. Wind turbine fault detection by monitoring its performance using high-resolution scada data.
- [33] Tanja A Mücke, Matthias Wächter, Patrick Milan, and Joachim Peinke. Langevin power curve analysis for numerical wind energy converter models with new insights on high frequency power performance. *Wind Energy*, 18(11):1953–1971, 2015.
- [34] E. Gonzalez, B. Stephen, D. Infield, and J.J. Melero. On the use of high-frequency scada data for improved wind turbine performance monitoring. In *Journal of Physics: Conference Series*, volume 926, page 012009. IOP Publishing, 2017.

- [35] Elena Gonzalez, Bruce Stephen, David Infield, and Julio J Melero. Using high-frequency scada data for wind turbine performance monitoring: A sensitivity study. *Renewable energy*, 131:841–853, 2019.
- [36] Kyusung Kim, Girija Parthasarathy, Onder Uluyol, Wendy Foslien, Shuangwen Sheng, and Paul Fleming. Use of scada data for failure detection in wind turbines. In *Energy Sustainability*, volume 54686, pages 2071–2079, 2011.
- [37] ASAE Zaher, SDJ McArthur, DG Infield, and Y Patel. Online wind turbine fault detection through automated scada data analysis. *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, 12(6):574–593, 2009.
- [38] CJ Crabtree, Y Feng, and PJ Tavner. Detecting incipient wind turbine gearbox failure: a signal analysis method for on-line condition monitoring. In *Proceedings of European Wind Energy Conference (EWECC 2010), Warsaw, Poland*, volume 2023, page 154156, 2010.
- [39] JZ Sikorska, Melinda Hodkiewicz, and Lin Ma. Prognostic modelling options for remaining useful life estimation by industry. *Mechanical systems and signal processing*, 25(5):1803–1836, 2011.
- [40] Adrian Cubillo, Suresh Perinpanayagam, and Manuel Esperon-Miguez. A review of physics-based models in prognostics: Application to gears and bearings of rotating machinery. *Advances in Mechanical Engineering*, 8(8):1687814016664660, 2016.
- [41] Mari Cruz Garcia, Miguel A Sanz-Bobi, and Javier Del Pico. Simap: Intelligent system for predictive maintenance: Application to the health condition monitoring of a windturbine gearbox. *Computers in Industry*, 57(6):552–568, 2006.
- [42] Andrew Kusiak and Wenyan Li. The prediction and diagnosis of wind turbine faults. *Renewable energy*, 36(1):16–23, 2011.
- [43] Yingying Zhao, Dongsheng Li, Ao Dong, Jiajia Lin, Dahai Kang, and Li Shang. Fault prognosis of wind turbine generator using scada data. In *2016 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2016.
- [44] Joyjit Chatterjee and Nina Dethlefs. Scientometric review of artificial intelligence for operations & maintenance of wind turbines: The past, present and future. *Renewable and Sustainable Energy Reviews*, 144:111051, 2021.
- [45] Clement Albert Anne Magniez. Combining information flow and physics-of-failure in mechatronic products. 2007.
- [46] Christopher S Gray and Simon J Watson. Physics of failure approach to wind turbine condition based maintenance. *Wind Energy*, 13(5):395–405, 2010.
- [47] Shane Butler, Frank O'Connor, Des Farren, and John V Ringwood. A feasibility study into prognostics for the main bearing of a wind turbine. In *2012 IEEE International Conference on Control Applications*, pages 1092–1097. IEEE, 2012.
- [48] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, Mao-Yin Chen, and Dong-Hua Zhou. A wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *Mechanical Systems and Signal Processing*, 35(1-2):219–237, 2013.
- [49] William G Garlick, Roger Dixon, and Simon Watson. A model-based approach to wind turbine condition monitoring using scada data. 2009.
- [50] Yaogang Hu, Hui Li, Pingping Shi, Zhaosen Chai, Kun Wang, Xiangjie Xie, and Zhe Chen. A prediction method for the real-time remaining useful life of wind turbine bearings based on the wiener process. *Renewable energy*, 127:452–460, 2018.
- [51] Khanh Le Son, Mitra Fouladirad, Anne Barros, Eric Levrat, and Benoît Iung. Remaining useful life estimation based on stochastic deterioration models: A comparative study. *Reliability Engineering & System Safety*, 112:165–175, 2013.

- [52] Tianyi Wang, Jianbo Yu, David Siegel, and Jay Lee. A similarity-based prognostics approach for remaining useful life estimation of engineered systems. In *2008 international conference on prognostics and health management*, pages 1–6. IEEE, 2008.
- [53] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. A brief survey of text mining. In *Ldv Forum*, volume 20, pages 19–62. Citeseer, 2005.
- [54] Faris Elasha, Suliman Shanbr, Xiaochuan Li, and David Mba. Prognosis of a wind turbine gearbox bearing using supervised machine learning. *Sensors*, 19(14):3092, 2019.
- [55] M Carolin Mabel and Eugene Fernandez. Analysis of wind power generation and prediction using ann: A case study. *Renewable energy*, 33(5):986–992, 2008.
- [56] Andrew Kusiak and Anoop Verma. Analyzing bearing faults in wind turbines: A data-mining approach. *Renewable Energy*, 48:110–116, 2012.
- [57] P Bangalore, S Letzgus, D Karlsson, and M Patriksson. An artificial neural network-based condition monitoring method for wind turbines, with application to the monitoring of the gearbox. *Wind Energy*, 20(8):1421–1438, 2017.
- [58] Alan Turnbull, James Carroll, and Alasdair McDonald. Combining scada and vibration data into a single anomaly detection model to predict wind turbine component failure. *Wind Energy*, 24(3):197–211, 2021.
- [59] Judith E Dayhoff. *Neural network architectures: an introduction*. Van Nostrand Reinhold Co., 1990.
- [60] Martin T Hagan, Howard B Demuth, Mark Hudson Beale, and Orlando De Jesús. *Neural network design*. 2nd edition. *Oklahoma: Martin Hagan*, 2014.
- [61] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.
- [62] Mohammad Hindi Bataineh. *Artificial neural network for studying human performance*. Citeseer, 2012.
- [63] Dana Hughes and Nikolaus Correll. Distributed machine learning in materials that couple sensing, actuation, computation and communication. *arXiv preprint arXiv:1606.03508*, 2016.
- [64] Andrew Ng. Basics of neural network programming. <https://cs230.stanford.edu/>.
- [65] Ahmad Alwosheel, Sander van Cranenburgh, and Caspar G Chorus. Is your dataset big enough? sample size requirements when using artificial neural networks for discrete choice analysis. *Journal of choice modelling*, 28:167–182, 2018.
- [66] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [67] Richard Sutton. Two problems with back propagation and other steepest descent learning procedures for networks. In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society, 1986*, pages 823–832, 1986.
- [68] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.
- [69] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [70] Paula Branco, Luís Torgo, and Rita P Ribeiro. Smogn: a pre-processing approach for imbalanced regression. In *First international workshop on learning with imbalanced domains: Theory and applications*, pages 36–50. PMLR, 2017.

- [71] Haibo He and Eduardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.
- [72] Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250:113–141, 2013.
- [73] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.
- [74] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50, 2016.
- [75] Luís Torgo, Paula Branco, Rita P Ribeiro, and Bernhard Pfahringer. Resampling strategies for regression. *Expert Systems*, 32(3):465–476, 2015.
- [76] Luís Torgo, Rita P Ribeiro, Bernhard Pfahringer, and Paula Branco. Smote for regression. In *Portuguese conference on artificial intelligence*, pages 378–389. Springer, 2013.
- [77] Paula Branco, Rita P Ribeiro, and Luis Torgo. Ubl: an r package for utility-based learning. *arXiv preprint arXiv:1604.08079*, 2016.
- [78] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [79] Sauchi Stephen Lee. Noisy replication in skewed binary classification. *Computational statistics & data analysis*, 34(2):165–191, 2000.
- [80] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.
- [81] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, et al. Support vector method for novelty detection. In *NIPS*, volume 12, pages 582–588. Citeseer, 1999.
- [82] Kai Guo, Datong Liu, Yu Peng, and Xiyuan Peng. Data-driven anomaly detection using ocsvm with boundary optimization. In *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*, pages 244–248. IEEE, 2018.
- [83] Tian Wang, Jie Chen, Yi Zhou, and Hichem Snoussi. Online least squares one-class support vector machines-based abnormal visual event detection. *Sensors*, 13(12):17130–17155, 2013.
- [84] Conor McKinnon, James Carroll, Alasdair McDonald, Sofia Koukoura, David Infield, and Conaill Soraghan. Comparison of new anomaly detection technique for wind turbine condition monitoring using gearbox scada data. *Energies*, 13(19):5152, 2020.
- [85] Douglas C Montgomery and George C Runger. *Applied statistics and probability for engineers*. John Wiley & Sons, 2010.
- [86] Hongyao Deng and Xiuli Song. The theory and practice of linear regression. *World Transactions on Engineering and Technology Education*, XI (4), pages 382–387, 2013.
- [87] Iain Pardoe. *Applied regression modeling*. John Wiley & Sons, 2020.
- [88] Bodhipaksha Thilakarathne. Finding a better confidence interval for a single regression change-point using different bootstrap confidence interval procedures. 2012.
- [89] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.
- [90] Samaradasa Weerahandi. Generalized confidence intervals. In *Exact statistical methods for data analysis*, pages 143–168. Springer, 1995.
- [91] Jason S Haukoos and Roger J Lewis. Advanced statistics: bootstrapping confidence intervals for statistics with “difficult” distributions. *Academic emergency medicine*, 12(4):360–365, 2005.

- [92] Thomas J DiCiccio and Bradley Efron. Bootstrap confidence intervals. *Statistical science*, 11(3):189–228, 1996.
- [93] Herman J Adèr and Gideon J Mellenbergh. *Advising on research methods: selected topics 2012*. Johannes van Kessel Publishing, 2013.
- [94] Pere Marti-Puig, Alejandro Blanco-M, Juan José Cárdenas, Jordi Cusidó, and Jordi Solé-Casals. Effects of the pre-processing algorithms in fault diagnosis of wind turbines. *Environmental modelling & software*, 110:119–128, 2018.
- [95] Meik Schlechtingen, Ilmar Ferreira Santos, and Sofiane Achiche. Wind turbine condition monitoring based on scada data using normal behavior models. part 1: System description. *Applied Soft Computing*, 13(1):259–270, 2013.
- [96] Jannis Tautz-Weinert and Simon J Watson. Using scada data for wind turbine condition monitoring—a review. *IET Renewable Power Generation*, 11(4):382–394, 2016.
- [97] Tarun Kumar Gupta and Khalid Raza. Optimization of ann architecture: a review on nature-inspired techniques. *Machine learning in bio-signal analysis and diagnostic imaging*, pages 159–182, 2019.
- [98] Jun Zhou, Sanaz Roshanmanesh, Farzad Hayati, V Jantara Junior, Taoran Wang, Siavash Hajiabady, XY Li, Hector Basoalto, Hanshan Dong, and Mayorkinos Papaelias. Improving the reliability of industrial multi-mw wind turbines. *Insight-Non-Destructive Testing and Condition Monitoring*, 59(4):189–195, 2017.
- [99] Jesús María Pinar Pérez, Fausto Pedro García Márquez, Andrew Tobias, and Mayorkinos Papaelias. Wind turbine reliability analysis. *Renewable and Sustainable Energy Reviews*, 23:463–472, 2013.
- [100] Christopher Z Mooney, Christopher F Mooney, Christopher L Mooney, Robert D Duval, and Robert Duvall. *Bootstrapping: A nonparametric approach to statistical inference*. Number 95. sage, 1993.
- [101] Kathryn E Johnson and Paul A Fleming. Development, implementation, and testing of fault detection strategies on the national wind technology center’s controls advanced research turbines. *Mechatronics*, 21(4):728–736, 2011.
- [102] E Bossanyi, A Wright, and P Fleming. Controller field tests on the nrel cart2 turbine. Technical report, National Renewable Energy Lab.(NREL), Golden, CO (United States), 2010.
- [103] Lee Jay Fingersh and K Johnson. Controls advanced research turbine (cart) commissioning and baseline data collection. Technical report, National Renewable Energy Lab., Golden, CO.(US), 2002.
- [104] Mingzhe Zou and Sasa Z Djokic. A review of approaches for the detection and treatment of outliers in processing wind turbine and wind farm measurements. *Energies*, 13(16):4228, 2020.
- [105] Yingning Qiu, Yanhui Feng, Juan Sun, Wenxiu Zhang, and David Infield. Applying thermophysics for wind turbine drivetrain fault diagnosis using scada data. *IET Renewable Power Generation*, 10(5):661–668, 2016.
- [106] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [107] Patricia S Crowther and Robert J Cox. A method for optimal division of data sets for use in neural networks. In *International conference on knowledge-based and intelligent information and engineering systems*, pages 1–7. Springer, 2005.

-
- [108] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [109] Z Hameed, YS Hong, YM Cho, SH Ahn, and CK Song. Condition monitoring and fault detection of wind turbines and related algorithms: A review. *Renewable and Sustainable energy reviews*, 13(1):1–39, 2009.