Fault detection of a wind turbine generator bearing using interpretable machine learning.

Oliver Trygve Bindingsbø

Thesis for the Degree of

Master of Science in Ocean Technology

Department of Physics and Technology

University of Bergen

August 2023

Acknowledgements

This master's thesis has been carried out at the University of Bergen and Western Norway University of Applied Sciences. It is submitted as a requirement for the degree of Master of Science at the University of Bergen (UiB), in the study program of Ocean Technology, within the specialization of Marine Installations.

I would like to express my sincere gratitude to my supervisor, Prof. Maneesh Singh (HVL), for his unwavering support, guidance, and patience throughout my research journey. His expertise in the field and willingness to go above and beyond has been invaluable in shaping my work and helping me overcome any challenges I faced. I would also like to thank Prof. Knut Øvsthus (HVL) and Assoc Prof. Arvind Keprate (OsloMet). It would not have been possible without our valuable conversations and your endless ideas and proposals. I truly felt like a member of the team during the thesis work, which was a great experience.

Furthermore, I want to thank my co-supervisors, Prof. Bjørn Tore Hjertaker (UiB) and Leading Researcher Jone Torsvik (Equinor), for their advice and support throughout my master's period.

Lastly, I would like to thank my friends and family for creating a healthy work-life balance. An especially large thanks to my fellow students that I have shared my griefs and joys with throughout these last five years.

Abstract

A wind turbine is subjected to a number of degradation mechanisms during its operational lifetime. If left unattended, the degradation of components will result in poor performance and potential failure. Hence, to mitigate the risk of failures, it is imperative that the wind turbines are regularly monitored, inspected, and optimally maintained.

Offshore wind turbines are normally inspected and maintained at fixed intervals (generally six-month intervals) and the maintenance program (list of tasks) is prepared using experience or risk-based reliability analysis, like risk-based inspection (RBI) and reliability-centered maintenance (RCM). This time-based maintenance program can be improved by incorporating results from condition monitoring (CM) involving data acquisition using sensors and fault detection using data analytics. It is important to ensure quality and quantity of data and to use correct procedures for data interpretation for fault detection to properly carry out condition assessment.

This thesis contains the work carried out to develop a machine learning (ML) based methodology for detecting faults in a wind turbine generator bearing. The methodology includes application of ML using supervisory control and data acquisition (SCADA) data for predicting the operating temperature of a healthy bearing, and then comparing the predicted bearing temperature with the actual bearing temperature. Consistent abnormal differences between predicted and actual temperatures may be attributed to the degradation and presence of a fault in the bearing. This fault detection can then be used for rescheduling the maintenance tasks. The methodology is discussed in detail using a case study.

In this thesis, interpretable ML tools are used to identify faults in a wind turbine generator bearing. Furthermore, variables affecting the generator bearing temperature are investigated. The analysis used two years of operational data from a 2 MW offshore wind turbine located in the Gulf of Guinea off the west coast of Africa. Out of the four ML models that were evaluated, the XGBoost model was determined to be the most effective performer. After utilizing the Shapley additive explanations (SHAP) to analyze the XGBoost model, it was determined that the temperature in the generator phase windings had the most significant effect on the model's predictions. Finally, based upon the deviation between the actual and the predicted temperatures, an anomaly in the generator bearing was successfully identified two months prior to a generator failure occurring.

Sammendrag

En vindturbin utsettes for en rekke degraderingsmekanismer i løpet av sin levetid. Hvis den ikke får tilsyn, vil degradering av komponenter resultere i lav ytelse og potensielt havari. Det er viktig at vindturbinene overvåkes, inspiseres og vedlikeholdes regelmessig for å redusere risikoen for feil.

Vindturbiner til havs blir normalt inspisert og vedlikeholdt med faste intervaller (vanligvis seksmåneders intervaller), og vedlikeholdsprogrammet (liste over oppgaver) er utarbeidet basert på erfaring eller risikobasert pålitelighetsanalyse, som risikobasert inspeksjon (RBI) og pålitelighetssentrert vedlikehold (RCM). Dette tidsbaserte vedlikeholdsprogrammet kan forbedres ved å inkludere resultater fra tilstandsovervåking (CM) som involverer datainnsamling ved hjelp av sensorer og feildeteksjon ved hjelp av dataanalyse. Det er viktig å sikre kvalitet og kvantitet på data og å bruke korrekte prosedyrer for datatolkning for feildeteksjon for å utføre tilstandsvurdering på riktig måte.

Denne oppgaven omhandler arbeidet som er utført for å utvikle en maskinlæringsbasert metodikk for å oppdage feil i et vindturbingeneratorlager. Metodikken inkluderer bruk av maskinlæring (ML) ved bruk av overordnet styring, kontroll og datainnsamling (SCADA) data for å forutsi driftstemperaturen til et sunt lager; og deretter sammenligne den predikerte lagertemperaturen med den faktiske lagertemperaturen. Konsekvente unormale forskjeller mellom predikerte og faktiske temperaturer kan tilskrives forringelse og tilstedeværelse av en feil i lageret. Denne feildeteksjonen kan deretter brukes til å planlegge vedlikeholdsoppgavene på nytt. Metodikken diskuteres i detalj ved hjelp av en casestudie.

I denne oppgaven brukes tolkbare ML-verktøy for å identifisere feil i et vindturbingeneratorlager. Videre undersøkes variabler som påvirker generatorens lagertemperatur. Analysen brukte to års driftsdata fra en 2 MW vindturbin til havs lokalisert i Guineabukta utenfor kysten av vest Afrika. Av de fire ML-modellene som ble evaluert, ble XGBoost-modellen funnet til å være den mest effektive. Etter å ha brukt «Shapley additive explanations» (SHAP) for å analysere XGBoost-modellen, ble det funnet at temperaturen i generatorfaseviklingene hadde den mest signifikante effekten på modellens prediksjoner. Basert på avviket mellom de faktiske og predikerte temperaturene, ble en anomali i generatorlageret identifisert to måneder før en generatorfeil oppstod.

Table of Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AI	=	Artificial Intelligence
CBM	=	Condition-Based Maintenance
CM	=	Condition Monitoring
HAWT	=	Horizontal Axis Wind Turbine
IQR	=	Interquartile Range
LCOE	=	Levelized Cost of Energy
LR	=	Linear Regression
ML	=	Machine Learning
O&M	=	Operation and Maintenance
PdM	=	Predictive Maintenance
PM	=	Preventive Maintenance
Q1	=	First Quartile
Q3	=	Third Quartile
RBI	=	Risk-Based Inspection
RCM	=	Reliability-Centered Maintenance
RF	=	Random Forest
RH	=	Relative Humidity
SCADA	=	Supervisory Control and Data Acquisition
SHAP	=	Shapley Additive Explanations
SVR	=	Support Vector Regression
VAWT	=	Vertical Axis Wind Turbine
WF	=	Wind Farm
WT	=	Wind Turbine

Symbols

P = Power

ρ = Air density

A_s = Area swept by rotor

U_∞ = Wind speed

C_p = Turbine efficiency

ϕ_i = Shapley value

r_{xy} = Correlation coefficient

$\bar{x}$ = Mean of variable X

$\bar{y}$ = Mean of variable Y

n = Sample size

W = Time-averaged wear rate

K = Wear coefficient

H = Hardness of the material being subjected to wear

Q = The time-dependent load at a given interaction

u = Sliding velocity

1. Introduction

1.1 Background and Motivation

To meet the demand for energy and reduce dependency on conventional fossil fuels, a large growth in energy supply from wind is required (IEA, 2021). It is crucial that wind turbines (WT) have high availability and low cost to be a dependable and cost-effective energy source. However, WTs suffer from high operation and maintenance (O&M) costs, especially offshore, which contribute to constraining rapid wind power development (IRENA, 2022). Condition-based maintenance (CBM) has been proposed as a solution to reduce O&M costs by identifying initial anomalies in WTs, thus providing the possibility to correct them before failure occurs (Ren *et al.*, 2021). The generator is a critical component of a WT and has bearings which are highly susceptible to failures leading to long downtime (Liu *et al.*, 2018). Hence, it is important to detect incipient anomalies in the generator bearings to reduce O&M costs.

Data-driven methods have been developed to monitor WT drivetrains based on data from the supervisory control and data acquisition (SCADA) system and are found to be effective for timely drivetrain fault diagnosis as well as anomaly detection of main bearings, generators, and gearboxes (Astolfi, 2023; Encalada-Dávila *et al.*, 2021; Jin, Xu and Qiao, 2021; Cui, Bangalore and Tjernberg, 2018). However, most of the existing works examining failure modes of large-scale WT bearings are only validated with purposely introduced defects in idealized lab settings, and hence there is a need to evaluate condition monitoring (CM) and fault diagnosis methods on naturally damaged WTs operating in the field (Liu and Zhang, 2020).

This work uses machine learning (ML) models based on data from the SCADA system to detect anomalies in the generator bearings and identify important variables affecting the generator bearing temperature. Shapley additive explanations (SHAP), a solution concept from game theory, is used to extract knowledge from the black box nature of ML models.

1.2 Aim of the Project

The aim of the project is to identify existence of faults in a WT generator bearing based on deviation in predicted and actual temperatures using interpretable ML models.

1.3 Scope of Work

The scope of work includes development of ML models to predict WT generator bearing temperature based on the variables: wind speed, generator rpm, nacelle temperature, generator temperature, and humidity. Several ML models are explored, and these are linear regression (LR), random forest regression (RF), support vector regression (SVR) and XGBoost. The importance of each variable will be discussed based upon Shapley values. Finally, an explanation of the best performing ML predictive model is performed.

1.4 Limitations

To be able to complete the project in due time and with the resources available, some limitations have been applied. Only data from one offshore WT is analyzed. Local environmental conditions and random component deficiencies may impact the results. Therefore, the proposed procedure should only be considered as general guidance. Additionally, only five variables are considered in the case study. These variables are wind speed, generator rpm, nacelle temperature, generator temperature, and humidity. There may exist other variables which could indicate faulty turbine behavior and cause cascading effects.

1.5 Structure of Thesis

The thesis is composed of 7 chapters and 1 appendix. In chapter 1, the motivation and background of the project is presented. A brief overview of previous work related to the subject of the thesis is given, before the scope of work is presented along with its limitations. Chapter 2 gives a brief introduction to offshore WTs and SCADA data for CM of bearings. In chapter 3, ML background theory is described along with relevant algorithms and methods for interpreting ML models. Chapter 4 contains the proposed fault detection methodology. Additionally, a case study is included. In chapter 5, the results and discussion are presented based upon the case study. Chapter 6 summarizes the findings from the study. The study's contributions and possible limitations of the approaches used are discussed. Finally, chapter 7 concludes the thesis by presenting possible directions for further work. Appendix 1 contains Bindingsbø *et al.* (2023), a paper under review which is an extension of the thesis.

2. Introduction to Wind Turbines

Wind turbines (WTs) vary a lot in size, and the length of the blades is the greatest contributor to the amount of electricity a WT can generate. A small WT powering a single home may have an electricity generating capacity of 10 kW, whereas the largest WTs in operation have an electricity generating capacity of around 14000 kW, or 14 MW, and are constantly increasing in size (GE, 2023). WTs are commonly grouped together to form wind farms providing power to electricity grids. This chapter describes offshore WTs and their main components. Finally, SCADA data from WTs and CM data from bearings are introduced.

2.1 Wind Turbine Functionality

Wind as mechanical energy is one of the oldest forms of energy humans have utilized, for example moving a sailboat by capturing the wind energy in the sails and converting it to thrust (Hofstad and Rosvold, 2022). Since around year 600, wind energy has been utilized in windmills to pump water and grind grain (Mæhlum and Rosvold, 2019). Today the term is mostly used for electric power generation in a WT. A WT is a machine used to convert the kinetic energy in the wind into electricity. The blades on the WT collect the kinetic energy from the wind flowing through them, creating lift, and causing the blades to turn. Rotational energy from the blades powers a drive shaft which turns a generator, converting the rotation to electricity (Hansen, 2020). In older windmills, rotation is mainly generated by the air pressing against the rotating blades, whereas in modern WTs, the rotation is driven by the aerodynamic lift force and the rotor blades having a small resistive force. Thus, the rotor blades of a modern WT have more in common with the wings on an airplane than with windmills. By utilizing the lift force, modern WTs are able to capture over half of the available energy in the wind, thus they have an efficiency of around 50 %. This is a high efficiency compared to other types of renewable energy, and considering that Betz's limit indicates that the maximum power that a WT can generate is limited to 16/27 or 59,3 % of the kinetic energy in the wind (Badurek, 2023).

The amount of electricity a WT can produce in one year can be calculated from the turbine's power capacity factor. The power capacity factor describes the actual power produced in one year and compares it to what would have been produced if the turbine had operated at full power the whole year. Modern onshore windfarms typically have a power capacity factor of 30 % to

45 % whereas modern offshore wind farms have a power capacity factor of 50 % and slightly above (Wind Europe, 2023). One year consists of 8 760 hours, and as an example, a WT with a power capacity factor of 50 % is equivalent to the turbine producing at full capacity for 50 % of the 8 760 hours of the year, which equals 4 380 hours. A 14 MW turbine would then produce $14\ \text{MW} \times 4\,380\ \text{hours} = 61\,320\ \text{MWh}$ per year.

2.1.1 Power Curve

For a particular WT, the expected power generated can be estimated from a wind speed power curve, typically a graph showing the relation between wind speed and power generated. The power curve for a WT is specified by the manufacturer and describes the expected power output at different wind speeds. Figure 2.1 displays a typical power curve where the blades start rotating at the cut-in wind speed and the generator starts producing power. As the wind speed increases, the turbine produces more power until it reaches its rated output speed. From there on the power generation remains constant while wind speed increases, until the wind speed reaches the cut-out speed. Here, the turbine shuts down to avoid unnecessary strain on components (EERE, 2022b). Cut-in speed, rated output speed and cut-out speed are supplied by the turbine manufacturer. To assure the WT is performing according to specification, power curve warranties are commonly included in contracts (Miceli, 2012).

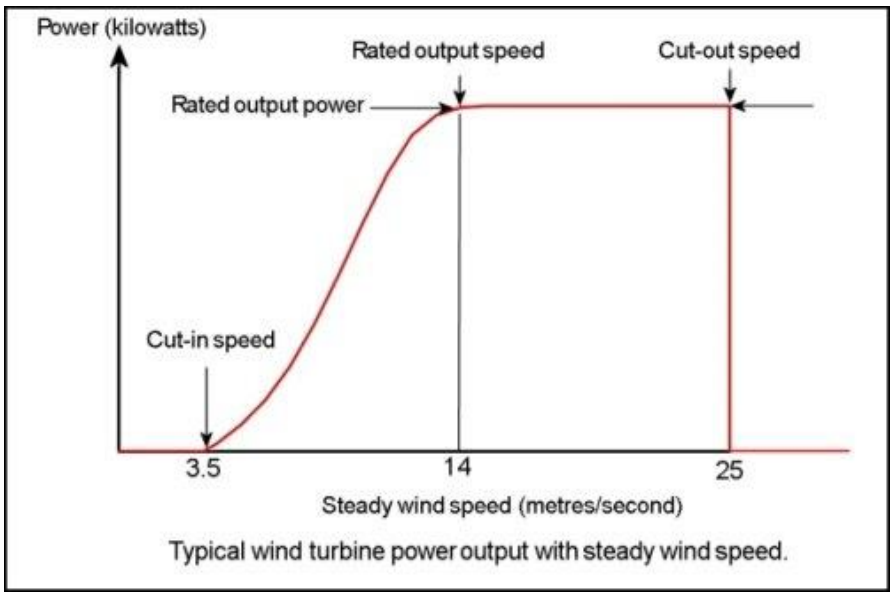

Figure 2.1: Example of power curve (Cole, 2023)

Variables such as area swept by rotor, number of blades, and rotation speed, unique to each turbine, impact the wind speed power curve. The power a WT can produce is given by:

$$P = 0{,}5\rho A_s U_\infty^3 C_p, \tag{2.1}$$

where it is shown that the power, P, is dependent on the wind speed to the power of three U_∞^3, the turbine efficiency C_p, the air density ρ, and the area swept by the rotor A_s. Air pressure is approximately constant at feasible wind farm locations, therefore only higher wind speed, higher turbine efficiency and larger area swept can increase power. Wind speed is area-dependent and to the power of three as shown in Equation (2.1). This contributes to making offshore WTs more attractive due to higher and more stable wind speeds. The turbine efficiency on modern turbines is difficult to improve, therefore it is mainly by increasing the area swept by the rotor blades that the power of the WT can be increased. This is done by increasing the length of the rotor blades; thus, the size of WTs is still increasing and will be dependent on what is technologically possible and economically feasible (Hansen, 2020). The power curve typically gives the gross power output, not considering factors such as wake effect, turbine availability, transmission efficiency, turbine performance and environmental conditions (WindFacts, 2009). These factors are described in the sections below.

2.1.2 System Availability

To understand the power production of the whole wind plant, it is necessary to consider the system availability. The system availability factor counts all downtime against availability, no matter the cause (DNV, 2022). This factor considers the turbine availability in addition to other factors such as grid availability and balance of plant availability. The turbine availability is a factor which describes the expected average availability of the WT or WF throughout its lifecycle. It is given as a percentage which is calculated into the gross energy output to account for the duration where the turbine is unavailable for power generation. In addition to the turbine availability factor, it is also necessary to consider the grid availability, which describes the availability of the grid to export power, as well as the balance of plant, which describes the reliability of the WTs components other than the turbine such as the electrical infrastructure.

2.1.3 Wind Turbine Components

There are two basic types of WTs (EIA, 2022), the horizontal-axis wind turbine (HAWT) and the vertical-axis wind turbine (VAWT), displayed in Figure 2.2. Not only do they look different, but they also function differently. The main rotor shaft of the VAWT is arranged vertically, whereas for the HAWT it is arranged horizontally. An advantage of the VAWT layout is that the turbine does not need to be positioned pointing into the wind, thus it can work well on sites where the wind direction is highly variable. In addition to this, the vertical axis allows for the gearbox and generator to be situated near the ground, making it more accessible for maintenance. However, the VAWT encounters difficulties such as creation of drag when the blades rotate, and some designs produce pulsating torque (Bhatia, 2014). HAWTs make up the vast majority of WTs currently in commercial utility-scale use, due to their higher efficiency and power output (Varghese, Roy and Awasthi, 2022), and are thus focused on in this thesis.

Figure 2.2: Horizontal axis wind turbine (left) and vertical axis wind turbine (middle and right) (Marietta, 2023)

Wind power is often separated into onshore wind and offshore wind, where the latter is usually larger in size and power capacity. Offshore WTs can be either bottom fixed or floating and require different support structures for its towers than onshore WTs. The main components of a WT are tower, blades, hub, and nacelle, shown in Figure 2.3 and described in this chapter.

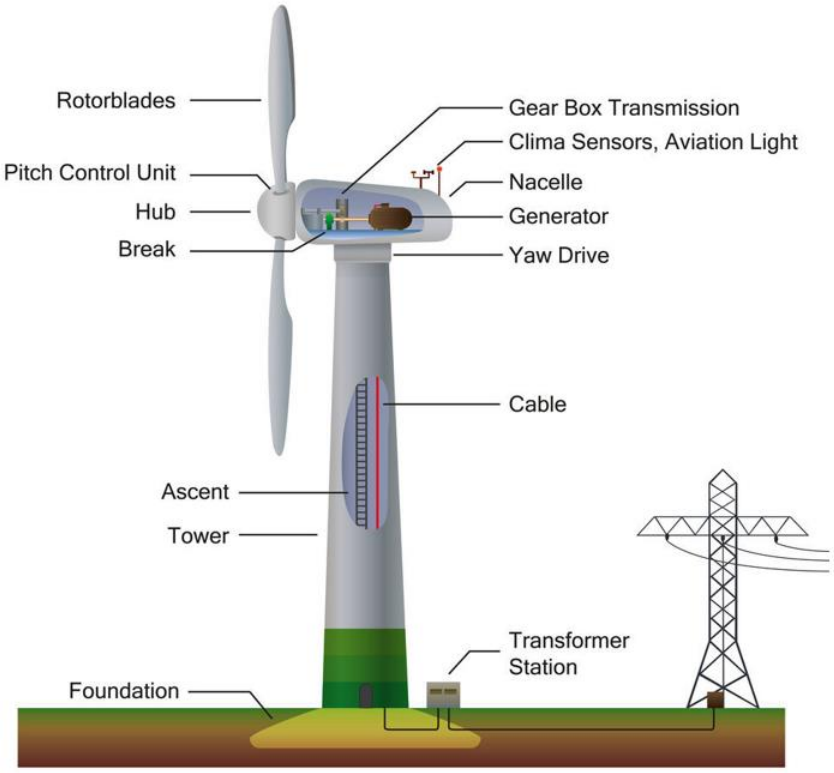

Figure 2.3: Common WT components (Bilderzwerg, 2022).

Tower

The tower of a HAWT carries the rotor blades, hub, and nacelle, and transmits loads from the nacelle to the foundation. A tall tower allows for longer rotor blades and taking advantage of larger and more stable wind speeds, thus increasing the power production capacity of the WT (EERE, 2022a). There are a variety of tower designs available as seen in Figure 2.4, such as lattice tower, tubular tower, and guyed pole. The most common tower design is the tubular tower made of steel due to the increasing size and weight of the rotor blades. Tubular towers often come in three sections and are assembled on-site. Another advantage of the tubular tower is that it encapsulates important components such as the transformer, which other tower designs are not capable of. The tubular steel tower is expensive and heavy due to the large amount of steel required. Lattice towers require less material; thus, they are lighter and cheaper, but are only suitable for small WTs (Lantz *et al.*, 2019). A hybrid tower combining the tubular design and the lattice design can offer the benefits from both and is suitable for medium sized WTs.

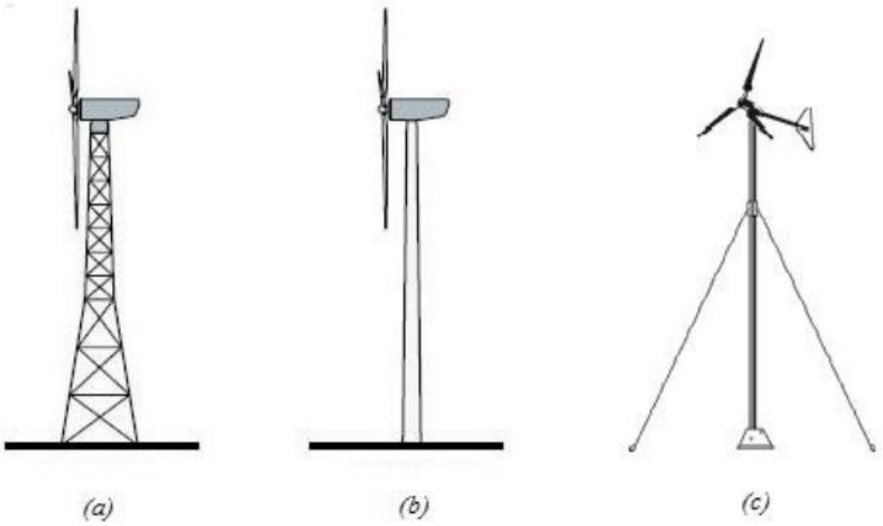

Figure 2.4: Different tower designs for WTs. (a): Lattice tower, (b): Tubular tower, (c): Guyed pole (Karmouche, 2016).

Rotor Blades

Most WTs have three rotor blades which are made mostly of fiberglass. The rotor blades vary in length and offshore WTs typically have longer blades than onshore WTs. Today, the largest offshore WTs have rotor blades longer than 100 m, whereas onshore WTs typically have rotor blades with a length of 50 m (EERE, 2022a). The air pressure on one side of the blade falls as wind passes across it. Both lift and drag are produced by the difference in air pressure on the blade's two sides. The rotor spins because the force of the lift is greater than the force of the drag. There are five main loads working on a WT blade (Thomsen, 2009):

- Flap- and edgewise bending from the pressure load on the blade.
- Gravitational loads which generate edgewise bending loading. This load changes direction when the blade is rotating.
- Torsional loading from the resulting shear of the flap- and edgewise loads.
- Normal loading caused by rotation of the blade.
- Loads from pitch de-accelerations and accelerations.

The flap- and edgewise loads are most important in determining structural design and the blade cross sections.

Pitch Control System

The pitch system adjusts the angle of the blades to control the rotor speed and how much energy the blades extract. It can also "feather" the blades by adjusting the blade angles so that there is no force to spin the rotor. This is typically performed to avoid damage during high wind speeds. The pitch control unit is commonly placed in the hub at the base of the rotor, as displayed in Figure 2.3. Modern WTs have separate pitch control systems for each blade instead of a collective pitch control system. This is because high-capacity WTs with long blades are subject to turbulence which impacts each blade differently thus causing uneven loads.

Pitch control systems are typically either hydraulic or electrical (Korkos *et al.*, 2022). Electric pitch control systems are gradually replacing hydraulic control systems due to their higher efficiency and mitigation of environmental concerns such as leakage and disposal of hydraulic fluid. A hydraulic pitch control system consists of components such as hydraulic cylinders, valves, pumps, pitch bearing and accumulator tanks.

According to a study, faults and failures occurring in the hydraulic pitch system account for 15,5 % of total failures and 20 % of total downtime of a WT (Wilkinson *et al.*, 2010). Furthermore, a newer study finds that 17% of the overall hydraulic/pitch failures were caused by oil issues (Carroll, McDonald and McMillan, 2016).

Yaw Control System

The yaw control system rotates the nacelle of the WT to keep the rotor blades facing the wind when the wind direction changes. To achieve optimal power extraction and prevent an unbalanced force distribution on the rotor, it is beneficial that the rotating plane of the rotor is perpendicular to the wind direction. An unbalanced force distribution may result in a failure of the WT. The yaw control system is placed right under the nacelle at the top of the tower, as displayed in Figure 2.3. Like the pitch control system, the yaw control system is also typically either electrical or hydraulic, with the electric option being used most frequently in modern WT units.

Drivetrain

The drivetrain on a WT is comprised of the bearings, shafts, gearbox, generator, and rotor. Low-speed rotation of the turbine's rotor blades is converted into electrical energy in the drivetrain. The gearbox increases the rotational speed of the shaft connected to the generator, so that the generator works at an optimal speed for electricity generation. The drivetrain also typically houses the WT braking system which can be mounted on the low-speed shaft by the hub or the

high-speed shaft by the generator as displayed in Figure 2.5. At high wind speeds the brake will regulate the rotor speed together with the pitch control system.

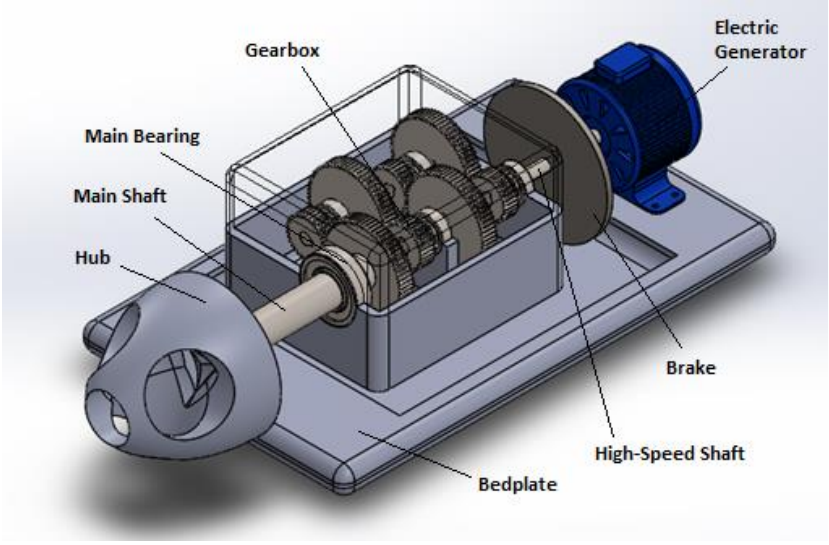

Figure 2.5: Drivetrain components (Tran, 2021).

Gearbox

The purpose of the gearbox in a WT is to increase the rotational speed of the low-speed rotor to the high-speed shaft that drives the generator. A gearbox typically consists of several stages of gears that increase the speed while reducing the torque of the rotational input (Letcher, 2017). The rotor turns at low speed and transfers torque to the input stage of the gearbox consisting of a set of gears that increase the speed of the rotor while reducing its torque. From here, the high-speed, low-torque output is transferred to the next stage of gears, which further increases the speed while reducing the torque, and the process continues until the final stage, which delivers a high-speed, low-torque output to the high-speed shaft that drives the generator. The gearbox is a critical component in a WT system, as it must withstand the high torque and forces generated by the rotor (Salameh *et al.*, 2018). A typical gearbox in a WT uses planetary or spur gears to achieve the speed increase and the gears are lubricated and cooled to ensure reliability and long service life.

Generator

The generator is an electromechanical component, the purpose of which is to convert mechanical power into electrical power. There are two main types of generators used in the

industry: synchronous and asynchronous. A synchronous generator operates at the synchronous speed determined by the connected grid frequency, independent of the amount of torque applied (Circuit Globe, 2023). It is more expensive and mechanically complex than a similarly sized asynchronous generator, but it has one important advantage in that it does not require any power compensation device. Asynchronous generators, also called induction generators, are mass produced and priced low, as well as being robust and having mechanical simplicity. However, this generator consumes reactive power for excitation. This can be supplied by the grid or power electronics. The interaction of the associated rotor magnetic field with the stator magnetic field produces a torque acting on the rotor (Mendes *et al.*, 2020).

Bearings

Bearings in WTs are used to support the rotating components, such as the shaft and rotor, and to allow for smooth and efficient rotation. They consist of two main components: an inner race and an outer race. The inner race is typically attached to the shaft, while the outer race is attached to the housing. Between the two races are rolling elements which reduce friction and allow the shaft to rotate smoothly (Manwell, McGowan and Rogers, 2009). The main type of bearings used in WTs are roller bearings, also known as rolling element bearings (Hart *et al.*, 2020), shown in Figure 2.6. Roller bearings can consist of different rolling elements such as balls, cylindrical rollers, tapered rollers, and spherical rollers. The shaft in the generator is supported by bearings operating at high speed, where excellent insulating properties and low vibration is essential (Whittle, 2013). The type of bearing used in a WT will depend on various factors such as the size and power of the turbine, the speed and loads on the rotating components, and the environmental conditions. Proper maintenance and lubrication of the bearings is critical to ensure their longevity and reliability (Hart *et al.*, 2020).

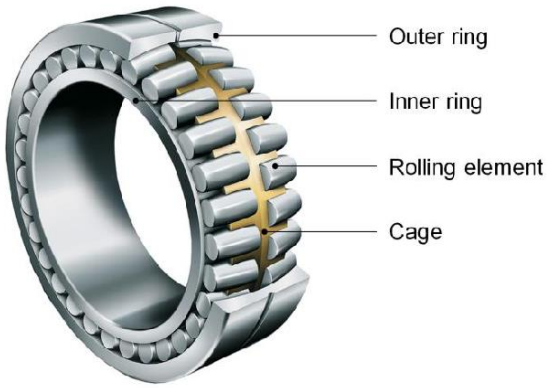

Figure 2.6: Roller bearing (Binderszewsky, 2014).

Support Structure

Offshore WTs are similar to onshore WTs but have different support structures. Offshore WTs are typically larger than onshore WTs due to less limitations from infrastructure such as road bends and are therefore able to capture more energy from the wind due to longer rotor blades. They also unlock a much larger surface area than onshore WTs. In addition to this, offshore wind tends to be stronger and more stable than onshore wind. A strong support structure is therefore required to withstand the harsh environmental conditions with large waves and strong winds that the offshore WT operates in (Fu, 2018). The support structure of an offshore WT is either bottom fixed or floating. Most of the bottom fixed structures are either monopile or jacket structures as shown in Figure 2.7, built of steel and fixed to the seabed by driven piles. Other bottom fixed structures exist, such as the gravity base, suction bucket and tripod structures.

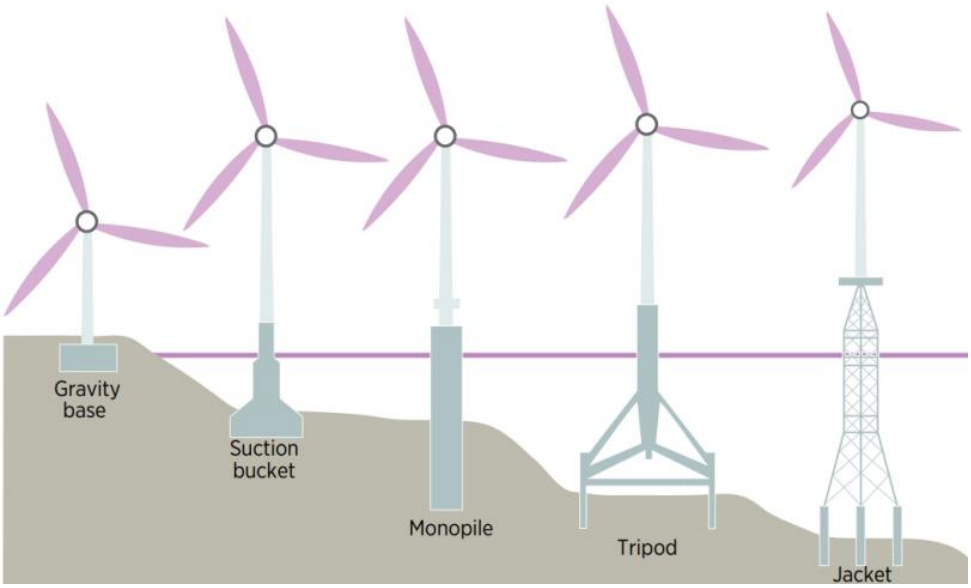

Figure 2.7: Offshore WT bottom fixed structures (IRENA, 2018).

Beyond a depth of 50 to 80 m, floating structures are more economical than bottom fixed structures (Matha, Lemmer and Muskulus, 2019). Removing the constraint of water depth opens up a world of new opportunities for wind power, allowing one to select the best sites in the world. Water deeper than 60 m accounts for nearly 80 % of the world's offshore wind resource potential (Equinor, 2022). The most used floating offshore WT support structures are spar buoy, semi-submersible and tension leg platform, displayed in Figure 2.8.

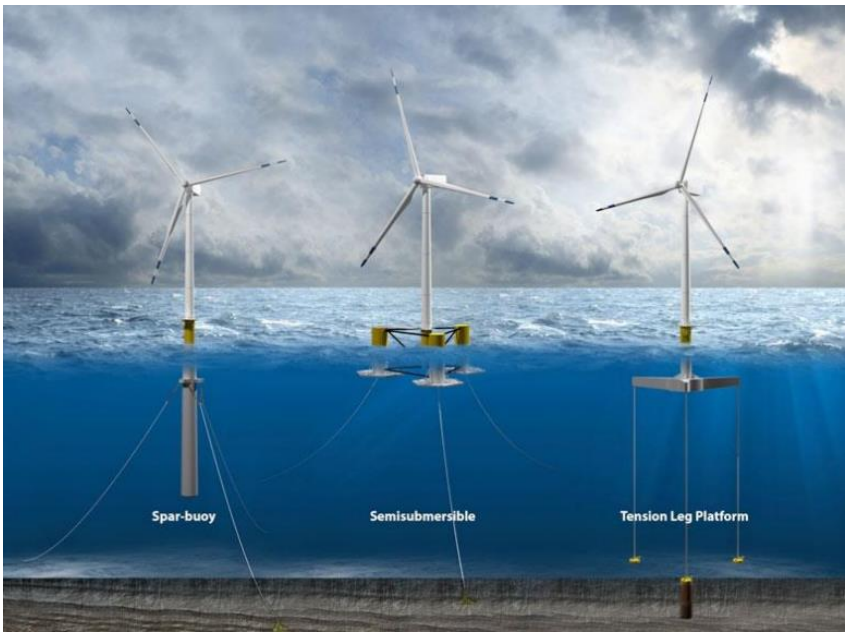

Figure 2.8: Offshore WT floating structures (IRENA, 2016).

2.2 SCADA Data from Wind Turbines

A supervisory control and data acquisition (SCADA) system is a computerized system that is capable of acquiring and processing data and applying operational controls over long distances (National Institute of Standards and Technology, 2023). They are often used to monitor and control WTs and can also be used to predict failures. One way to use SCADA to predict WT failures is to collect data from sensors on the turbine, such as vibration, temperature, and power output. This data can be analyzed to identify patterns that may indicate an impending failure.

Machines initially learn from the data they are fed, and it is therefore important to acquire high quality SCADA data so that a machine learning (ML) model can find the correct patterns. The quality of the data fed to the machine will determine how accurate the model is. If the data is incorrect or outdated, the model will give wrong outcomes or predictions which are not relevant.

2.3 Condition Monitoring Data from Bearings

This section describes maintenance strategies for offshore WTs along with failure modes associated with WT generator bearings.

2.3.1 Maintenance of Offshore Wind Turbines

Today, O&M costs account for 34 % of the levelized cost of energy (LCOE) of an offshore WT (Stehly, Beiter and Duffy, 2020). A reliable and effective maintenance strategy plays an essential part of daily operations of an offshore WF, which has higher maintenance costs than a WF on land. This is due to the environment they operate in and because maintenance technicians must take a service vessel out to the offshore WF, which can be located far from shore. The technicians must perform maintenance tasks under harsh weather conditions and are thus at risk. It is therefore important to reduce the time that maintenance technicians are exposed to the hazardous environment, as well as reduce the number of service vessel trips in order to reduce safety risk, cost, and emissions. However, reducing the visit frequency may lead to a higher failure rate and thus longer downtime. Maintenance frequency is therefore a trade-off among vessel costs, technician costs and risk.

2.3.2 Failure Modes

It is important to identify which failures cause downtime in order to improve future WT designs and solutions as well as tailor a maintenance program. A failure mode is the manner in which failure occurs while a fault is an inability to perform as required, due to an internal state. A failure may lead to a fault of a component, or the component itself can cause the fault from a deficiency in an earlier stage of the life cycle, such as specification, design, manufacture or maintenance (NORSOK, 2017). The failures occurring in an offshore WT are typically either from ageing, having endured a long operational life, or more sudden failures caused by sudden breakdown or short-term overload.

WT components such as the gearbox and the generator might have a low failure rate. However, their failure may result in a long downtime period lasting over six days (Sheng, 2013). High reliability of these components is therefore of importance to reduce O&M costs. Mechanical components cause large downtime, in some cases accounting for over 75 % of total downtime compared to electrical/control components (Artigao *et al.*, 2018). Examining the cause of failure for WT components shows that wear out is the main driver of failure in the gear shaft, the generator bearings, and the gearbox bearings (Faulstich, Lyding and Hahn, 2010).

2.3.3 Generator Bearings

ISO (2017) classifies the failure modes occurring while the bearing is installed and operating into six categories. These failure mode categories are rolling contact fatigue, wear, corrosion, electrical erosion, plastic deformation and cracking and fracture. Each of these are further divided into subcategories for more detailed classification of failure mode.

WT bearings can degrade due to several factors such as wear and tear from friction and vibration during operation, exposure to environmental factors such as moisture, dust, and extreme temperatures, and lack of lubrication or contamination of lubricant (Encalada-Dávila *et al.*, 2021). Bearing degradation can also result from overloading of the bearing, thus leading to excessive stress on the components, corrosion due to exposure to harsh weather conditions, and poor quality of bearings or manufacturing defects. All these factors can cause degradation of bearings in WTs and eventually lead to failures, which can impact the performance and reliability of the turbine.

2.3.4 Condition Monitoring

Condition monitoring (CM) in offshore WTs involves continuously monitoring health and performance of the turbines using sensors and advanced algorithms. Data on vibration, temperature, and other parameters are collected and analyzed in real-time to detect faults and optimize performance. This approach allows for predictive maintenance, reduces downtime, and ensures efficient offshore wind energy production (Ren *et al.*, 2021). Furthermore, integrating CM can enhance planning and prevent both excessive and insufficient maintenance. For instance, CM data can be used to predict the remaining useful life of the equipment (Shafiee, Finkelstein and Bérenguer, 2015). When selecting components for monitoring, it is crucial to consider the failure rates and downtimes associated with various sub-components. Priority should be given to components with higher probabilities of failure or those that can cause significant downtimes, as they have the most significant potential impact (Stetco *et al.*, 2019).

2.3.5 Maintenance Strategies

There exist different maintenance strategies for offshore WTs, and these are typically categorized as either failure-based, resulting in a reactive response, or proactive maintenance (Shafiee, 2015). The categorization is based on when the maintenance is performed. The former option is a reactive response which is carried out after a failure has occurred, whilst the latter is a proactive response carried out before any failure might occur, as displayed in Figure 2.9. Proactive maintenance is typically divided into preventive maintenance (PM) and predictive maintenance (PdM). Details of the most common maintenance strategies are explained in this section.

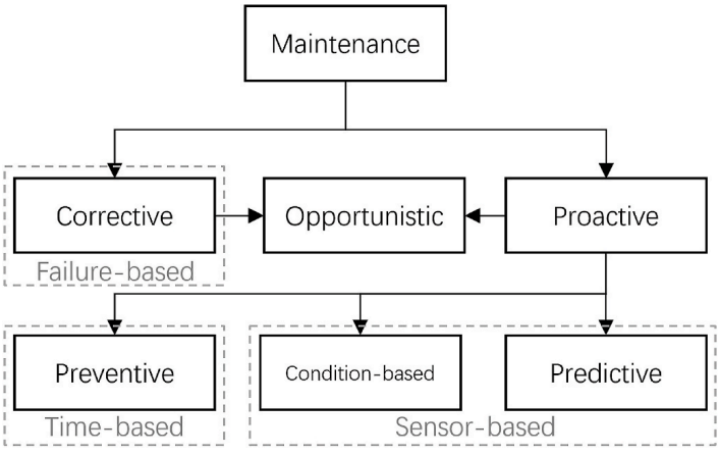

Figure 2.9: Categorization of maintenance strategies (Ren et al., 2021).

Corrective Maintenance

Corrective maintenance, also referred to as reactive maintenance, can be either planned or unplanned, and is failure-based. This means that maintenance is carried out after a failure has occurred and it is described as "maintenance carried out after fault detection to affect restoration" (NORSOK, 2017). Corrective maintenance can be useful to avoid unnecessary maintenance visits and inspections for a system with negligible downtime loss. However, for large-scale systems such as offshore WFs, corrective maintenance is impractical due to their high failure rate and relatively low reliability. Hence, unexpected failures may turn out to cost more than expected downtime (Ren *et al.*, 2021). Additionally, the offshore environment suffers from restricted accessibility and reduced reliability, hence a failure may go unnoticed for a long period of time. A corrective maintenance strategy for offshore WFs is therefore not practical,

as future offshore WFs are increasingly being constructed farther away from the coast (Shafiee, 2015).

Preventive Maintenance

Major failures account for 25 % of all failures, yet they contribute to 95 % of total downtime of a WT (Faulstich, Hahn and Tavner, 2011). PM, a proactive maintenance strategy, schedules inspection and replacement to be performed before failure to prevent minor faults developing into major failure (Ren *et al.*, 2021). It is described as "maintenance carried out to mitigate degradation and reduce the probability of failure. Preventive maintenance contains condition-based maintenance and predetermined maintenance" (NORSOK, 2017). Thus, the maintenance is usually scheduled to take place during a predetermined period or at a given level of power generation.

Predictive Maintenance

PdM is described as "maintenance based on the prediction of the future condition of an item estimated or calculated from a defined set of historic data and known future operational parameters" (NORSOK, 2017). The proactive PdM strategy utilizes sensor data from a CM system to perform analysis determining when maintenance should be carried out before failure occurs. It can contribute to reduced downtime and increased availability through performing maintenance only when strictly necessary. PdM requires investment in measurement equipment, but reduced downtime, cost of spare parts, and maintenance frequency make up for this (Ren *et al.*, 2021). The Digital Twin (DT) concept, which has become a popular research topic in recent years, can be used for PdM and scheduling. A comprehensive DT can generate real-time big data through multiple sensors monitoring a physical asset. This data can be used to better schedule maintenance by performing a smart analysis of the data to detect faults in the system long before they can occur (Rasheed, San and Kvamsdal, 2020).

Condition-Based Maintenance

Condition-based maintenance (CBM) combines measurements performed by a CM system and the results of a health diagnosis or fault system. It is defined as "preventive maintenance based on the assessment of physical condition" (NORSOK, 2017). The degree of deterioration of a WT can be observed with the CBM strategy.

Opportunistic Maintenance

Opportunistic maintenance takes advantage of the unplanned or planned shutdown of a system where appropriate maintenance resources are already on location, and is a form of preventive maintenance (Borges, 2023).

3. Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) where statistical methods are applied to make computers identify patterns in large amounts of data (Tidemann and Elster, 2022). Instead of being programmed, the machine learns through training. A data set is typically split into a training set and a test set, where the training set is used for training the model, and the test set is then used to check if the model has learned what is necessary.

Today, ML is used in everything from image recognition and self-driven cars, to improved web-search. ML relies on various algorithms to solve data problems. There is no one size fits all algorithm for problem solving. The type of algorithm to use depends on the type of problem to be solved, the number of variables, the type of model that works best, and other factors. Typically, it belongs to one of the three learning paradigms, depending on the nature of the problem (Choi *et al.*, 2020):

- **Supervised learning** – The model learns by understanding the relationship between features and a given set of labels.
- **Unsupervised learning** – No labels are provided to the model, leaving it on its own to learn patterns for a set of features.
- **Reinforcement learning** – The model learns iteratively through receiving feedback from its output.

Each of these strategies require different models and performance metrics. Only supervised learning will be used in the thesis.

3.1 Algorithms

An ML model predicts the output after running an ML algorithm on the collected data. Four models are selected in order to compare results and find the best fit. In the case of predicting the generator bearing temperature, a continuous numerical value, regression models are used. The selected models are linear regression (LR), random forest (RF), support vector regression (SVR) and XGBoost. These models are commonly used and are known as supervised learning algorithms (Mahesh, 2020). The four models are explained further in this section, along with an introduction to decision trees, a supervised learning method used in RF and XGBoost.

3.1.1 Decision Tree

A decision tree is a graph used to visualize choices and subsequent results in the form of a tree, such as shown in Figure 3.1. A parent node is a node that splits into sub-nodes, and the sub-nodes are the child nodes of the parent node from which they derived (Belyadi and Haghighat, 2021). The root node is the first node in the path from which all decisions initially start, and it has two child nodes and no parent node. A decision node has one parent node and splits into two child nodes. The child nodes can be of both decision nodes and leaf nodes. A leaf node has one parent and does not split further, and it is this node that represents the prediction. The maximum depth represents the length of the longest path from the tree root to a leaf, and the root node is considered to have a depth of 0. A subsection of the tree is called a branch. When the target variable can take continuous values, the decision tree is called a regression tree.

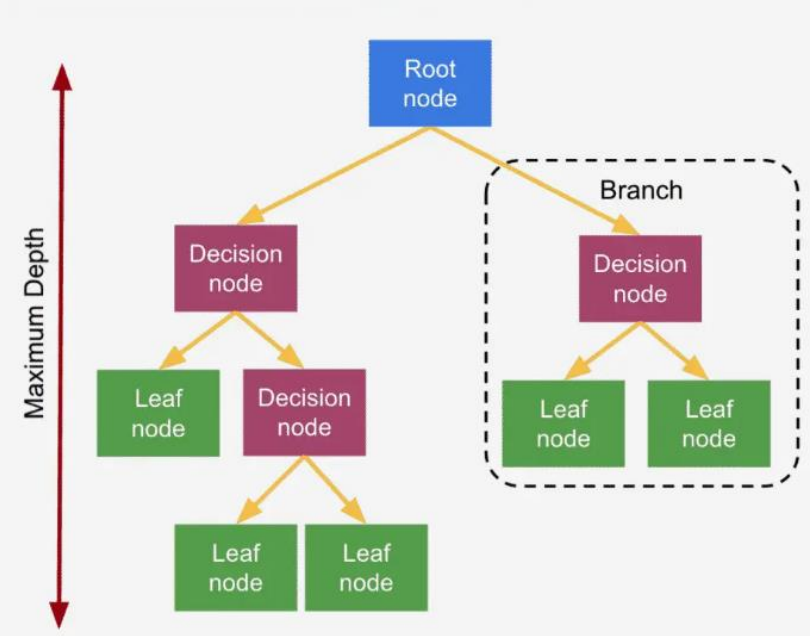

Figure 3.1: Example of decision tree (Chouinard, 2023)

3.1.2 Linear Regression

Linear regression (LR) is a statistical tool commonly used for predicting numeric values. The goal of LR is to find a linear relationship between the independent variables and the dependent variable, such that the model can predict the value of the dependent variable for new values of the independent variables. Assuming there exists a linear relationship between the variables, the model estimates parameters, including the intercept and slope coefficients.

LR is a simple and easily interpretable algorithm and is therefore widely used in ML. It can provide insights into the relationship between the independent and dependent variables, such as the strength and direction of correlations (Montgomery, Peck and Vining, 2021). In the task of predicting the generator bearing temperature, the performance of the LR model is evaluated using ordinary least squares during training.

3.1.3 Random Forest

Random forest (RF) is a learning method that combines multiple decision trees, also thought of as a forest of decision trees, to create a model. Instead of using a single decision tree to predict, RF takes predictions from thousands of decision trees, thus utilizing much more knowledge. Since it combines many decision tree models into one, it is known as an ensemble algorithm.

RF combines decision trees into one by using bagging, also called bootstrapping. In bagging, a large number of decision trees are created by sampling different subsets of the training data and a random selection of features. The sampling is performed with replacement, and each tree makes its own prediction. For regression problems, the final prediction is determined by averaging over the predictions of all the trees. Bagging has the advantage of reducing variance by averaging and avoiding overfitting by using random samples and features (Breiman, 2001).

3.1.4 Support Vector Regression

Support vector regression (SVR) is another ML algorithm based on statistical theory. Its objective is to maximize the margin between support vectors through a separating hyperplane, where the margin is the distance between the hyperplane and the closest data points. It maps the data x from the input space to the high-dimensional feature space M through a nonlinear mapping. The nonlinear regression analysis is performed in the feature space M to find an optimal function $f(x)$, which is used to predict the dependent variable based on the independent variables in x. SVR has the advantage of having a good generalization ability and handling high-dimensional data, however, it can be computationally intensive when dealing with large data sets with more than a few ten thousand samples since the fit time complexity is more than quadratic with the number of samples (Smola and Schölkopf, 2004).

3.1.5 XGBoost

XGBoost (eXtreme Gradient Boosting) is an ensemble learning method that uses a combination of weak decision trees to create a model. It is an extension of the gradient boosting algorithm that uses a gradient-based optimization approach to minimize the loss function. In boosting, instances with inaccurate predictions are given more weight. Therefore, the tough cases that are being incorrectly predicted are the focus of boosting. In contrast to bagging, which employs an equal weighted average, boosting makes use of a weighted average but gives the models with the best performance more weight. Put in another way, boosting gives samples whose predictions are of a higher weight, which encourages more frequent sampling of those samples.

Gradient boosting compares the predictions with the actual values after building the weak learners. The error rate of the model is represented by the difference between the predicted and actual values. A gradient is calculated from the error rate and is used to identify the direction that the model parameters would have to change in order to reduce the error in the next round of training. XGBoost builds each tree in a greedy manner, selecting the best split at each step based on the reduction in the loss function. It improves gradient boosting for scale and computational speed in several ways, making it suitable to handle large data sets and big data applications. XGBoost achieves this by parallel learning, meaning the usage of multiple CPU cores (Chen and Guestrin, 2016).

3.2 Interpretable Machine Learning

ML can offer decision-making assistance and even prompt decisions. However, an ML model may have high prediction accuracy but lack interpretability. The ML model acts as a black box which makes it challenging to understand, and users may not have knowledge of the underlying decisions in the predicting process (Ekanayake, Meddage and Rathnayake, 2022). Interpretable ML methods can be applied to provide insight into how ML models work, what factors are driving their predictions, and how confident they are in their predictions. This may be used to justify the model and its predictions, and to further improve the model (Adadi and Berrada, 2018). Interpretable ML is currently at a stage where it is sufficiently developed and mature, but there are still some challenges that need to be addressed (Molnar, Casalicchio and Bischl, 2020). It aims to transparentize black box ML models, revealing how predictions are performed and the importance of features and dependencies. Interpretable ML methods are typically

categorized depending on what part of the ML model they analyze, such as the model components, model sensitivity or surrogate models (Molnar, Casalicchio and Bischl, 2020).

3.2.1 Interpretable Model Components analysis

Analysis of model components is possible for models which can be decomposed into pieces that can be interpreted individually, and it is not necessary for the user to comprehend the model in its totality. Since component analysis is dependent on the model's structure, it is always model-specific. Models having structures and parameters that may be given a specific interpretation are said to be intrinsically interpretable, such as linear regression models and decision trees (Molnar, Casalicchio and Bischl, 2020).

3.2.2 Complex Model Components analysis

Analyzing more complex models such as random forests and deep convolutional neural networks requires more work. Model component analysis can be the right method if an ML algorithm is well known and often used in a community, but it has the drawback of being bound to that particular model. Additionally, it does not work well with the typical ML model selection method, which typically involves cross validating a large class of various ML models (Molnar, Casalicchio and Bischl, 2020).

3.2.3 Individual Prediction Explaining

Most methods for testing an ML model's sensitivity are model agnostic, meaning that they can be used to explain any ML method, and work by manipulating the input data and scrutinizing the corresponding model predictions. These IML techniques are divided into local and global explanations and typically treat the ML model as a closed system that takes inputs in the form of feature values and outputs predictions. Individual predictions made by ML models are explained by local IML methods. Shapley additive explanations (SHAP) is one such method, explained further in section 3.2.6 (Molnar, Casalicchio and Bischl, 2020).

3.2.4 Global Model Behavior Explaining

The expected model behavior, or how the model acts typically for a given data set, is explained using global model-agnostic explanation techniques. Global explanations possess values for feature importance and feature effect. Based on the relevancy of features for a prediction, they are ranked in the feature importance, whereas the feature effect describes how altering a feature alters the prediction (Molnar, Casalicchio and Bischl, 2020).

3.2.5 Surrogate Models

Interpretable models known as surrogate models are created to replicate the behavior of ML models. The surrogate technique sees the ML model as a black box, and all that is needed to train a surrogate ML model is the model's input and output data. The interpretation is thereafter performed by analyzing the surrogate models' components. Numerous IML techniques are surrogate model approaches and vary, for instance, in the data sampling technique or the employed interpretable model (Molnar, Casalicchio and Bischl, 2020).

3.2.6 Shapley Additive Explanations

The problem of interpretability while utilizing ML approaches has previously been addressed (Vilone and Longo, 2020). Recently, the research community has become more interested in the Lundberg and Lee (2017) SHAP method which proposes a model agnostic representation of feature importance estimated by Shapley values in a computationally efficient manner. Shapley values are a solution concept from collaborative game theory. The SHAP method is an additive feature attribution method that considers the features as "the players", combinations of different features as "the coalitions", and the prediction as "the total payout". The average marginal contribution for feature i over all possible coalitions is the Shapley value ϕ_i, hence it explains each feature's contribution to a prediction. Additive feature attribution methods have an explanation model g defined by Lundberg, Erion and Lee (2018) as a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i', \tag{3.1}$$

where $z' \in \{0,1\}^M$, M represents the number of input features and $\phi_i \in \mathbb{R}$ is the feature attribution values (defined above as Shapley value ϕ_i). Variables z_i generally represent a feature being observed ($z_i = 1$) or unknown ($z_i = 0$). Shapley values are computed by first defining $f_x(S) = f\left(h_x(z')\right) = E[f(x)|x_S]$ where S contains the set of non-zero indexes in z', and $E[f(x)|x_S]$ is the function's expected value on the condition of subset S of the input features. To compute each input feature's attribution, SHAP combines the conditional expectations with classic Shapley values from game theory, attributing ϕ_i values to each feature:

$$\phi_i = \sum_{S \subseteq N\{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)], \tag{3.2}$$

where N is the set of all input features. SHAP offers explanation methods suited specifically for tree-based models, linear models, and neural networks, as well as model agnostic explanations capable of explaining arbitrary ML pipelines (Lundberg, 2018).

4. Methods and Methodologies

This chapter includes description of the process and proposed fault detection methodology. Finally, the case study is presented.

4.1 Description of the Process

It is important to understand the process in terms of the structure, environment, and operation in order to develop an effective and efficient asset management program for a component.

A wind-turbine contains 20 to 25 bearings, all of which must be considered in a system-level reliability calculation of life expectancy (Froese, 2018). A typical roller bearing consists of four components as seen in Figure 2.6:

- Inner ring
- Outer ring
- Cage
- Rollers

During an operation, these components are subjected to different levels of dynamic and static loads, which can be in axial, radial or combination direction under constant or alternating conditions. These loads cause degradation of the material because of wear (contact wear – peeling, scoring, smearing, etc.), fatigue (contact fatigue – flaking, spalling, etc.), corrosion, electrical erosion, plastic deformation, and fracture & cracking (ISO, 2017), thereby resulting in the deterioration of the components and ultimately failure (Sankar, Nataraj and Raja, 2012). As the degradation progresses, it also results in changes in the behavior patterns of parameters like temperature, vibration, noise, rotational speed, etc. By monitoring these parameters using appropriate sensors, it may be possible to diagnose the health of the bearings. Commonly used parameters for identifying faults in a bearing are temperature, vibration, and noise.

4.2 Proposed Fault Detection Methodology

This section describes the proposed fault detection methodology starting with feature selection to building a model and interpreting it using SHAP.

4.2.1 Feature Selection

As discussed in the previous section, temperature is a commonly measured parameter to monitor the health of a bearing, because it is easy to continuously monitor and analyze in order to identify any abnormal behavior.

Figure 4.1 shows the simplified flowchart of heat transfers taking place in a bearing. A bearing is at a thermal equilibrium when it reaches a steady temperature. At this temperature, there is a balance between:

- **Heat generation due to bearing friction (rolling, sliding, etc.) and seal friction** – During operation, the friction among the components of a bearing results in generation of heat, the amount of which is dependent upon a number of factors, including the rotational speed, type of bearing, bearing geometry, elastic deformation under load of the rolling elements and raceways, type of lubricant and its application, and sliding friction between the components. The friction also results in wear due to an increase in bearing surface imperfections caused by pitting, cracking, etc. The formation of surface imperfections leads to an increase in friction resulting in increased heat generation. Thus, an increase in friction due to structural imperfections or deterioration in lubrication increases the temperature of bearings.
- **Conductive heat transfer from or to the adjacent parts** – Temperature of a bearing depends upon the heat input from or heat output to the adjacent parts. One piece of equipment that can significantly affect the bearing temperature is the generator itself. When the generator shaft rotates, heat is generated due to electrical resistance in the windings, resulting in heating of the generator. Since the temperature of the generator is higher than the temperature of the bearing, there is subsequently a heat transfer from the generator to the bearing. By measuring the temperature of the generator stator windings, it may be possible to estimate the effect of the generator temperature on the temperature of the bearing.
- **Convective heat dissipation to environment** – Temperature of a bearing in operation is generally above the environmental temperature, hence the bearing continuously dissipates heat to the environment. The rate of convective heat transfer is a function of:
    - **Convective heat transfer coefficient** – The convective heat transfer coefficient depends on several parameters, including the air velocity over the solid surface and the specific heat capacity of humid air. The specific heat capacity of humid air is approximately proportional to the absolute humidity of air. Thus, as the humidity increases, the value of convective heat transfer coefficient increases, resulting in an

increase in heat loss (Boukhriss, Khalifa and Ghribi, 2013). Thus, the temperature of a bearing depends upon the speed of air circulation around it and the relative humidity of air.

- **Temperature difference between the bearing and the environment** – The rate of heat loss is proportional to the difference in the temperatures of the solid (bearing) and the environment. As a result, the temperature of the bearing depends upon the ambient temperature.

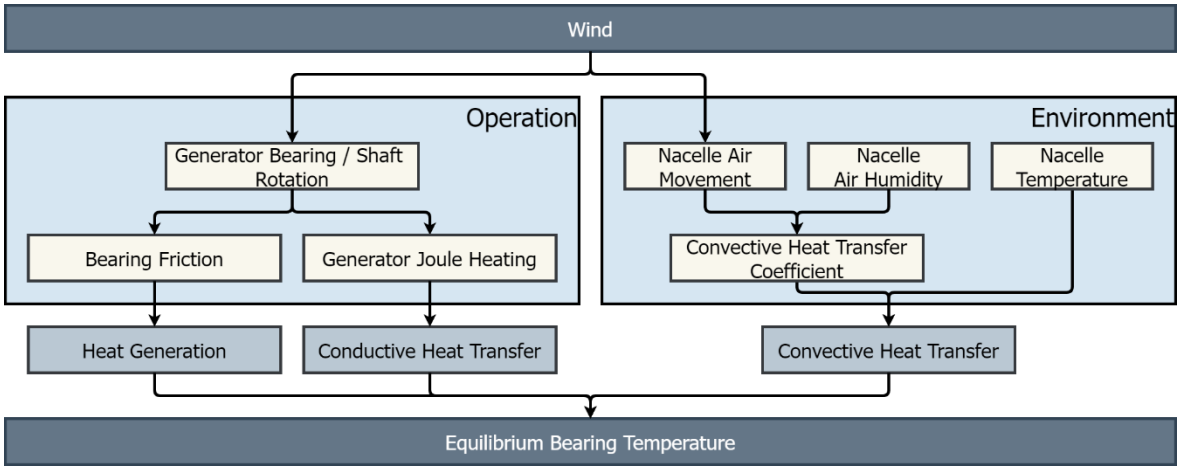

Figure 4.1: Flowchart showing the heat transfers taking place in bearings.

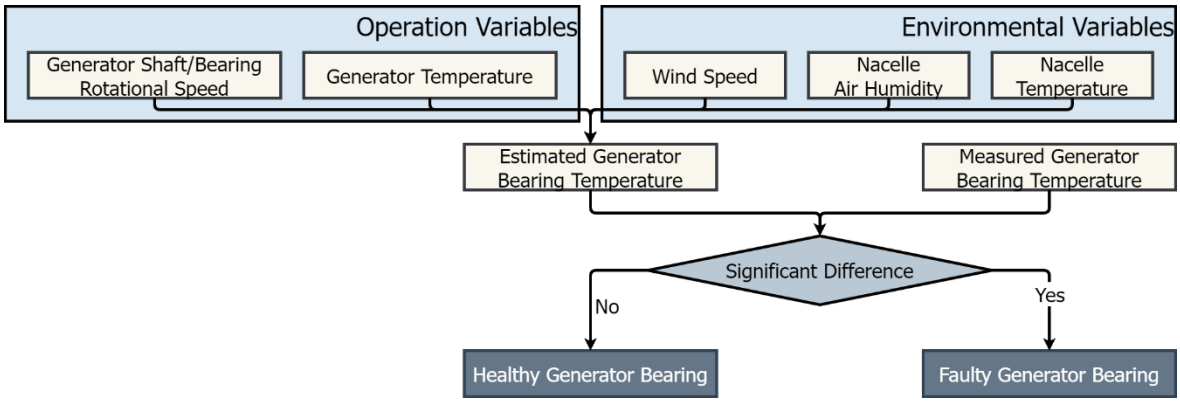

Figure 4.2: Flowchart showing the proposed fault detection methodology.

Based on the understanding of the heat transfers, five variables have been selected to predict the bearing temperature. These are:

- **Generator Shaft / Bearing Rotational Speed** – This is the rotational speed of the high-speed shaft connected to the generator. The shaft is supported by the generator bearings, and thus rotation of the shaft leads to rotation of the bearing resulting in generation of heat in the bearings due to friction.
- **Generator Temperature** – This measures the temperature of the generator stator windings. When the generator shaft rotates, heat is generated by electrical resistance in the windings. The windings are located close to the generator bearings and heat is transferred from the windings to the bearings.
- **Wind Speed** – In a WT, wind turns its rotor which in turn rotates the shaft of the generator. Hence, wind speed determines the rotational speed of the generator shaft and bearing. Additionally, since the nacelle is not airtight, the wind speed impacts air movement inside the nacelle, which in turn influences the convective heat transfer rate.
- **Nacelle Air Humidity** – This is the relative humidity of air inside the nacelle.
- **Nacelle Temperature** – This is the temperature measured in the confined space housing the WT drivetrain. The generator is located at the back of the nacelle and is therefore affected by the ambient temperature in the nacelle.

Figure 4.2 shows the flowchart of the methodology employed for detecting fault in a bearing. Using the five parameters, it may be possible to estimate temperature of a healthy bearing and if the measured temperature is above the predicted value, then there is a possibility that the higher temperature is the result of increased friction due to degradations in the bearing or lubrication.

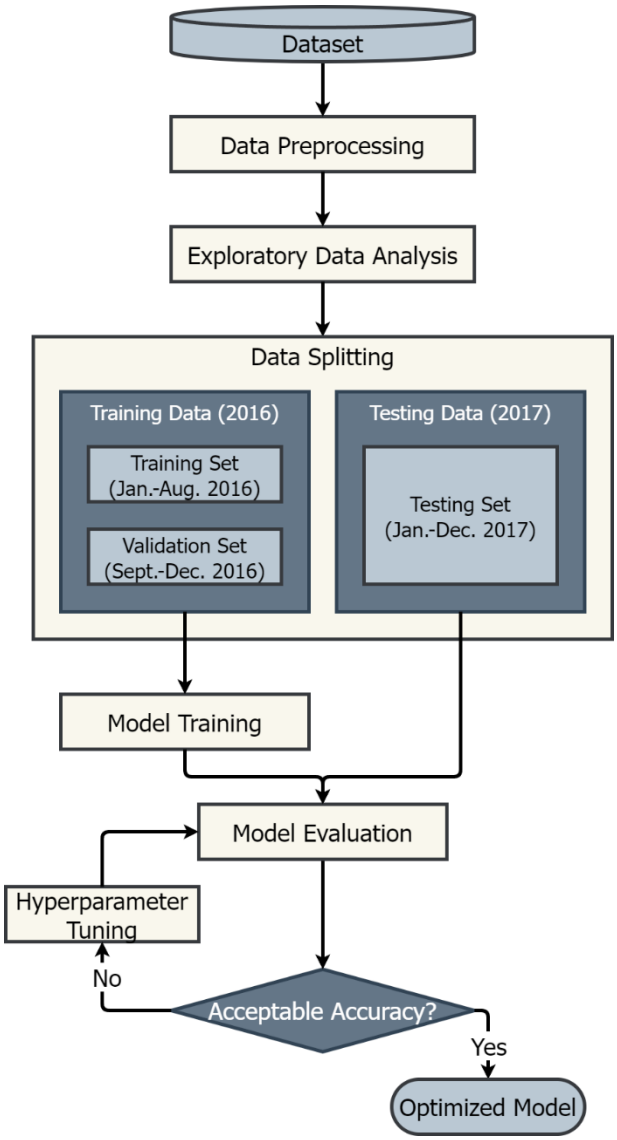

Figure 4.3: Flowchart for developing the proposed interpretable ML model.

4.2.2 Proposed Model for Predicting Bearing Temperature

As discussed in the previous section, the first step is to predict the bearing temperature using the five input variables. Figure 4.3 shows the flowchart of proposed methodology for predicting bearing temperature using ML algorithms.

Selection of Regression Algorithms

In this project a number of ML algorithms have been considered for developing a predictive model. These included:

- **Linear Models** – Linear Regression (LR), Lasso, Ridge, and Bayesian Ridge Regression.
- **Tree-based Models** – Decision Trees, Random Forest (RF).
- **Boosting Models** – AdaBoost, XGBoost and LGBoost.
- **Support Vector Machines** – Support Vector Regression (SVR).

Out of these, four algorithms – Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost – have been shortlisted for further testing. These algorithms are commonly used and are known as supervised learning algorithms, a subcategory of ML algorithms (Mahesh, 2020).

Data Preprocessing

Data preprocessing is an important step in the development of any ML model. This is because raw data is typically created, processed, and stored by a mix of humans and business processes, often resulting in imperfections like vague, inconsistent, irrational, duplicate or missing values. These imperfections need to be corrected for the algorithms to work properly. Hence, an important step in preprocessing is to identify and handle (often remove) outliers. The outliers are removed only from the training and evaluation data so that the models can be trained and evaluated on healthy turbine operation data. This improves the models' capability to detect anomalies in the test data.

Exploratory Data Analysis

Exploratory data analysis is a method of analyzing data sets to highlight their key features, frequently utilizing statistical graphics and other techniques for data visualization. Common methods include the use of Pearson, Kendall, or Spearman correlation matrices (Xiao *et al.*, 2016). These matrices depict the correlation between all the possible pairs of values and are a powerful tool to identify and visualize patterns in data.

It is important to understand the relationship between input and output SCADA signals for the model. This is established by looking into the correlation between the signals, using Pearson correlation coefficient which describes the linear correlation between two variables using their covariances and standard deviations. This method assigns a value between -1 and 1 where 0 is

no correlation, 1 is total positive correlation, and -1 is total negative correlation (Benesty *et al.*, 2009). The Pearson correlation coefficient, r_{xy}, is calculated using Equation (4.2) where n is the sample size, and x_i and y_i are the individual sample points indexed with i. $\bar{x}$ and $\bar{y}$ are the sample means of x and y, respectively, calculated using Equation (4.1).

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.1}$$

$$r_{xy} = \frac{\sum_i x_i y_i - n\bar{x}\bar{y}}{\sqrt{\sum_i x_i^2 - n\bar{x}^2}\sqrt{\sum_i y_i^2 - n\bar{y}^2}} \tag{4.2}$$

Data Splitting – Training, Validation and Testing Data

In supervised ML tasks, best practice is to split data into three independent data sets:

- **Training set** – This is the data set that is fed to the model for it to learn relationships and recognize patterns in the data.
- **Validation set** – The validation data set is used to test model performance and configuration of hyperparameters. This data set needs to be independent from the training data set so that the model does not overfit and fail to generalize.
- **Testing set** – After the validation data set is used to determine algorithm and parameter choices, the test data set is used to understand the model's performance on unseen data.

Model Training

Model training refers to the phase where an ML model is exposed to a number of training examples. The aim is to use these examples to learn patterns or relationships in the data that can later be used to make predictions on new, unseen data. During the training period, the model updates its parameters, typically through an optimization process, to minimize the prediction error made on the training examples. This error is usually measured using a loss function, which quantifies the difference between the model's predictions and the true outputs. The training period is a crucial step in the ML process, as the performance of the model on new data will depend on how well it was trained. If the model is trained well, it should be able to generalize well to new examples, meaning it will make accurate predictions even on examples it has not seen during training. If the model is overfit, it may perform well on the training data but poorly

on new data, as it has learned patterns that are specific to the training data rather than the underlying relationship between inputs and outputs.

Model Evaluation

Criteria for evaluation need to be applied in order to select the best performing algorithm out of the four. These criteria should be able to judge a model's performance regarding:

- Accuracy of prediction
- Compatibility with interpretable ML tools
- Time usage for carrying out the calculations
- Simplicity

The selection of the best model is based on an overall assessment of all the criteria.

The performances of the models are evaluated using data that has never been seen before to judge the accuracy of prediction, specifically the validation data from the data split. The models are already accustomed to the training data and find the same patterns in it as they did during training, so using the same data for testing will result in inaccurate measurements and thus receiving disproportionately high accuracy as a result. Keeping the test data separate from the training data is therefore essential in order to avoid "leakage". If test data "leaks" into the training of the model, the model will fail to generalize to data it has not seen before.

Time series cross validation is a method for evaluating the performance of ML models on time series data. In traditional cross-validation, the data is randomly divided into multiple folds, and the model is trained and evaluated on different folds, to obtain an estimate of its generalization performance. However, this approach does not consider the temporal dependencies in the time series data, which can lead to overoptimistic evaluation results.

In time series cross validation, the data is divided into folds in a way that takes into account the temporal relationships between the data points. For example, one common approach is to divide the data into contiguous chunks of time, such that each fold contains data from a different time period. This ensures that the training data for each fold is distinct from the validation data, while still capturing the temporal dependencies in the data. The use of time series cross validation is particularly important when building time series forecasting models, as the goal is to make predictions on future time points based on past observations. By properly evaluating the performance of the model on time series data, the user can have more confidence in its ability to make accurate predictions on unseen data in the future.

To evaluate the accuracy of prediction, the metrics mean absolute error (MAE), mean absolute percentage error (MAPE), root mean squared error (RMSE), and coefficient of determination (R^2) are used. MAE is the mean absolute difference between the observed data (actual values) and the model output (predicted values). The average of the absolute percentage deviations between the model output and observed values is called MAPE. MSE is the mean square errors between actual and predicted values, and RMSE is the square root of the mean squared errors. R^2 measures how closely the output values match the observed values, where a score of 1,0 is a perfect match. The equations (Pan *et al.*, 2022) for all metrics are defined below.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.3}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{4.4}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.5}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.6}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.7}$$

In the above formulas, y_i represents the actual value of the i-th sample and $\hat{y}_i$ represents the corresponding predicted value for total n samples. $\bar{y}$ represents the mean of the observed data, calculated from Equation (4.1).

Hyperparameter Tuning

Many ML algorithms require hyperparameters that need to be defined before running them. First-level model parameters are decided during training, but the second-level tuning parameters need to be tuned to optimize the performance. Typically, this is done by performing

cross-validation or evaluating predictions on a separate test set (Probst, Boulesteix and Bischl, 2019).

In this analysis, hyperparameter tuning is performed using grid search (Bergstra and Bengio, 2012) and hyperparameter values suggested by Probst, Boulesteix and Bischl (2019). This method runs through all possible combinations of the parameters within their search ranges forming a grid. It is performed using the scikit-learn library for python programming language. The grid search finally ranks all the combinations by their mean RMSE score across the same cross-validation folds used for model evaluation. Results from the grid search are used to select the optimal values for the hyperparameters.

Besides grid search there are additional hyperparameter tuning methods such as random search and Bayesian optimization (Snoek, Larochelle and Adams, 2012). Grid search is selected due to its transparency and reproducibility, as well as its robustness against local optima. By evaluating all possible combinations, it reduces the risk of getting stuck in suboptimal regions of the hyperparameter space, and hence it increases the likelihood of finding the best set of hyperparameters for a given problem.

4.2.3 Model Interpretation Using SHAP

Once the model has been tuned using optimal hyperparameters, it is ready to be interpreted. SHAP has been used to interpret outputs of the best performing ML model and quantify the impact of each feature to predictions. A negative SHAP value indicates a negative impact that decreases the value of the model output, whereas a positive SHAP value indicates a positive impact that increases the value of the model output. Although a SHAP analysis does not explicitly imply causalities, it helps in interpreting how each feature contributes to the model output and helps to identify importance of a feature in a model prediction.

4.3 Case Study

This section presents the case study performed to build a predictive ML model. The steps involve acquisition of SCADA data, preprocessing the data, and exploratory data analysis. Furthermore, the data is split into three sets, for training, evaluation, and testing. Finally, model evaluation and hyperparameter tuning is explained. All analysis is performed using Python, an object-oriented programming language, and Pandas, a software library written for Python to use for data manipulation and analysis. Pandas offers data structures and operations for manipulating numerical tables and time series. Figures are created using Seaborn, a Python data visualization library based on matplotlib, another library. It provides a high-level interface for drawing attractive and informative statistical graphics.

4.3.1 SCADA Data

To demonstrate the feasibility of the proposed methodology, supervisory control and data acquisition (SCADA) data made available by the energy company Energias de Portugal (EDP) have been used (EDP, 2017). It contains data from four 2 MW horizontal axis offshore WTs located off the west coast of Africa. These data are considered to be of good quality due to them being provided by a recognized company and recorded recently, thus representing an updated view of the technology in WTs. In addition to this, they contain a wide range of measured variables and few missing values.

The data has been recorded over a period of 2 years (2016 and 2017) at a 10-minute averaging interval. It contains values of 76 parameters. Besides this, associated data sets about meteorological conditions have also been provided for the same time instances. The parameters selected as features and target for developing the model are displayed in The generator uses two bearings, one on the drive-end and one on the driven end. From the failure log shown in Table 4.2 damage is recorded for generator bearings on August 20, 2017, at 08:08:00, and damage of generator shortly afterwards on August 21, 2017, at 16:47:00. The downtime caused by the generator failures is highlighted in green in Figure 4.4 and lasts from August 20, 2017, at 08:10:00 until August 28, 2017, at 21:50:00. The model will attempt to predict these failures.

Table 4.1. Failure logs containing timestamp, damaged component and associated remarks are also available. For this work, Turbine Number 7 (*T07*) has been selected because its failure log

has recorded generator bearing failure. For Turbine Number 7, the total number of instances is 52445 and 52294 for 2016 and 2017, respectively.

The generator uses two bearings, one on the drive-end and one on the driven end. From the failure log shown in Table 4.2 damage is recorded for generator bearings on August 20, 2017, at 08:08:00, and damage of generator shortly afterwards on August 21, 2017, at 16:47:00. The downtime caused by the generator failures is highlighted in green in Figure 4.4 and lasts from August 20, 2017, at 08:10:00 until August 28, 2017, at 21:50:00. The model will attempt to predict these failures.

Table 4.1: Selected features and target for developing the model.

Variable	Description	Unit
Timestamp	10-minute resolution	
Features		
Gen_RPM	Generator shaft / bearing rotational speed	rpm
Gen_Phase_Temp	SCADA data set gives the average temperature inside generator in stator windings Phase 1, 2 and 3. Since the temperatures are nearly the same, Gen_Phase_Temp is an average temperature of the three temperatures	°C
Wind_Speed	Ambient wind speed	m/s
Humidity	Relative nacelle air humidity	%
Nac_Temp	Nacelle temperature	°C
Target		
Gen_Bear_Temp	Temperature in generator bearing 1 (Driven End)	°C

Table 4.2: Failure log for Turbine Number 7 (T07).

Timestamp	Component	Remarks
August 20, 2017, 08:08:00	Generator bearing	Generator bearings damaged
August 21, 2017, 16:47:00	Generator	Generator damaged

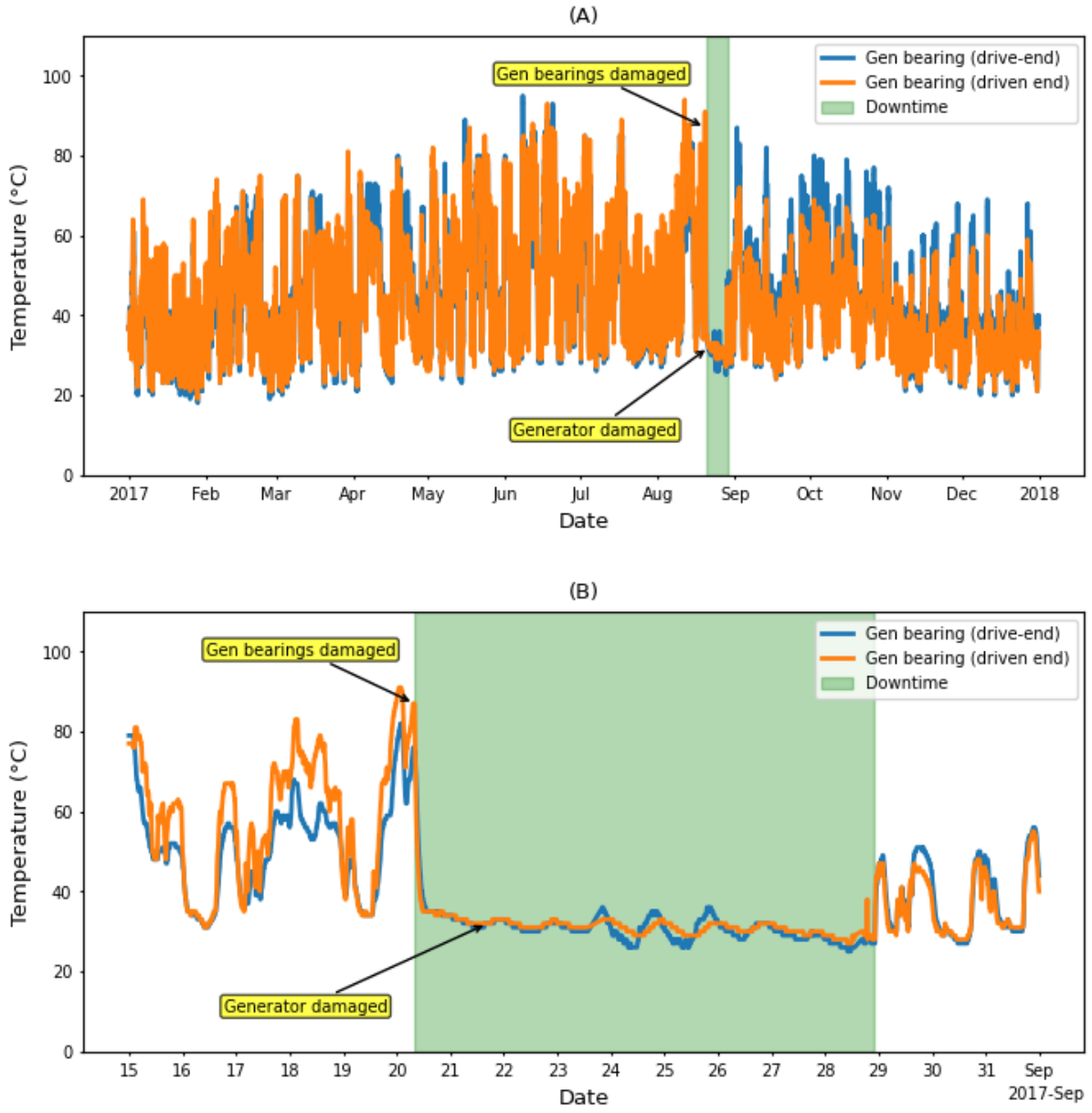

Figure 4.4: Bearings temperature during the bearing and generator failures in (A) 2017 and (B) August 2017.

4.3.2 Data Preprocessing

Identification of Data Outliers

Quite often SCADA data contains outliers that arise due to imperfections in the SCADA system and do not reflect the actual condition of process, environment, or component. For the development of a predictive model, it is important to remove these outliers because their presence can lead to biases in the model.

One common reason for outliers in the data is the inputs from faulty sensors. Since health prognosis of a bearing relies heavily on the data collected by the sensors, the reliability of analysis thus depends upon the reliability of the collected data. Hence, the reliability of results from the proposed methodology also depends upon the quality of data used for the analysis.

Figure 4.5 shows plots of the temperature data versus selected periods of the two bearings. Sudden spike in the recorded temperatures can only be due to errors in the data collection, possibly arising due to the faulty sensor. This is justified by the record showing that the sensor was replaced on April 30, 2016, at 12:40 after recording *High temperature in generator bearing 1* (driven end). Outliers like those shown in the figure need to be handled during the data preprocessing.

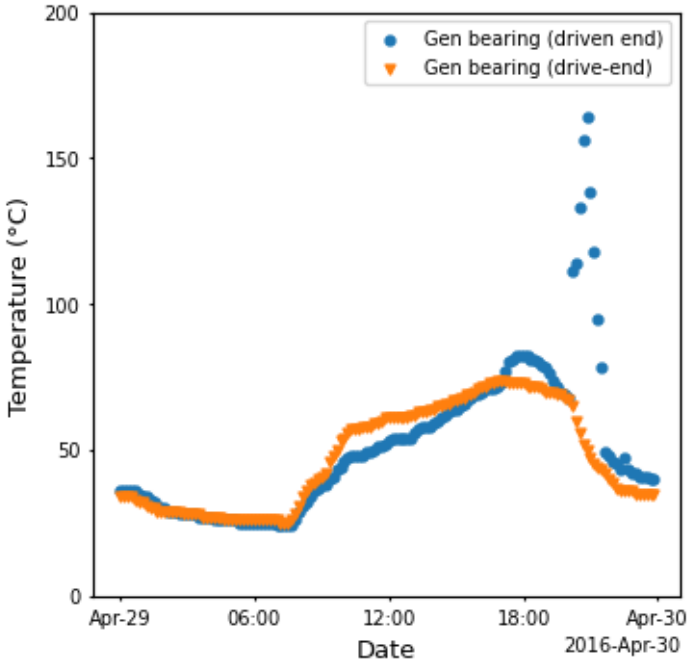

Figure 4.5: Effect of faulty sensors on recorded temperature of bearings.

In this model outliers have been identified using box plots, shown in Figure 4.6. In a box plot, the lower limit of the whisker marks the minimum value, excluding outliers, whereas the upper limit of the whisker marks the maximum value, excluding outliers. The lower limit of the box is the first quartile Q1 or the 25th percentile), whereas the upper point of the box is the third quartile (Q3 or the 75th percentile). All values within the box between Q1 and Q3, also called the interquartile range (IQR), are calculated using Equation (4.8). The horizontal red line in the

box is the median value. An outlier in this case is defined as a value outside 1,5 times the IQR above Q3 or below Q1.

$$IQR = Q3 - Q1 \tag{4.8}$$

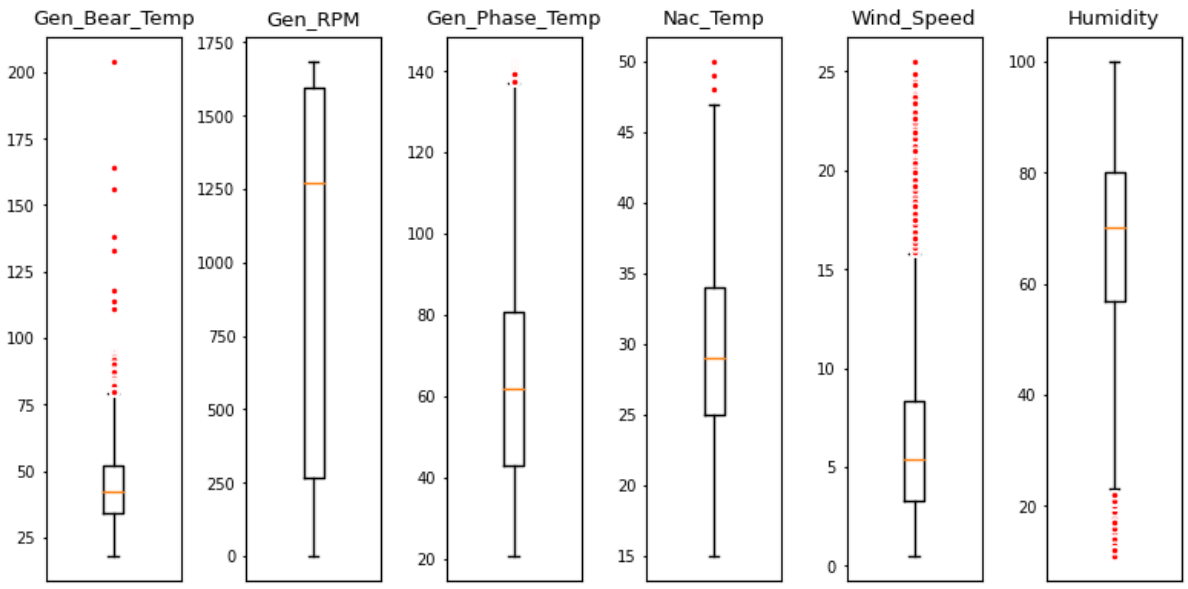

Figure 4.6: Box plot of SCADA signals.

Data Cleaning

Depending upon the characteristics of specific variables, rules for identification and handling of outliers have also been adopted. For example, a threshold of 100 ℃ has been set for the generator bearing temperature and all values higher than this have been removed. Similarly, relative humidity values are missing in the period January 3, 2017 to May 6, 2017, and this gap has been filled with values from the previous year.

Further cleaning has been performed using DBSCAN (Ester *et al.*, 1996). DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density. 'Densely clustered' data points are gathered into a single cluster.

The results before and after cleaning are shown in Figure 4.7. Figure 4.7A shows the presence of a significant number of outliers which indicate that either the turbine is not operating despite the wind blowing, or the sensors are not working properly. Additionally, there are many

instances of the turbine not operating at its maximum potential. Figure 4.7B shows the plot after the removal of the most significant outliers and the remaining data points sufficiently fit the theoretical power curve.

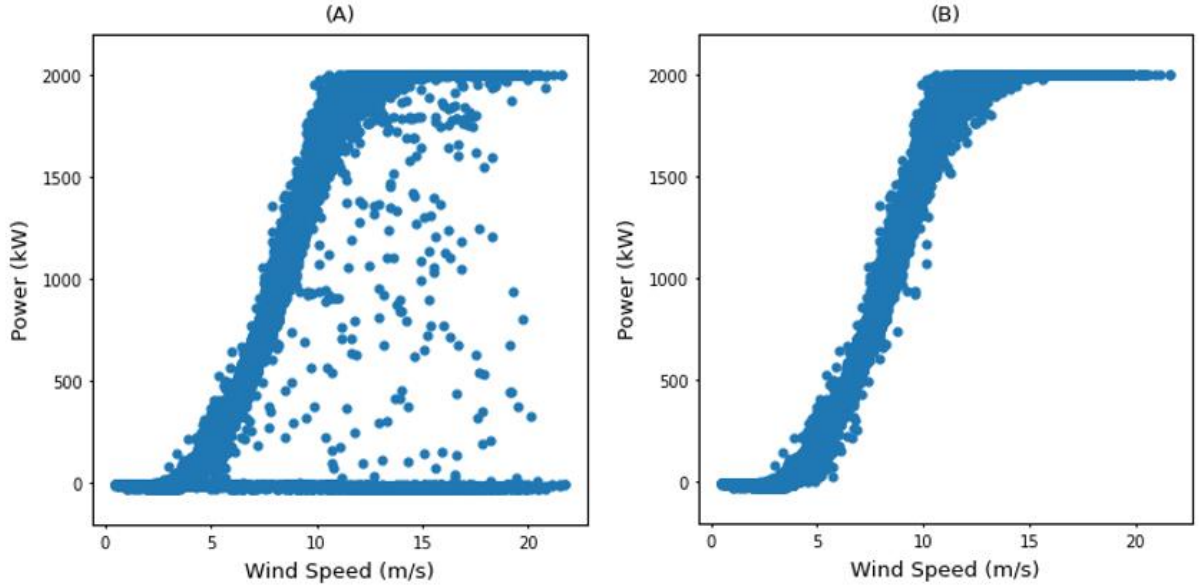

Figure 4.7: Plot of power generated versus wind speed using data of training period (A) Using raw. (B) Using data after cleaning outliers.

4.3.3 Exploratory Data Analysis

Figure 4.8 shows the Pearson correlation matrix of the input features and target. Some signals are highly correlated, for example, wind speed and generator rotational speed, wind speed and generator phase temperature, and generator phase temperature and bearing temperature. The matrix shows that the selected features are significantly relevant to the target variable.

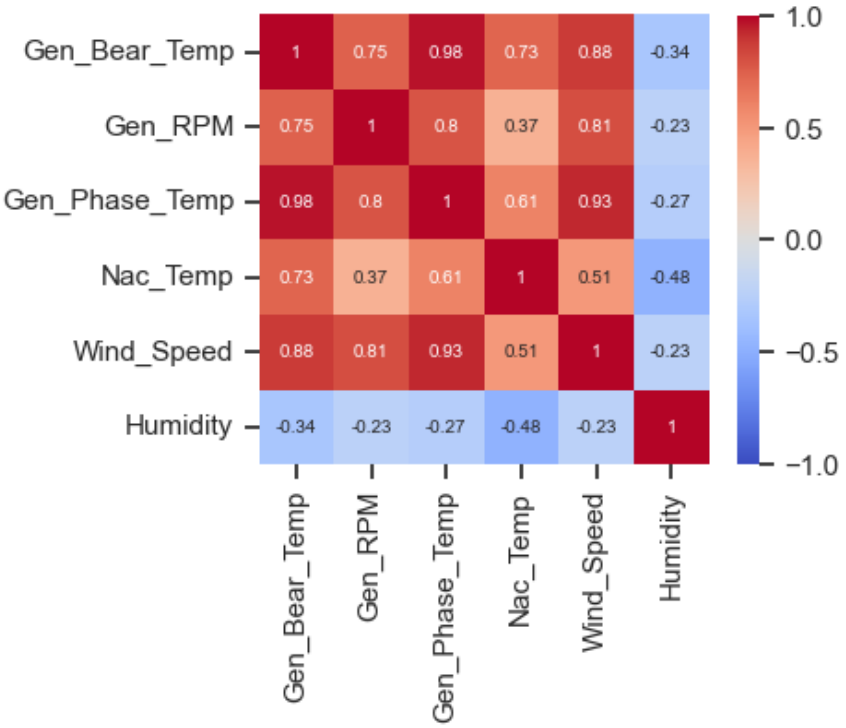

Figure 4.8: Pearson correlation matrix of the input features and target.

To further understand the correlation between the features and target, pairwise relationships between them in the training set have been plotted in Figure 4.9. The marginal histograms have been prepared by dividing signal values into 25 bins.

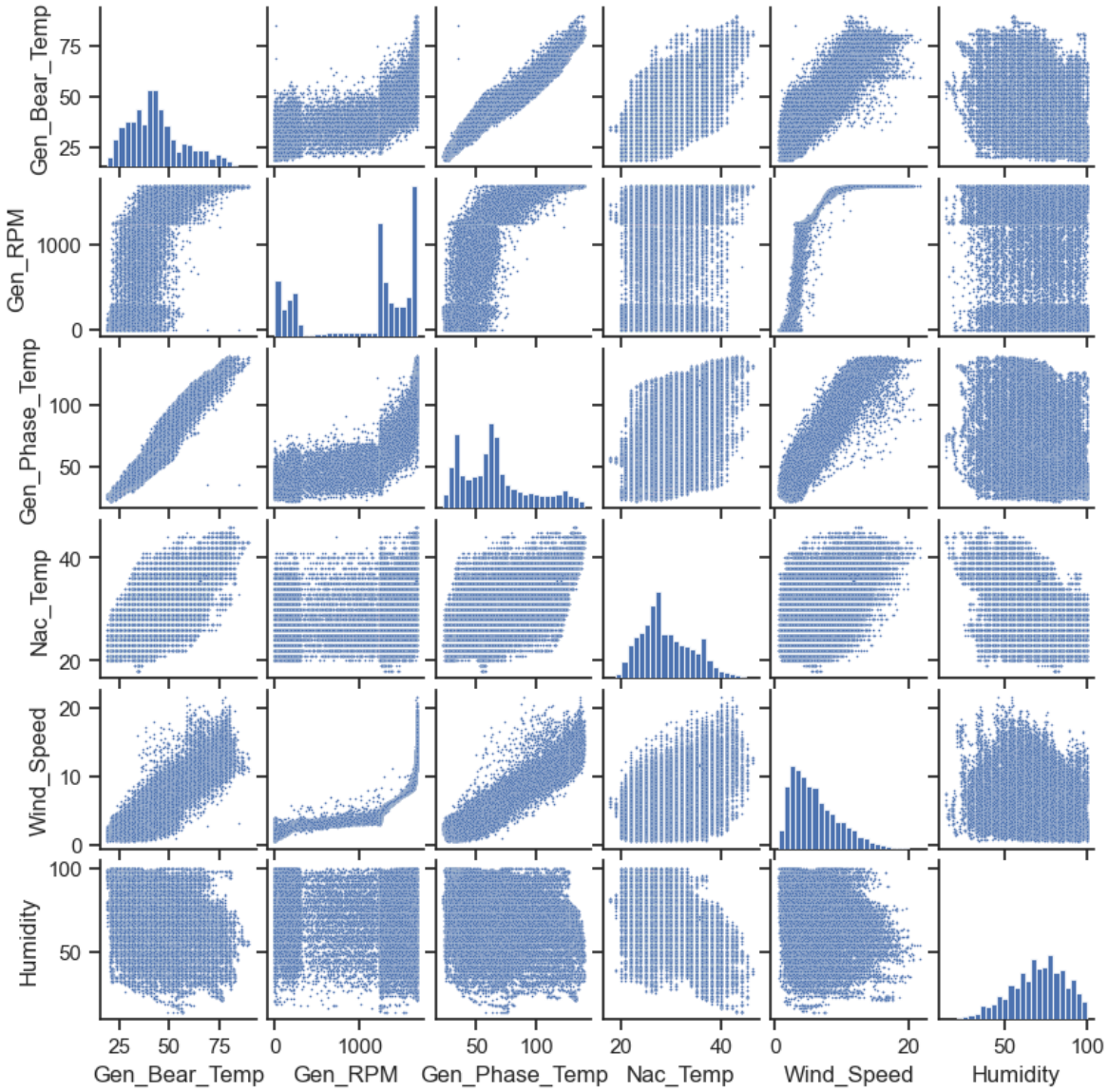

Figure 4.9: Pairwise relationships between input features and target.

Effect of Generator Shaft / Bearing Rotational Speed on Bearing Temperature

The time-averaged wear rate of a bearing can be given as (Gupta, 2013):

$$W(T) = \frac{1}{T}\frac{K}{H}\int_{o}^{T} Q(t)u(t)\,dt \tag{4.9}$$

Where:

W = Time-averaged wear rate at any time, T

K = Dimensionless wear coefficient

H = Hardness of the material being subjected to wear

Q = Contact load at any interaction at time, t

u = Sliding speed at any interaction at time, t

The equation shows the dependence of wear, W, on the parameters Q and u. Thus, the wear rate increases with an increase in the rotational speed of the generator. Corresponding to the increase in wear, the heat generated due to friction also increases with an increase in the rotational speed. This increase in heat generation manifests itself as an increase in the temperature.

Figure 4.9 shows the bearing temperature (*Gen_Bear_Temp*) is a function of the rotational speed of generator shaft / bearing (*Gen_RPM*). $r = 0{,}75$.

Effect of Generator Temperature on Bearing Temperature

In a generator, heat is produced in the windings of the stators due to the passage of electricity through the electric wiring (Joule Heating). This heat is dissipated to the surrounding through conduction and convection. A part of dissipated heat also increases the temperature of the generator bearings.

Figure 4.9 shows the approximately linear relationship between the generator temperature (*Gen_Phase_Temp*) and the bearing temperature (*Gen_Bear_Temp*). $r = 0{,}98$.

Effect of Nacelle Temperature on Bearing Temperature

The ambient temperature in the nacelle follows an annual cycle, whereby the temperature is lower during winters and higher during summers. Since the convective heat transfer is proportional to the temperature difference between a bearing's surface temperature and the ambient temperature, this variation in the ambient temperature has an effect on the heat dissipation from bearing to the environment.

Figure 4.9 shows an increase in the bearing temperature (*Gen_Bear_Temp*) with an increase in ambient temperature inside nacelle (*Nac_Temp*). $r = 0{,}73$.

Effect of Wind Speed on Bearing Temperature

Wind speed has two opposing effects on the bearing temperature. On the one hand, an increase in wind speed increases the rotational speed of bearing, resulting in increase in temperature due to friction. On the other hand, wind speed also increases air circulation within the nacelle, thereby increasing the convective heat transfer coefficient and subsequently heat loss from the bearing.

Figure 4.9 shows that there is a net increase in bearing temperature (*Gen_Bear_Temp*) with an increase in wind speed (*Wind_Speed*). $r = 0{,}88$.

Effect of Nacelle Air Humidity on Bearing Temperature

Since the specific heat capacity of humid air increases with an increase in the relative humidity of air, expectedly an increase in relative humidity increases the convective heat transfer coefficient and subsequently increases heat loss from the bearing.

Figure 4.9 shows a weak correlation between the relative humidity of air (*Humidity*) and the bearing temperature (*Gen_Bear_Temp*). $r = -0{,}34$.

4.3.4 Data Splitting – Training, Validation and Test Data

The data from 2016, after the removal of outliers, has been used for training the model in two steps. In the first step, the clean 2016 data is split into two parts – training data and validation data. The data from the first eight months is used to train the algorithms, while the data from the last four months is used to evaluate (validate) the algorithms. The validation data has been divided into four folds, each lasting for nearly a month. The initial part of the validation set is correlated with the last part of the training set. A gap of 24 hours is removed from the end of the training set close to the validation set to increase independence between training and validation data.

In the second step, the best performing model has been trained on all data in 2016 to capture any seasonal variations. Thus, the complete data set has been split into training data (33 %), validation data (17 %) and test data (50 %). The data set contains over 100 000 timestamps, and hence using only 33 % (in the first step) and 50 % (in the second step) of the data for training is sufficient. Holding out 17 % of the data for validation is in the recommended range (Belyadi and Haghighat, 2021).

4.3.5 Model Training

The four shortlisted algorithms – Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost – are trained using the training data set. For the algorithms to be evaluated on equal terms, all algorithm parameters are set to their default values during initial training.

4.3.6 Model Evaluation

In the first step, performance of the four algorithms – LR, RF, SVR and XGBoost – have been evaluated. Table 4.3 presents the RMSE scores for the four algorithms from the cross validation. The table shows that SVR has the best RMSE mean score whereas LR has the worst. The existence of almost equal RMSE values across different folds signifies that the data is evenly distributed over the time period.

Table 4.4 presents the results of the evaluation of the four models on the whole one-year test set (2017). There is a noticeable difference in the RMSE scores when the models predict a whole year compared to only the folds in the cross validation. This is due to the test set

containing faulty turbine operational data, whereas the cross validation set consists of only healthy turbine operational data similar to the training set used to learn the model. The evaluation results suggest that:

- **Linear Regression (LR)** – This has a decent score and shortest fit and prediction time.
- **Random Forest (RF)** – This has a good score but somewhat long fit time.
- **Support Vector Regression (SVR)** – This goes from top performing algorithm on the validation data to worst performing on the test data in almost all parameters, highest RMSE and longest fit and predict time.
- **XGBoost** – This scores on top while having an acceptable fit and predict time.

Table 4.3: Cross validation RMSE scores.

Model	Fold 0	Fold 1	Fold 2	Fold 3	Mean
LR	1,61	1,74	1,62	1,57	1,64
RF	1,53	1,68	1,57	1,58	1,59
SVR	1,48	1,55	1,46	1,31	1,45
XGBoost	1,48	1,74	1,48	1,51	1,55

Table 4.4: Performance of models with default parameters.

Model	MAE	MAPE	MSE	RMSE	R^2	Fit time [s]	Predict time [s]
LR	1,569	0,039	4,436	2,106	0,980	0,011	0,005
RF	1,479	0,035	3,888	1,972	0,982	18,104	0,889
SVR	1,521	0,037	4,887	2,211	0,978	90,701	188,590
XGBoost	1,436	0,034	3,824	1,955	0,983	1,266	0,019

To visualize the performance of the algorithms, plots of the predicted temperatures versus observed temperatures are shown in Figure 4.10.

- **Linear Regression (LR)** – This tends to predict rather low values.
- **Random Forest (RF)** – Along with XGBoost this appears to give the best fit.
- **Support Vector Regression (SVR)** – This predicts high values for some low bearing temperatures and low values for some high bearing temperatures.
- **XGBoost** – This appears to be the most accurate model, even though at times it predicts high values for some low bearing temperatures.

Based on the detailed evaluation, XGBoost can be considered the most suitable algorithm for the model and ready for optimization using hyperparameter tuning techniques.

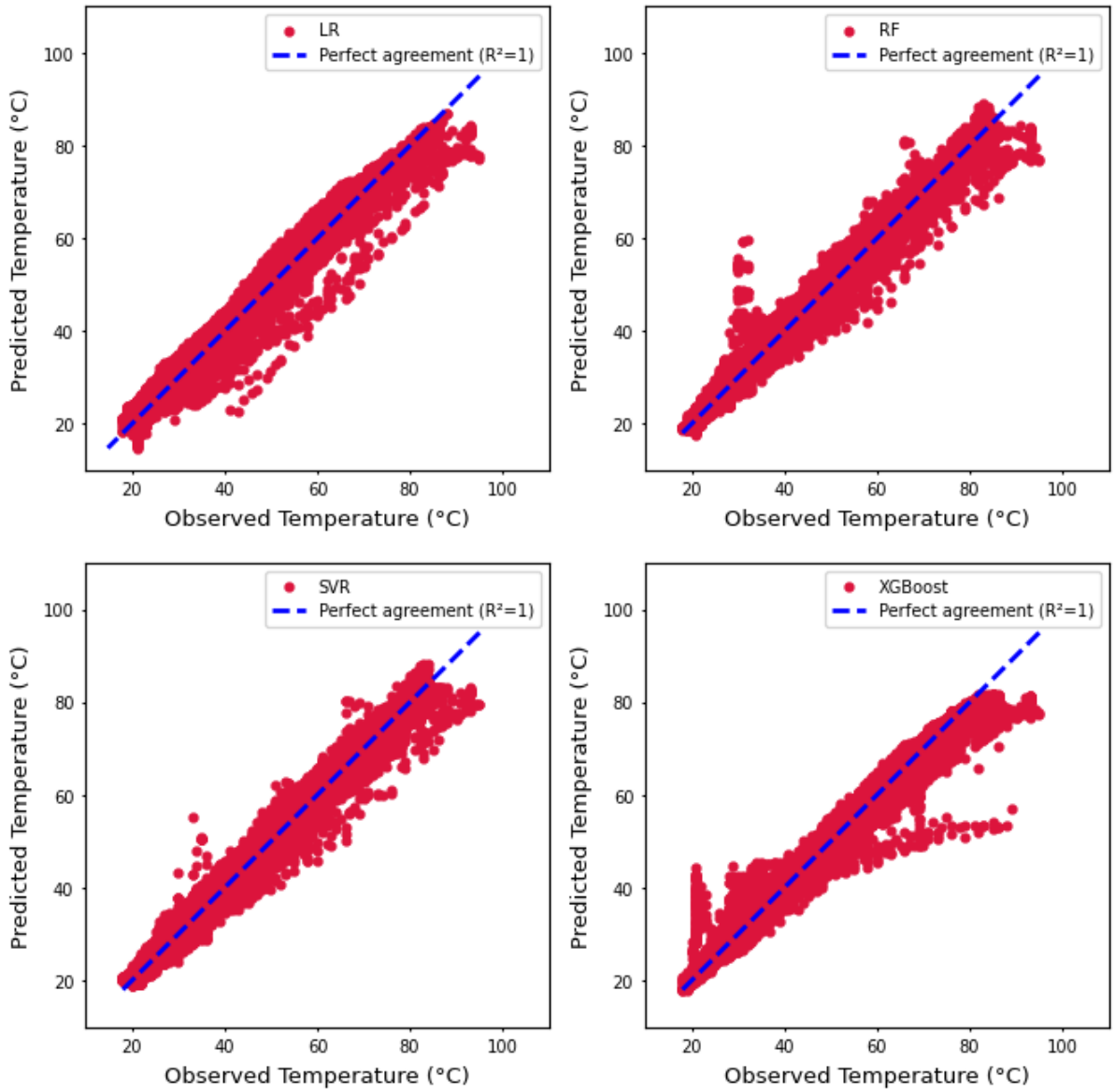

Figure 4.10: Predicted and observed temperatures for all models.

4.3.7 Hyperparameter Tuning

As described in the previous section, the XGBoost model has been selected as the most suitable model for further analysis. An important part of ML optimization is the tweaking and tuning of hyperparameters. Hyperparameter tuning is performed in the XGBoost model to enhance the model's accuracy before trying it on the test data set. The selected hyperparameters and their suggested ranges for tuning (Probst, Boulesteix and Bischl, 2019) are presented in Table 4.5. In addition to the parameters in Table 4.5, the parameters *colsample_bytree* and *colsample_bylevel* have been set to 0,6. Grid search with cross validation strategy has been performed to determine the optimal combination of hyperparameters.

Table 4.5: Hyperparameter search range.

Hyperparameter	Search range	Optimal value
n_estimators	[200, 400, 600, 800, 1000]	1000
max_depth	[3, 4, 5, 6, 7, 8, 9]	4
learning_rate	[0,1 , 0,05 , 0,01]	0,05

n_estimators is the number of runs XGBoost will try to learn, whereas *max_depth* sets the maximum depth of a tree (XGBoost, 2022). An increase in this value will make the model more complex and more likely to overfit. *learning_rate* is the shrinkage performed at every boosting step to shrink the feature weights, thus preventing overfitting. In addition to this, the parameters *colsample_bytree* and *colsample_bylevel* are set to 0,6. *colsample_bytree* assigns the subsample ratio of columns when constructing each tree. Subsampling occurs once for every tree constructed. A value of 0,6 results in three of the five features being used per tree constructed. *colsample_bylevel* assigns the subsample ratio of columns for each level in a tree. For every new depth level in a tree, a subsampling takes place. A value of 0,6 results in two of the three features continuing on to the next level, and since the model rounds up, the remaining subsampling uses these same two features. This results in not using the whole training set every time but building a tree on slightly different data at each step, which makes it less likely to overfit to a single sample or feature.

The grid search method runs through all possible combinations of the parameters in the grid in Table 4.5. The search ranges give a total of 105 candidates which are tested on the model, and including the four cross-validation folds, it results in 420 unique fits. It is a time-consuming process which makes it important to limit the number of parameters for tuning as well as the search range of each parameter. Results from the grid search are displayed in Figure 4.11, and

compared to *max_depth*, it is shown that *learning_rate* and *n_estimators* have more effect on performance of the algorithm in terms of RMSE, MAE and R^2. The optimal values of these parameters are given in Table 4.5.

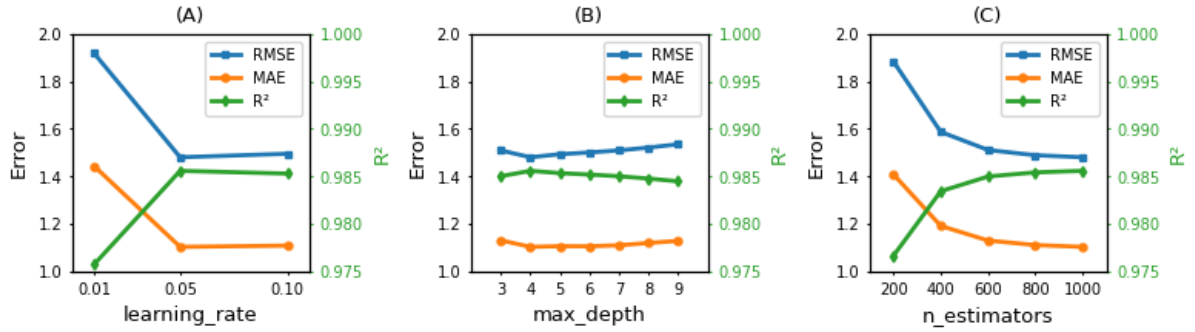

Figure 4.11: Model impact changing (A) learning_rate, (B) max_depth and (C) n_estimators.

Table 4.6 shows the performance of XGBoost algorithm after hyperparameter tuning using the optimized parameter values given in Table 4.5. As shown, there is an improvement in the performance of the algorithm after hyperparameter tuning.

Table 4.6: Optimized XGBoost performance on test data and validation data.

Test Data Performance					
Model	**MAE**	**MAPE**	**MSE**	**RMSE**	**R^2**
XGBoost	1,436	0,034	3,824	1,955	0,983
Optimized XGBoost	1,389	0,033	3,354	1,832	0,985
Change [%]	3,272	2,941	12,291	6,292	0,203
Validation Data Performance [RMSE]					
Model	**Fold 0**	**Fold 1**	**Fold 2**	**Fold 3**	**Mean**
XGBoost	1,48	1,74	1,48	1,51	1,55
Optimized XGBoost	1,41	1,65	1,44	1,40	1,48
Change [%]	4,73	5,17	2,70	7,29	4,52

5. Results and Discussion

5.1 Prediction of Generator Bearing Temperature

The optimized XGBoost algorithm-based model has been used to predict generator bearing temperature using the test data (2017).

Figure 5.1 shows the plot of the actual and predicted values for the period January 1 to January 15, 2017, the curves of which are for:

- Actual temperature
- Predicted temperature
- Predicted ±2 standard deviation temperature (3,56 ℃)

The figure shows that the actual temperature remains within the (predicted ±2 standard deviation) temperature range.

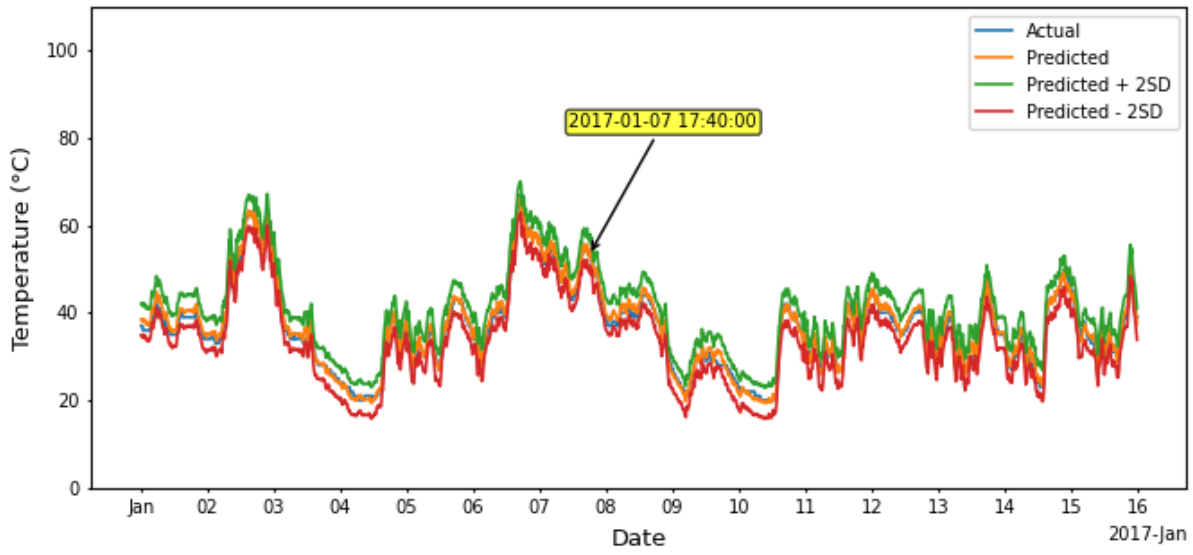

Figure 5.1: Actual and predicted temperatures of generator bearing for the period January 1 to January 15, 2017.

5.2 Sources of Error

Data sets will always have some inaccuracies that may affect their quality. It is therefore important to understand why errors exist in the data and how they impact results. This section contains an analysis of the data quality and its impact on the model.

Inaccuracies in the output results may arise due to high correlations between feature and target variables which may impact how the ML model learns. This risk is partly mitigated by using hyperparameters *colsample_bytree* and *colsample_bylevel*. Other sources of error include faulty sensors which can explain the recorded high generator bearing temperatures and the missing humidity data. This can be due to wrong calibration or drift in calibration of sensors.

In the case study there may be additional sources of errors, such as replacing the missing humidity data with the values from the previous year. This includes values as low as 11 % relative humidity (RH), which is low considering that the WT is located out in the ocean on the equator. Meteorological data from São Tomé and Príncipe, an island some hundred kilometers away, show records of a minimum of 37 % RH on September 12, 2017, the same day as the 11 % RH was measured (Time and Date, 2017). The uncertainty of the validity of these low measured humidity values may impact the error of the model. Additionally, the nacelle and rotor blades not rotating according to the wind direction may also be a source of error.

5.3 Fault Detection and Recommendation for Rescheduling Maintenance Plan

Figure 5.2 shows the plot of the actual and predicted values for the period from June 7 to June 23, 2017. During this period there are times when the actual bearing temperature exceeds the predicted value by more than two standard deviations (3,56 °C) over significantly long periods, and this is highlighted in green. For example, on June 7, 2017 the actual value reaches 95 °C whereas the model prediction is 76 °C, a difference of 19 °C.

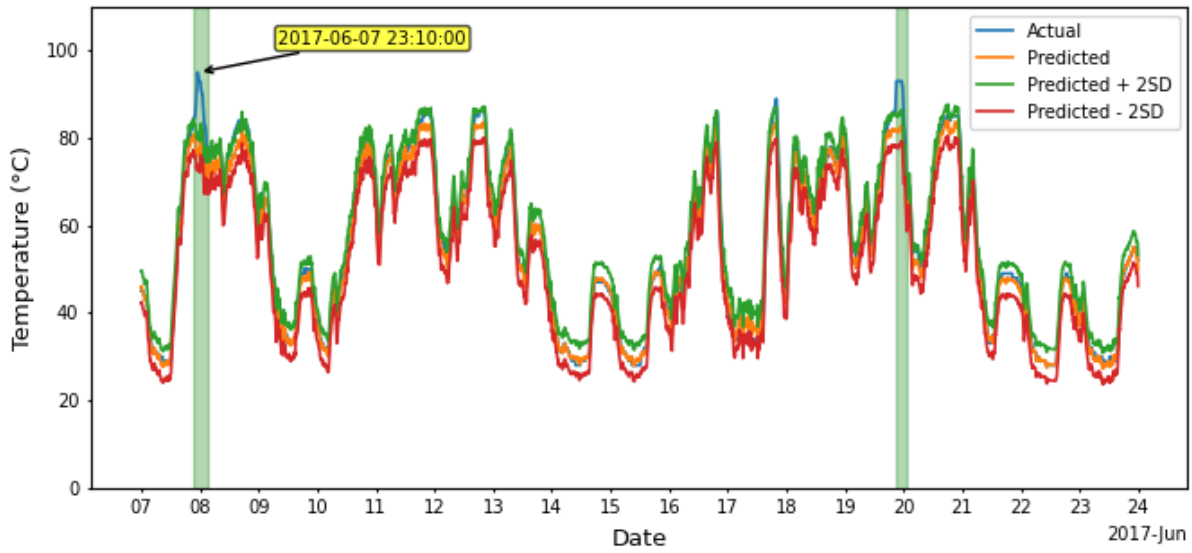

Figure 5.2: Actual and predicted temperatures of generator bearing for the period June 7 to June 23, 2017.

After June 7, 2017, there is a tendency for the actual bearing temperature to be higher than the predicted bearing temperature. At times it often crosses the two standard deviation limit. This indicates two possibilities:

- Malfunctioning of the bearing sensor.
- Possibility that the bearing is getting hotter than expected perhaps due to increased friction. The increased friction could be either because of increased wear or improper lubrication. Both of these possibilities warrant special inspection and monitoring activity.

Based on the detection of faulty bearing, recommendation may be made for scheduling maintenance activities at the earliest opportunity. This recommendation is justified by the fact that the bearing breaks down two months later on August 20, 2017.

5.4 Model Interpretation Using SHAP

The XGBoost algorithm-based model used for the case study gives reasonably good predictions for the temperature of a generator bearing. The model needs to be further evaluated to interpret how it is working. Since XGBoost is a tree-based model, the Tree SHAP algorithm proposed

by Lundberg, Erion and Lee (2018) for tree ensembles can be used to calculate the SHAP values that could be used for the interpretation of how the model is working.

5.4.1 Global Explanations

Figure 5.3A shows the mean absolute SHAP values for the used features. The figure shows that:

- The generator phase temperature has by far the highest impact on the model predictions. This is reasonable due to the adjacent location of the bearing and generator.
- Nacelle temperature and wind speed have moderate average impact on the model predictions, which should be expected since the convective heat loss from bearing is directly proportional to the difference in temperature between the bearing and the nacelle temperature. Wind speed affects not only the rotational speed but also the convective heat loss.
- Generator or bearing rotational speed and relative humidity have low impact.

Figure 5.3B shows the changes in the SHAP value for changes in the feature value. For all features except the humidity, a higher feature value has a positive impact on the model prediction, and a low feature value has a negative impact on the model output. As is to be expected, the humidity has the opposite impact for its feature values, because increase in humidity increases the specific heat capacity of air, resulting in higher convective heat loss from the bearing and a decrease in temperature.

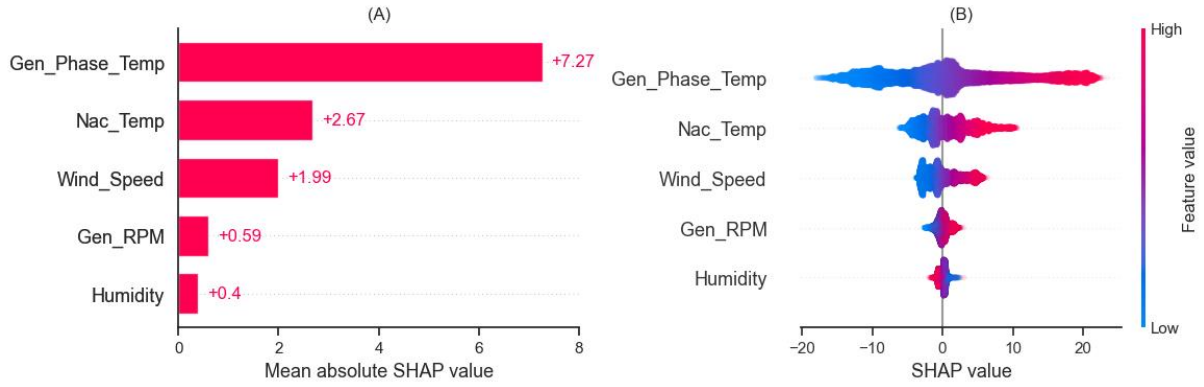

Figure 5.3: (A) Mean absolute SHAP value per feature. (B) Matrix plot of SHAP values for different features.

SHAP treats each feature as a "player", hence there are interaction effects between features. The SHAP main effect plots in Figure 5.4 remove all interaction effects between features and thus display the raw impact of each feature. The figure shows that:

- **Generator Shaft / Bearing Rotational Speed** – Generator rotational speed has a low impact with a small positive spike near its max rotation speed.
- **Generator Temperature** – The generator phase temperature has a dominant and nearly linear impact on the model output.
- **Nacelle Temperature** – Nacelle temperature has an increased positive impact in the temperature range 20-45 °C.
- **Wind Speed** – At the cut-in wind speed of 4 m/s, there is a marked increase in the impact of wind speed. It increases up until the rated wind speed of 12 m/s and from there on stays constant.
- **Nacelle Air Humidity** – The impact of humidity is rather weak and decreases slowly across its range.

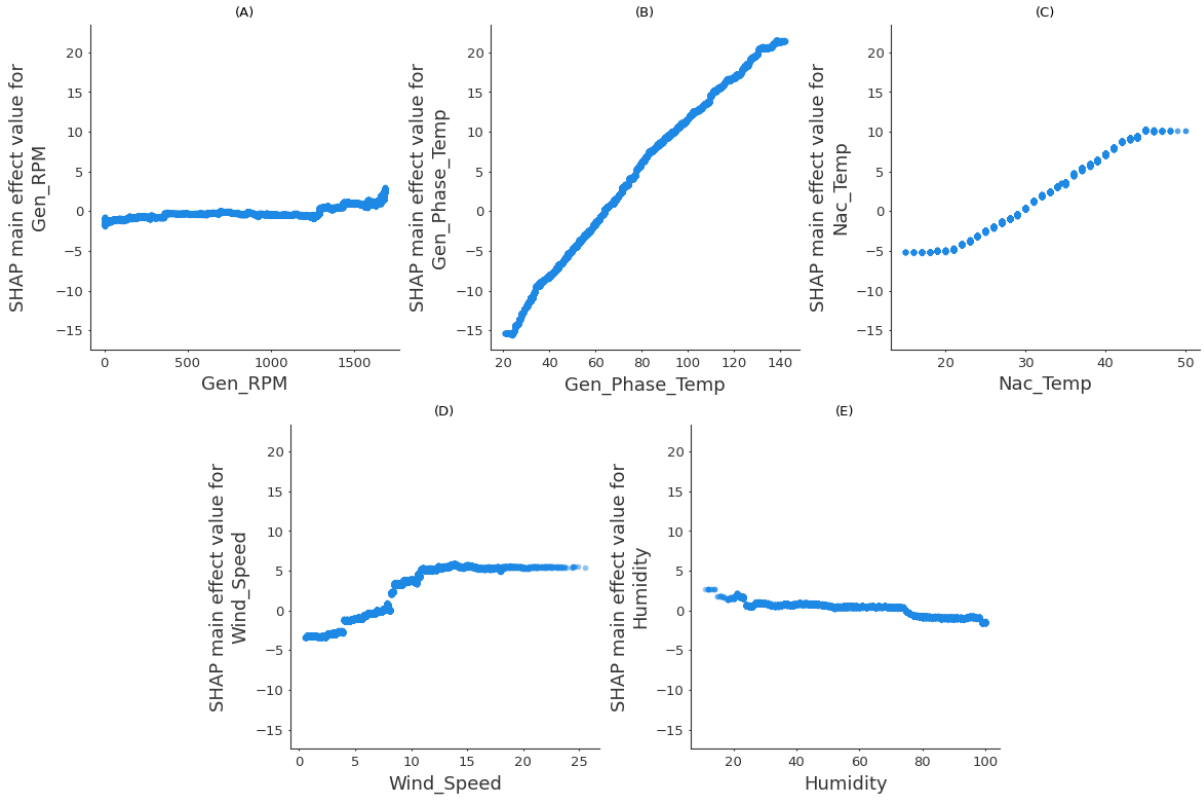

Figure 5.4: SHAP main effects plot for (A) generator rpm, (B) generator phase temperature, (C) nacelle temperature, (D) wind speed and (E) humidity.

5.4.2 Local Explanations

SHAP waterfall plots are used for explaining individual predictions. Starting from the expected value of the model output (the average prediction of the model on the training data) at the bottom of the waterfall plot, each row shows the contribution of each feature to the model output for a prediction. A positive (red) contribution moves the initial output value higher, whereas a negative (blue) contribution moves the initial output value lower.

Explanation of Prediction for January 7, 2017

Figure 5.1 shows the plots of the actual and predicted values for the period of January 1 to January 15, 2017. During this period all predicted values are within two standard deviations of the actual value, indicating a possibility that the bearing is operating normally. From this period, an instance (January 7, 2017, at 17:40:00) has been randomly selected for local explanation.

According to Figure 5.4, the bearing temperature is influenced most by the generator temperature because of its high temperature and proximity to the bearing. This is followed by the nacelle temperature and wind speed. The generator rotational speed and humidity have relatively minor effect.

On January 7, 2017, at 17:40:00 the actual generator bearing temperature is 53 °C. The SHAP waterfall plot in Figure 5.5 explains how the XGBoost model arrived at a prediction of 54 °C.

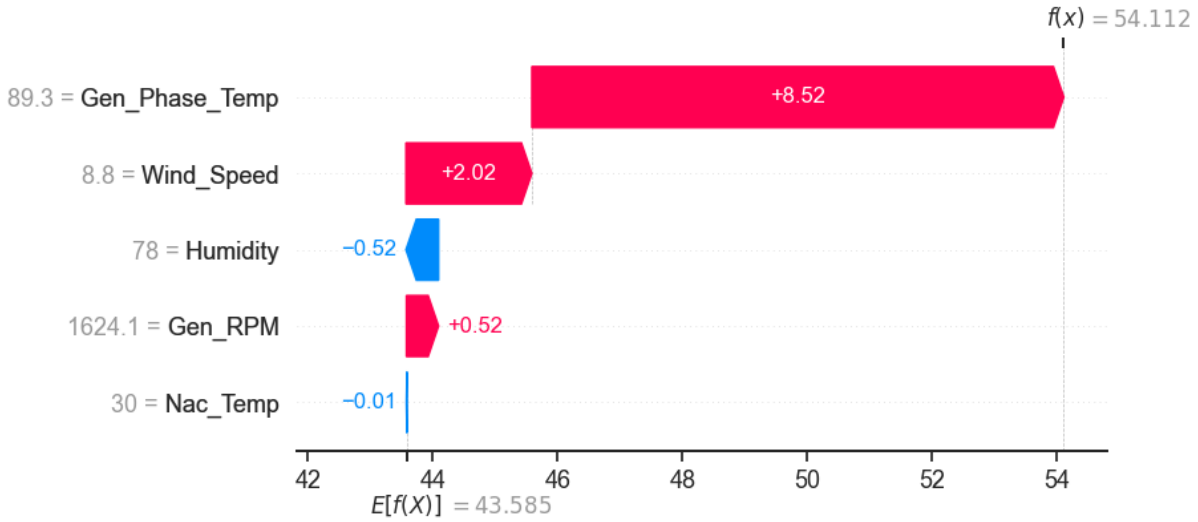

Figure 5.5: Local explanation on January 7, 2017, at 17:40:00 by waterfall plot.

- **Generator Shaft / Bearing Rotational Speed** – Rotational speed has minor effect on the predicted temperature value, hence the net heating effect on the predicted bearing temperature (+0,52 ℃) is relatively small.
- **Generator Temperature** – The high generator phase temperature (89,3 ℃) has by far the most significant positive influence (+8,52 ℃) on the bearing temperature.
- **Wind Speed** – Wind speed makes relatively small positive effect (+2,02 ℃) on the predicted value. Wind speed has two opposing effects – increase in temperature due to increased friction and decrease in temperature due to increased convective heat loss. In this case the rotational speed has small effect (+0,52 ℃) and hence a greater positive effect may be due to the interaction between the wind speed, the generator temperature, and the bearing temperature.
- **Nacelle Air Humidity** – The high relative humidity (78 %) also does not significantly (-0,52 ℃) affect the predicted temperature value, because relative humidity itself does not have any significant role.
- **Nacelle Temperature** – The nacelle temperature (30 ℃) is close to the average annual temperature, ranging between 15-50 ℃, and hence does not play a significant role (-0,01 ℃) in the fall of temperature on predicted value.

5.4.3 Explanation of Prediction for June 7, 2017

Figure 5.2 shows the plots of the actual and predicted values for the period June 7 to June 23, 2017. On June 7, 2017 (Summer), the environmental and operating temperatures are quite different from those of January 7, 2017 (Winter). Based on the SHAP waterfall plot (Figure 5.6**)** an attempt is made to explain the working of the model.

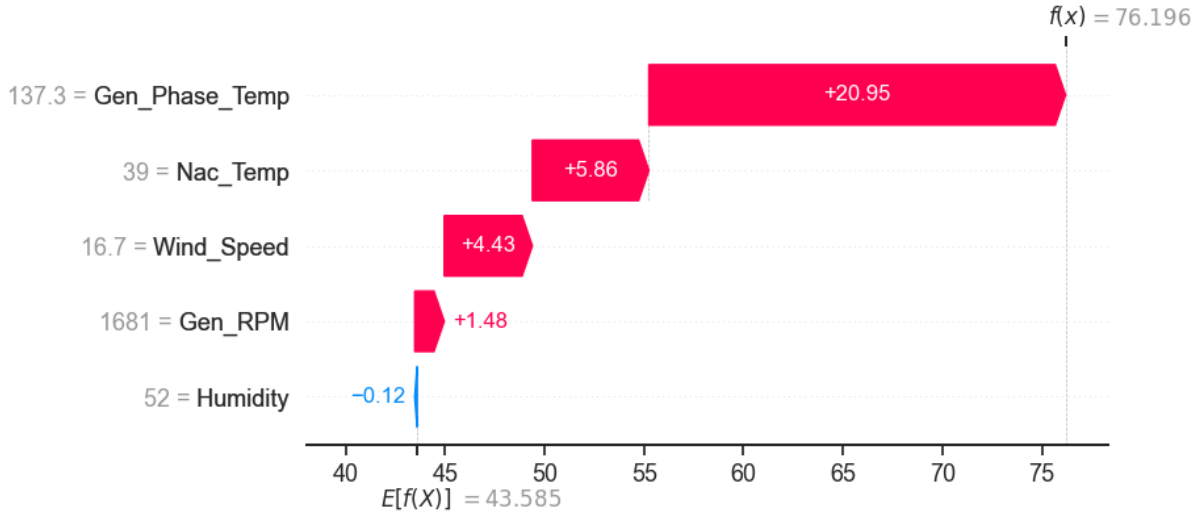

Figure 5.6: Local explanation on June 7, 2017, at 23:10:00 by waterfall plot.

- **Generator Shaft / Bearing Rotational Speed** – As in the previous case (January 7, 2017), the rotational speed has a minor effect on the predicted temperature value, and hence the net heating effect on the predicted bearing temperature (+1,48 ℃) is relatively small. The small increase could be due to the small positive spike that appears near its max rotation speed (Figure 5.4A).
- **Generator Temperature** – The generator temperature is very high (137,3 ℃), and this significantly (+20,95 ℃) raises the temperature of the bearing.
- **Wind Speed** – Compared to the previous case, wind speed gives relatively higher positive effect (+4,43 ℃) on the predicted value. This may be because of higher interaction between the wind speed, the generator temperature, and the bearing temperature.
- **Nacelle Air Humidity** – As in the previous case, nacelle relative humidity has negligible (-0,12 ℃) effect on the predicted temperature value.
- **Nacelle Temperature** – Compared to the previous case, the nacelle temperature (39 ℃) is 9 ℃ higher than the previous case, and hence there is significantly (+5,86 ℃) higher effect on the predicted temperature.

The analysis provides a reasonable explanation for the predicted bearing temperature. A high generator temperature (137 ℃) increases the predicted bearing temperature significantly (+20,95 ℃) and the remaining features also contribute to bringing the predicted bearing temperature to 76,2 ℃.

6. Conclusion

This thesis presents a simple and robust methodology for making a machine learning (ML) based model for detecting faults in a wind turbine generator bearing. In this model, the predicted bearing temperature is compared against the actual bearing temperature and a significant difference between the two indicates a possibility of fault(s) in the bearing and/or deterioration of lubrication. Either of these may result in failure. As a case study, the idea has been demonstrated on a generator bearing, using real-life supervisory control and data acquisition (SCADA) data from a 2 MW wind turbine. The results show that it is possible to detect potential failure well in advance. This knowledge can be used for planning maintenance.

Four different ML algorithms, Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost, have been evaluated based on criteria accuracy, compatibility with interpretable ML tools, time usage, and simplicity. XGBoost has been found to be the most suitable algorithm for the task and is further optimized using hyperparameter tuning.

The thesis also examines the role of five features, generator shaft / bearing rotational speed, generator temperature, wind speed, nacelle air humidity, and nacelle temperature, on the predicted bearing temperature. This was performed using Shapley additive explanations (SHAP), an approach to explain the output of any machine learning model. Out of the five features, the generator temperature has been found to play the major role, followed by the wind speed and nacelle temperature. Bearing rotational speed and relative humidity of nacelle air play minor roles.

7. Suggestions for Further Work

To take the research work further, the following tasks have been identified:

- Analysis of data from different WTs.
- Imputing missing data using other methods.
- Testing of other ML/artificial intelligence algorithms, like artificial neural networks.
- Consideration of the impact of more features.
- Use of other interpretable ML tools such as Individual Conditional Expectation (ICE) plots (Goldstein *et al.*, 2015) and LIME (Local interpretable model-agnostic explanations (LIME) (Ribeiro, Singh and Guestrin, 2016).
- Expanding the scope from component to system level.

8. References

Adadi, A. and Berrada, M. (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI), *IEEE Access*, 6, pp. 52138-52160. doi: 10.1109/ACCESS.2018.2870052.

Artigao, E. *et al.* (2018) Wind turbine reliability: A comprehensive review towards effective condition monitoring development, *Applied energy*, 228, pp. 1569-1583. doi: 10.1016/j.apenergy.2018.07.037.

Astolfi, D. (2023) Wind Turbine Drivetrain Condition Monitoring through SCADA-Collected Temperature Data: Discussion of Selected Recent Papers, *Energies*, 16(9), pp. 3614. doi: 10.3390/en16093614.

Badurek, C. A. (2023) Wind turbine *Encyclopedia Britannica.* Available at: https://www.britannica.com/technology/wind-turbine (Accessed: 28.03.2023).

Belyadi, H. and Haghighat, A. (2021) *Machine Learning Guide for Oil and Gas Using Python: A Step-by-Step Breakdown with Data, Algorithms, Codes, and Applications*. Gulf Professional Publishing.

Benesty, J. *et al.* (2009) Pearson Correlation Coefficient *Noise reduction in speech processing.* Springer, Berlin, Heidelberg.

Bergstra, J. and Bengio, Y. (2012) Random search for hyper-parameter optimization, *Journal of machine learning research*, 13(2). Available at: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

Bhatia, S. C. (2014) 8 - Wind energy, in Bhatia, S. C. (ed.) *Advanced Renewable Energy Systems.* Woodhead Publishing India, pp. 184-222.

Bilderzwerg (2022) *Wind power generator english vector illustration* [Image]. Available at: https://stock.adobe.com/no/images/wind-power-generator-english-vector-illustration/39580742.

Binderszewsky, J. (2014) Components of rolling bearings pp. 1. Available at: https://www.semanticscholar.org/paper/Analysis-of-Dynamically-Loaded-Rolling-Bearing-with-Binderszewsky/06eec9e98f15b73ab22978fb96741a076b47a640/figure/0.

Bindingsbø, O. T. *et al.* (2023) Fault Detection of a Wind Turbine Generator Bearing Using Interpretable Machine Learning, Submitted for publication in *Engineering Failure Analysis*.

Borges, V. (2023) *Opportunistic maintenance*. Available at: https://www.dnv.com/article/opportunistic-maintenance-207906 (Accessed: 28.03.2023).

Boukhriss, M., Khalifa, Z. and Ghribi, R. (2013) Study of thermophysical properties of a solar desalination system using solar energy, *Desalination and Water Treatment*, 51, pp. 1290-1295. doi: 10.1080/19443994.2012.714925.

Breiman, L. (2001) Random forests, *Machine learning*, 45, pp. 5-32.

Carroll, J., McDonald, A. and McMillan, D. (2016) Failure rate, repair time and unscheduled O&M cost analysis of offshore wind turbines, *Wind Energy*, 19, pp. 1107-1119. doi: 10.1002/we.1887.

Chen, T. and Guestrin, C. (2016) Xgboost: A scalable tree boosting system, *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. pp. 785-794.

Choi, R. Y. *et al*. (2020) Introduction to machine learning, neural networks, and deep learning, *Translational vision science & technology*, 9(2), pp. 14-14.

Chouinard, J. C. (2023) *Decision Tree* [Image]. Available at: https://www.jcchouinard.com/decision-trees-in-machine-learning/.

Circuit Globe (2023) *Synchronous Generators*. Available at: https://circuitglobe.com/synchronous-generators.html (Accessed: 28.03.2023).

Cole, S. (2023) *Wind turbine power output variation with steady wind speed* [Image]. Available at: https://theroundup.org/wind-turbine-power-curve/.

Cui, Y., Bangalore, P. and Tjernberg, L. B. (2018) An Anomaly Detection Approach Based on Machine Learning and SCADA Data for Condition Monitoring of Wind Turbines, *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), 24-28 June 2018*. pp. 1-6.

DNV (2022) *Definitions of Availability Terms for the Wind Industry*. (DNV-WP-15). Available at: https://www.dnv.com/Publications/definitions-of-availability-terms-for-the-wind-industry-98846 (Accessed: 28.03.2023).

EDP (2017) *Data*. Available at: https://www.edp.com/en/innovation/open-data/data (Accessed: 14.06.2023).

EERE (2022a) *How a Wind Turbine Works - Text Version*. Available at: https://www.energy.gov/eere/wind/how-wind-turbine-works-text-version (Accessed: 30.10.2022).

EERE (2022b) *How Do Wind Turbines Survive Severe Storms?* Available at: https://www.energy.gov/eere/articles/how-do-wind-turbines-survive-severe-storms (Accessed: 14.11.2022).

EIA (2022) *Types of wind turbines*. Available at: https://www.eia.gov/energyexplained/wind/types-of-wind-turbines.php (Accessed: 31.11.2022).

Ekanayake, I., Meddage, D. and Rathnayake, U. (2022) A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP), *Case Studies in Construction Materials*, 16. doi: 10.1016/j.cscm.2022.e01059.

Encalada-Dávila, Á. *et al.* (2021) Wind turbine main bearing fault prognosis based solely on scada data, *Sensors*, 21(6), pp. 2228. doi: 10.3390/s21062228.

Equinor (2022) *Floating wind*. Available at: https://www.equinor.com/energy/floating-wind (Accessed: 28.03.2023).

Ester, M. *et al.* (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Second International Conference on Knowledge Discovery and Data Mining (KDD'96). Proceedings of a conference held August 2-4*. pp. 226-231.

Faulstich, S., Lyding, P. and Hahn, B. (2010) *Component reliability ranking with respect to WT concept and external environmental conditions*. (UpWind Deliverable WP7.3.3, WP7 Condition monitoring). Kassel (Germany): Fraunhofer IWES

Faulstich, S., Hahn, B. and Tavner, P. J. (2011) Wind turbine downtime and its importance for offshore deployment, *Wind Energy*, 14(3), pp. 327-337. doi: 10.1002/we.421.

Froese, M. (2018) *Why wind-turbine gearboxes fail to hit the 20-year mark.* Available at: https://www.windpowerengineering.com/wind-turbine-gearboxes-fail-hit-20-year-mark/ (Accessed: 30.07.2023).

Fu, F. (2018) Chapter Eight - Design of Offshore Structures, in Fu, F. (ed.) *Design and Analysis of Tall and Complex Structures.* Butterworth-Heinemann, pp. 251-293.

GE (2023) *Haliade-X offshore wind turbine*. Available at: https://www.ge.com/renewableenergy/wind-energy/offshore-wind/haliade-x-offshore-turbine (Accessed: 28.05.2023).

Goldstein, A. *et al.* (2015) Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation, *journal of Computational and Graphical Statistics*, 24(1), pp. 44-65. doi: 10.1080/10618600.2014.907095.

Gupta, P. K. (2013) Analytical Modeling of Rolling Bearings, in Wang, Q. J. and Chung, Y.-W. (ed.) *Encyclopedia of Tribology.* Boston, MA: Springer US, pp. 72-80.

Hansen, T. (2020) *Havvind - enkelt forklart*. Available at: https://www.uib.no/sites/w3.uib.no/files/attachments/havvind_enkelt_forklart.pdf (Accessed: 28.03.2023).

Hart, E. *et al.* (2020) A review of wind turbine main bearings: design, operation, modelling, damage mechanisms and fault detection, *Wind Energy Science*, 5(1), pp. 105-124. doi: 10.5194/wes-5-105-2020.

Hofstad, K. and Rosvold, K. A. (2022) Vindkraft *Store norske leksikon.* Available at: https://snl.no/vindkraft (Accessed: 28.03.2023).

IEA (2021) *Net Zero by 2050*. Paris. Available at: https://www.iea.org/reports/net-zero-by-2050 (Accessed: 26.09.2022).

IRENA (2016) Offshore wind floating foundation concepts, pp. 2. Available at: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2016/IRENA_Offshore_Wind_Floating_Foundations_2016.pdf.

IRENA (2018) Fixed-bottom foundations used in the offshore wind industry, pp. 40. Available at: https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2018/May/IRENA_Leveraging_for_Offshore_Wind_2018.pdf.

IRENA (2022) *Renewable power generation costs in 2020*. eBook Partnership.

ISO (2017) *ISO 15243:2017 Rolling bearings — Damage and failures — Terms, characteristics and causes*. Available at: https://www.iso.org/standard/59619.html (Accessed: 13.06.2023).

Jin, X., Xu, Z. and Qiao, W. (2021) Condition Monitoring of Wind Turbine Generators Using SCADA Data Analysis, *IEEE Transactions on Sustainable Energy*, 12(1), pp. 202-210. doi: 10.1109/TSTE.2020.2989220.

Karmouche, S. (2016) Different tower designs for small wind turbines, pp. 16. Available at: http://www.aui.ma/sse-capstone-repository/pdf/spring2016/Tower%20Design%20And%20Analysis%20For%20A%20Small%20Wind%20Turbine.pdf.

Korkos, P. *et al.* (2022) Data annotation and feature extraction in fault detection in a wind turbine hydraulic pitch system, *Renewable Energy*, 185, pp. 692-703. doi: 10.1016/j.renene.2021.12.047.

Lantz, E. J. *et al.* (2019) *Increasing Wind Turbine Tower Heights: Opportunities and Challenges*. Available at: https://doi.org/10.2172/1515397.

Letcher, T. M. (2017) *Wind energy engineering: A handbook for onshore and offshore wind turbines*. Academic Press.

Liu, R. *et al.* (2018) Comparative analysis of bearing current in wind turbine generators, *Energies*, 11(5), pp. 1305.

Liu, Z. and Zhang, L. (2020) A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings, *Measurement*, 149. doi: 10.1016/j.measurement.2019.107002.

Lundberg, S. (2018) *Tabular examples*. Available at: https://shap.readthedocs.io/en/latest/tabular_examples.html#tree-based-models (Accessed: 28.05.2023).

Lundberg, S. M. and Lee, S.-I. (2017) A unified approach to interpreting model predictions, *Advances in neural information processing systems*, 30. doi: 10.48550/arXiv.1705.07874.

Lundberg, S. M., Erion, G. G. and Lee, S.-I. (2018) Consistent individualized feature attribution for tree ensembles, *arXiv e-prints*. doi: 10.48550/arXiv.1802.03888.

Mahesh, B. (2020) Machine Learning Algorithms - A review, *International Journal of Science and Research (IJSR)*, 9(1), pp. 381-386. Available at: https://www.ijsr.net/archive/v9i1/ART20203995.pdf.

Manwell, J. F., McGowan, J. G. and Rogers, A. L. (2009) *Wind energy explained: theory, design and application*. 2 edn. John Wiley & Sons.

Marietta (2023) *Types of wind turbines* [Image]. Available at: https://c03.apogee.net/mvc/home/hes/land/el?utilityname=mp&spc=kids&id=16214.

Matha, D., Lemmer, F. and Muskulus, M. (2019) Offshore turbines with bottom-fixed or floating substructures *Wind Energy Modeling and Simulation - Volume 2: Turbine and System.* Institution of Engineering and Technology, pp. 125-167. doi: 10.1049/PBPO125G_ch5.

Mendes, M. *et al.* (2020) Wind Farm and Resource Datasets: A Comprehensive Survey and Overview, *Energies*, 13. doi: 10.3390/en13184702.

Miceli, F. (2012) *Power curve: what is it and how to measure it*. Available at: https://www.windfarmbop.com/power-curve-what-is-and-how-to-measure-it/ (Accessed: 28.03.2023).

Molnar, C., Casalicchio, G. and Bischl, B. (2020) Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges, in Koprinska, I., *et al.* (ed.) *ECML PKDD 2020 Workshops*, *Cham*. Springer International Publishing, pp. 417-431.

Montgomery, D. C., Peck, E. A. and Vining, G. G. (2021) *Introduction to linear regression analysis*. John Wiley & Sons.

Mæhlum, L. and Rosvold, K. A. (2019) Vindmølle. Available at: https://snl.no/vindm%C3%B8lle (Accessed: 06.08.2023).

National Institute of Standards and Technology (2023) *Supervisory Control and Data Acquisition (SCADA)*. Available at: https://csrc.nist.gov/glossary/term/supervisory_control_and_data_acquisition (Accessed: 24.03.2023).

NORSOK (2017) *Risk based maintenance and consequence classification* (NORSOK Z-008:2017). Available at: https://www.standard.no/no/Nettbutikk/produktkatalogen/Produktpresentasjon/?ProductID=956000.

Pan, S. *et al.* (2022) An optimized XGBoost method for predicting reservoir porosity using petrophysical logs, *Journal of Petroleum Science and Engineering*, 208. doi: 10.1016/j.petrol.2021.109520.

Probst, P., Boulesteix, A.-L. and Bischl, B. (2019) Tunability: Importance of hyperparameters of machine learning algorithms, *The Journal of Machine Learning Research*, 20(1), pp. 1934-1965. doi: 10.48550/arXiv.1802.09596.

Rasheed, A., San, O. and Kvamsdal, T. (2020) Digital twin: Values, challenges and enablers from a modeling perspective, *IEEE Access*, 8, pp. 21980-22012. doi: 10.1109/ACCESS.2020.2970143.

Ren, Z. *et al.* (2021) Offshore wind turbine operations and maintenance: A state-of-the-art review, *Renewable and Sustainable Energy Reviews*, 144. doi: 10.1016/j.rser.2021.110886.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) *" Why should i trust you?" Explaining the predictions of any classifier*. Unpublished paper presented at Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA (United States).

Salameh, J. P. *et al.* (2018) Gearbox condition monitoring in wind turbines: A review, *Mechanical Systems and Signal Processing*, 111, pp. 251-264. doi: 10.1016/j.ymssp.2018.03.052.

Sankar, S., Nataraj, M. and Raja, P. V. (2012) Failure analysis of bearing in wind turbine generator gearbox, *Journal of Information Systems and Communication*, 3(1), pp. 302.

Shafiee, M. (2015) Maintenance logistics organization for offshore wind energy: Current progress and future perspectives, *Renewable Energy*, 77, pp. 182-193. doi: 10.1016/j.renene.2014.11.045.

Shafiee, M., Finkelstein, M. and Bérenguer, C. (2015) An opportunistic condition-based maintenance policy for offshore wind turbine blades subjected to degradation and environmental shocks, *Reliability Engineering & System Safety*, 142, pp. 463-471.

Sheng, S. (2013) *Report on wind turbine subsystem reliability-a survey of various databases (presentation)*. (NREL/PR-5000-59111). Golden, CO (United States): National Renewable Energy Laboratory. Available at: https://www.nrel.gov/docs/fy13osti/59111.pdf.

Smola, A. J. and Schölkopf, B. (2004) A tutorial on support vector regression, *Statistics and computing*, 14, pp. 199-222.

Snoek, J., Larochelle, H. and Adams, R. P. (2012) Practical bayesian optimization of machine learning algorithms, *Advances in neural information processing systems*, 25.

Stehly, T., Beiter, P. and Duffy, P. (2020) *2019 cost of wind energy review*. (NREL/TP-5000-78471). Golden, CO (United States): National Renewable Energy Laboratory. Available at: https://www.nrel.gov/docs/fy21osti/78471.pdf.

Stetco, A. *et al.* (2019) Machine learning methods for wind turbine condition monitoring: A review, *Renewable Energy*, 133, pp. 620-635. doi: 10.1016/j.renene.2018.10.047.

Thomsen, O. T. (2009) Sandwich Materials for Wind Turbine Blades - Present and Future, *Journal of Sandwich Structures & Materials*, 11(1), pp. 7-26. doi: 10.1177/1099636208099710.

Tidemann, A. and Elster, A. C. (2022) Maskinlæring *Store norske leksikon.* Available at: https://snl.no/maskinl%C3%A6ring (Accessed: 28.03.2023).

Time and Date (2017) *Past Weather in São Tomé, Sao Tome and Principe - September 2017*. Available at: https://www.timeanddate.com/weather/sao-tome-and-principe/sao-tome/historic?month=9&year=2017 (Accessed: 28.03.2023).

Tran, H. N. (2021) *A conceptual design of a wind turbine gearbox* [Image]. Available at: https://grabcad.com/library/wind-turbine-gearbox-1.

Varghese, A. J., Roy, R. and Awasthi, S. R. (2022) Performance Evaluation of HAWT-and VAWT-Based WECS with Advanced Hill Climb Search MPPT and Fuzzy Logic Controller for Low Wind Speed Regions, in Dawn, S., *et al.* (ed.) *Smart and Intelligent Systems.* Singapore: Springer Singapore, pp. 97-113.

Vilone, G. and Longo, L. (2020) Explainable artificial intelligence: a systematic review, *arXiv preprint*. doi: 10.48550/arXiv.2006.00093.

Whittle, M. (2013) *Wind turbine generator reliability: an exploration of the root causes of generator bearing failures*. PhD Thesis, Durham University. Available at: http://etheses.dur.ac.uk/9422/.

Wilkinson, M. *et al.* (2010) *Methodology and results of the reliawind reliability field study*. Unpublished paper presented at European Wind Energy Conference, EWEC 2010. Warsaw, Poland.

Wind Europe (2023) *Wind energy today*. Available at: https://windeurope.org/about-wind/wind-energy-today (Accessed: 28.03.2023).

WindFacts (2009) *Detailed loss factors*. Available at: https://www.wind-energy-the-facts.org/detailed-loss-factors.html (Accessed: 14.11.2022).

XGBoost (2022) *XGBoost Documentation*. Available at: https://xgboost.readthedocs.io/en/stable/ (Accessed: 28.03.2023).

Xiao, C. *et al.* (2016) Using Spearman's correlation coefficients for exploratory data analysis on big dataset, *Concurrency and Computation: Practice and Experience*, 28(14), pp. 3866-3878.

Appendix 1

Fault Detection of a Wind Turbine Generator Bearing Using Interpretable Machine Learning

Oliver Trygve Bindingsbø[1], Maneesh Singh[1], Knut Øvsthus[1], Arvind Keprate[2]

1. *Western Norway University of Applied Sciences, Bergen, Norway.*
2. *OsloMet, Norway*

Abstract

During its operational lifetime, a wind turbine is subjected to a number of degradation mechanisms. If left unattended, the degradation of components will result in its suboptimal performance and eventual failure. Hence, to mitigate the risk of failures, it is imperative that the wind turbine be regularly monitored, inspected, and optimally maintained.

Offshore wind turbines are normally inspected and maintained at fixed intervals (generally six-month intervals) and the program (list of tasks) is prepared using experience or risk-reliability analysis, like Risk-based inspection (RBI) and Reliability-centered maintenance (RCM). This time-based maintenance program can be improved upon by incorporating results from condition monitoring involving data collection using sensors and fault detection using data analytics. In order to properly carry out condition assessment, it is important to assure quality & quantity of data and to use correct procedures for interpretation of data for fault detection.

This paper discusses the work carried out to develop a machine learning based methodology for detecting faults in a wind turbine generator bearing. Explanation of the working of the machine learning model has also been discussed in detail. The methodology includes application of machine learning model using SCADA data for predicting operating temperature of a healthy bearing; and then comparing the predicted bearing temperature against the actual bearing temperature. Consistent abnormal differences between predicted and actual temperatures may be attributed to the degradation and presence of a fault in the bearing. This fault detection can then be used for rescheduling the maintenance tasks. The working of this methodology is discussed in detail using a case study.

Keywords: Bearing, condition monitoring, fault detection, machine learning, offshore wind turbine, SCADA, SHAP

1 INTRODUCTION

In order to meet the increasing demand for energy and yet reduce dependency on conventional fossil fuels, there has been a spurt in growth of wind farms [IEA, 2021]. These wind farms are comprised of arrays of wind turbines (typically horizontal), installed either onshore or offshore, to produce electricity from the wind. However, despite recent advances in the design, manufacturing, operation and maintenance of wind turbines, their acceptance has been muted due to a number of reasons, including difficulties and high costs associated with their operation and maintenance.

When compared to the onshore wind turbines, the offshore counterparts offer more reliable power generation due to higher mean wind speeds and more steady wind supply. Unfortunately, the operation and maintenance difficulties and costs are also higher due to multiple reasons, including faster degradation of equipment by harsh marine conditions, difficulties in accessing the site from distant shores, rough weather conditions, scarcity of skilled personnel and need for specialized vessels. Thus, the operation and maintenance costs account for approximately a third of the Levelized Cost of Energy (LCOE) [Wiggelinkhuizen et al., 2007; Stehly, Beiter and Duffy, 2020].

During their operational lifetime, various components of a wind turbine are subjected to a number of environmental & operational attacks resulting in their degradation. This degradation results in deterioration in performance and at times failure. Failure of a component takes place when the applied load is greater than the maximum safe working load of the component. The applied load and maximum safe working load of the component vary with time. The applied load can vary due to the changes in the operating conditions, environmental conditions or accident; and the maximum safe working load may change with time due to degradation caused to the component by different types of degradation mechanisms. Hence, it becomes difficult to predict when the failure will take place [Arabian-Hoseynabadi, et al. (2010), Kahrobaee, S. and Asgarpoor, S. (2011), Shafiee and Dinmohammadi (2014), Luengo and Kolios (2015), Zhang et al. (2016)].

To help in predicting the time of failure, detailed failure analysis involving the following stages needs to be carried out [Kandukuri et al., 2016]:

- *Fault Detection* – detection of abnormal changes in the structure or behavior of a component that can help to identify faulty condition
- *Fault Diagnosis* – analysis of the abnormal changes in the structure or behavior to identify cause or mechanism of the degradation that would cause the failure
- *Fault Quantification* – analysis of the behavior or performance to quantify the degree of degradation and fault (partial or complete)
- *Fault Prognosis* – analysis of the time-based changes to predict the outcome of further degradation or prognosis of fault

Failure (or fault) analysis can be used to develop detailed failure profiles (failure causes, failure mechanisms, etc.), which can subsequently be used for developing an appropriate maintenance schedule to prevent or manage the failure. In a maintenance schedule, the maintenance activities can be either preventive or corrective depending on whether the task is carried out before or after failure. These maintenance activities involve detailed inspection (visual, auditory, NDT), testing, service (lubrication, cleaning, repair, etc.), repair and replacement tasks.

The preventive maintenance programs are often time-based, for example, preventive maintenance activities of wind turbines are normally planned at 6-month intervals [Nilsson and Bertling, 2007]. Since these time-based inspection and maintenance programs are expensive, there have been efforts to develop methodologies for preparing more efficient and effective maintenance programs. This involves development of maintenance schedules based on formalized risk/reliability analysis (e.g., Risk Based Inspection and Maintenance or Reliability Centered Maintenance).

In order to improve the technical asset integrity management of wind farms there is an increasing move towards condition-based maintenance as opposed to scheduled or reactive maintenance to reduce downtime and lost production. This is achieved by (a) continuous monitoring using sensors; (b) data analytics; and (c) developing condition-based maintenance plans.

To continuously monitor, all modern wind turbines come with a Supervisory Control and Data Acquisition (SCADA) system. This system is comprised of a multitude of sensors that constantly monitor various parameters regarding environment, process, operation, and condition of components (equipment or structure). The data from the sensors is transmitted and stored in SCADA supervisory computers. At the control office the data is interpreted, and the information gained is then used to control the process or operation. The same data can be used to develop optimized condition-based maintenance schedules.

Since selection of sensors is an important aspect of the concept, this is carried out judiciously and a number of factors are considered such as feasibility, cost, and benefits of measurements. Having decided the need for monitoring, the next stage is to decide the type, quantity and quality of data required to satisfy the needs of the application, scope of the assessment, and expected level of detail (or acceptable uncertainty), etc.

The selected sensors may be installed permanently (e.g., thermocouples, tachometers, rotation angle sensor), semi-permanently, or as portable measuring instruments (example, handheld vibration sensor, thermal cameras). For use in wind turbines, a number of condition monitoring techniques are commercially available. Since each technique is suitable within a specific domain, a combination of different techniques needs to be applied [Tchakoua et al., 2014].

While the collection, transmission and storage of data has become relatively easy in recent years, the challenge is to identify and extract relevant information from the available data. Thus, sensible data collection requires understanding the system, making decisions related to collection and rationalization of data to make it suitable for further analysis, and finally, to use the preprocessed data to extract useful information, like, fault detection and identification, so that necessary decisions can be taken. There are a number of approaches by which the data analysis can be carried out, to include machine learning, fuzzy logic, artificial neural networks, and deep learning.

Machine learning techniques have been widely explored for analyzing data from offshore wind turbines and these have been found to be suitable for detecting anomalies and assisting in decision-making [Stetco, et al., 2019]. However, while machine learning models may have high prediction accuracy, they often lack interpretability. This is because models often act as black-boxes, thereby making their results challenging to understand and interpret, and users may not have knowledge of the underlying decisions in the predicting process [Ekanayake, Meddage and Rathnayake, 2022].

Interpretable machine learning tools can be applied to gain insight into the working of machine learning models. Thus, it is easier to understand the factors that drive their predictions and increase confidence in their predictions. This understanding may be used to justify the use of the model and to further improve its working [Adadi and Berrada, 2018]. Interpretable machine learning is currently at a stage where it is sufficiently developed and mature, but there are still some challenges that need to be addressed [Molnar, Casalicchio and Bischl, 2020, Vilone and Longo, 2020].

In recent years, the research community has become more interested in Shapley additive explanations (SHAP) method, which proposes a model agnostic representation of feature importance estimated by Shapley values in a computationally efficient manner. Shapley values are a solution concept from collaborative game theory. The SHAP method is an additive feature attribution method that considers the features as "the players", combinations of different features as "the coalitions", and the prediction as "the total payout". The average marginal contribution for feature i over all possible coalitions is the Shapley value ϕ_i, hence it explains each feature's contribution to a prediction [Lundberg and Lee, 2017; Lundberg, Erion and Lee 2018].

Besides SHAP there are other methods for interpreting machine learning results such as Individual Conditional Expectation (ICE) plots [Goldstein et al., 2015] and Local interpretable model-agnostic explanations (LIME) [Ribeiro, Singh and Guestrin, 2016]. ICE plots visualize the dependence of model predictions on a feature for each instance separately. By varying the values of a feature for a particular instance while keeping the values of all other features fixed, it shows the relationship between the feature and the model's predictions across a range of values by repeating this process. Each line in the ICE plot represents the predicted outcome for a different instance, allowing us to see the individual effects of a feature on the model's predictions. LIME works by approximating the machine learning model locally around a specific instance, using a simpler, interpretable model. It perturbs the instance, creates a dataset, fits an interpretable model on the perturbed instances, and generates explanations based on the model's feature weights. These explanations help us understand why a particular prediction was made on a local level.

While ICE plots and LIME focus on local explanations for individual predictions, SHAP provides both model-agnostic and global explanations. SHAP values capture the contribution of each feature to a prediction across the entire dataset, allowing for a more comprehensive understanding of feature importance. Additionally, SHAP is applicable to a wide range of models and is able to handle feature interactions, thus providing a more nuanced understanding of how features interact to influence predictions. Based on these advantages, SHAP is selected as the best fitting interpretable machine learning method.

After the SCADA data has been analyzed using appropriate models, the results from the model have to be used to decide maintenance activities. These activities are triggered when some condition indicator crosses a preset limit. This guides the maintenance activities to take place based on the actual condition, as against faulty condition in corrective maintenance and perceived condition in preventive maintenance. Hence, condition-based maintenance strategy offers advantages that are associated with [Koukoura, 2021]:

- maintenance activities carried out only when required, e.g., reduced human errors in maintenance
- not conducting unnecessary scheduled replacement of parts before their end of useful life, e.g., cost saving

- advanced planning of maintenance activities, e.g., better planning

In spite of these advantages, use of a condition-based maintenance approach is still restricted and needs further research and development. This is because of the difficulties associated with the:

- quality and quantity of collected data
- handling of imperfect (spurious, inconsistent, inaccurate, uncertain, or irrational) data collected from faulty sensors
- interpretation of data to information regarding failure profile
- reasoning of information into knowledge about the existing status of the equipment
- converting knowledge to decision regarding maintenance scheduling
- handling of unreliable analysis that may trigger false alarm (false positive) or failure to respond (false negative)

Hence, a solution that integrates the traditional (corrective and preventive) maintenance methods with condition-based maintenance methods may provide a solution that is robust, effective, and efficient. In this integrated method:

- the failure analysis is carried out in the traditional manner, and then the results of failure profile is used judiciously to develop a maintenance strategy;
- the time for inspection and maintenance of a component is adjusted based upon condition monitoring.

This paper discusses the work carried out to develop methodology for identifying faults in a wind turbine generator bearing using interpretable machine learning models and using the results for rescheduling of its maintenance time. The methodology includes preprocessing of data to remove outlier data, use of machine learning models to predict bearing temperature, identification of deviation between predicted and actual temperatures, critical analysis of results, and recommendations for rescheduling of maintenance tasks.

2 DESCRIPTION OF PROCESS

In order to develop an effective and efficient asset management program for a component, it is important to understand the process in terms of the structure, environment, and operation.

A wind-turbine contains 20 to 25 bearings, all of which must be considered in a system-level reliability calculation of life expectancy [wind power engineering]. A typical roller bearing consists of four components: (a) inner ring, (b) outer ring, (c) cage, and (d) rollers. During an operation, these components are subjected to different levels of dynamic and static loads, which can be in axial, radial or combination direction under constant or alternating conditions. These loads cause degradation of the material because of wear (contact wear – peeling, scoring, smearing, etc.), fatigue (contact fatigue – flaking, spalling, etc.), corrosion, electrical erosion, plastic deformation, and fracture & cracking [ISO 15243], thereby resulting in the deterioration of the components and ultimately failure [Sankar, 2012]. As the degradation progresses, it also results in changes in the behavior patterns of parameters like temperature, vibration, noise, rotational speed, etc. By monitoring these parameters using appropriate sensors, it may be possible to diagnose the health of the bearings. Commonly used parameters for identifying fault in a bearing include temperature, vibration, and noise.

3 PROPOSED FAULT DIAGNOSIS METHODOLOGY

3.a Feature Selection

As discussed in the previous section, temperature is a commonly measured parameter to monitor the health of a bearing, because it is easy to continuously monitor and analyze in order to identify any abnormal behavior.

Figure 1 shows the simplified flowchart of heat transfers taking place in a bearing. A bearing is at a thermal equilibrium when it reaches a steady temperature. At this temperature, there is a balance between:

1. **Heat generation due to bearing friction (rolling, sliding, etc.) and seal friction** – During an operation, the friction among the components of a bearing results in generation of heat, the amount of which is dependent upon a number of factors, including the rotational speed, type of bearing, bearing geometry, elastic deformation under load of the rolling elements and raceways, type of lubricant and its application, and sliding friction between the components. The friction also results in its wear as a result of which there is an increase in bearing surface imperfections (deformation, pitting, craters, depressions, surface irregularities, spalling, cracking, etc.). The formation of surface imperfections leads to an increase in friction resulting in an increase in heat generation. Thus, an increase in friction due to structural imperfections or deterioration in lubrication increases the temperature of bearings.
2. **Conductive heat transfer from or to the adjacent parts** – Temperature of a bearing depends upon the heat input from or heat output to the adjacent parts. One piece of equipment that can significantly affect the bearing temperature is the generator itself. When the generator shaft rotates, heat is generated due to electrical resistance in the windings, resulting in heating of the generator. Since the temperature of the generator is higher than the temperature of the bearing, there is thus a heat transfer from generator to bearing. By measuring the temperature of the generator in stator windings, it may be possible to estimate the effect of the generator temperature on the temperature of the bearing.
3. **Convective heat dissipation to environment** – Temperature of a bearing in operation is generally above the environmental temperature, hence the bearing continuously dissipates heat to the environment. The rate of convective heat transfer is a function of:
    - **Convective heat transfer coefficient** – The convective heat transfer coefficient depends upon a number of parameters, including the air velocity over the solid surface and the specific heat capacity of humid air. The specific heat capacity of humid air is approximately proportional to the absolute humidity of air. Thus, as the humidity increases the value of convective heat transfer coefficient increases, resulting in an increase in heat loss [Boukhriss, Khalifa et al. 2013]. Thus, the temperature of a bearing depends upon the speed of air circulation around it and the relative humidity of air.
    - **Temperature difference between the bearing and the environment** – The rate of heat loss is proportional to the difference in the temperatures of the solid (bearing) and the environment. Thus, the temperature of the bearing depends upon the ambient temperature.

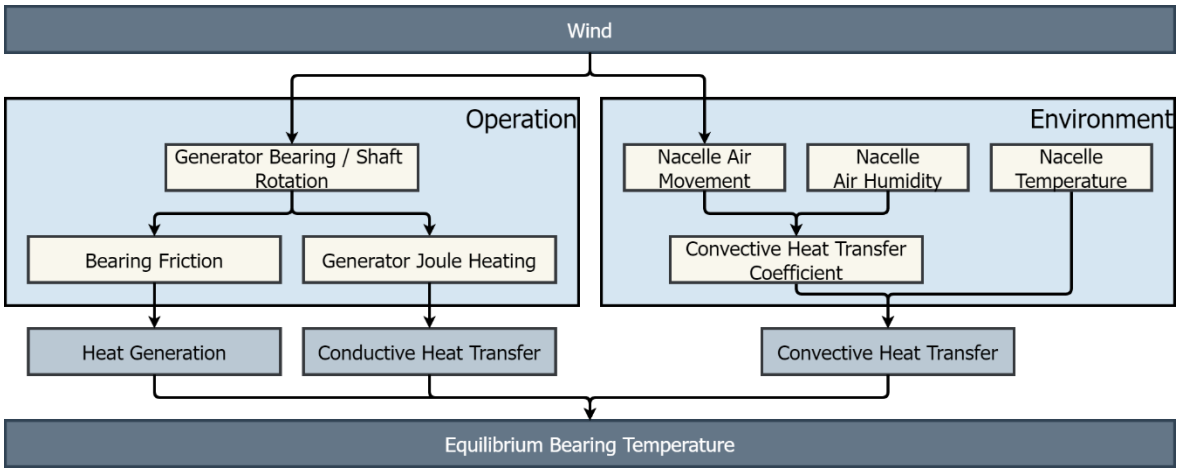

FIGURE 1. Flowchart showing the heat transfers taking place in bearings.

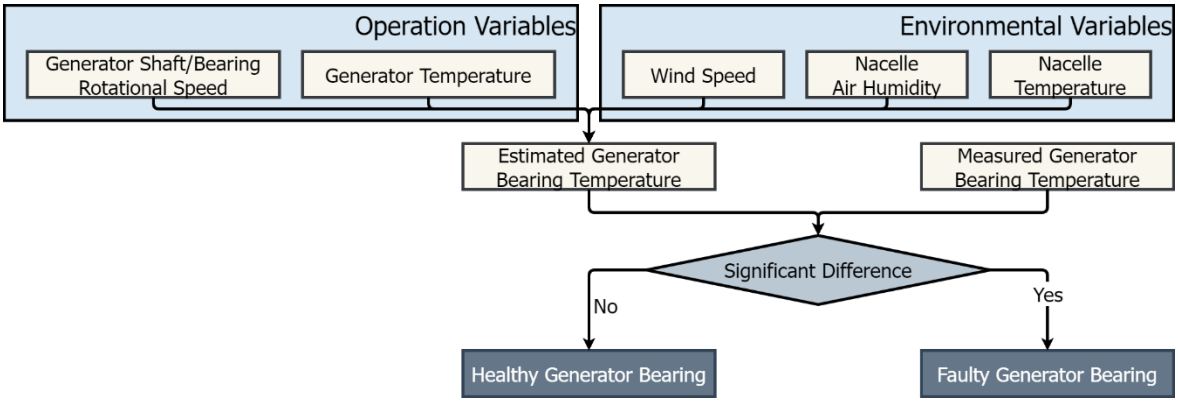

FIGURE 2. Flowchart showing the proposed fault detection methodology.

Based on the understanding of the heat transfers, five variables have been selected to predict the bearing temperature. These are:

1. **Generator Shaft / Bearing Rotational Speed** – This is the rotational speed of the high-speed shaft connected to the generator. The shaft is supported by the generator bearings, and thus rotation of the shaft leads to rotation of the bearing resulting in generation of heat in the bearings due to friction.
2. **Generator Temperature** – This measures the temperature of the generator stator windings. When the generator shaft rotates, heat is generated by electrical resistance in the windings. The windings are located close to the generator bearings and heat is transferred from the windings to the bearings.
3. **Wind Speed** – In a wind turbine, wind turns its rotor which in-turn rotates the shaft of the generator. Thus, wind speed determines the rotational speed of the generator shaft and bearing. Additionally, since the nacelle is not airtight, the wind speed impacts air movement inside the nacelle, which in turn influences the convective heat transfer rate.
4. **Nacelle Air Humidity** – This is the relative humidity of air inside the nacelle.

5. **Nacelle Temperature** – This is the temperature measured in the confined space housing the wind turbine drivetrain. The generator is located at the back of the nacelle and is therefore affected by the ambient temperature in the nacelle.

Figure 2 shows the flowchart of the methodology employed for detecting fault in a bearing. Using the five parameters, it may be possible to estimate temperature of a healthy bearing and if the measured temperature is above the predicted value, then there is a possibility that the higher temperature is the result of increased friction due to degradations in the bearing or lubrication.

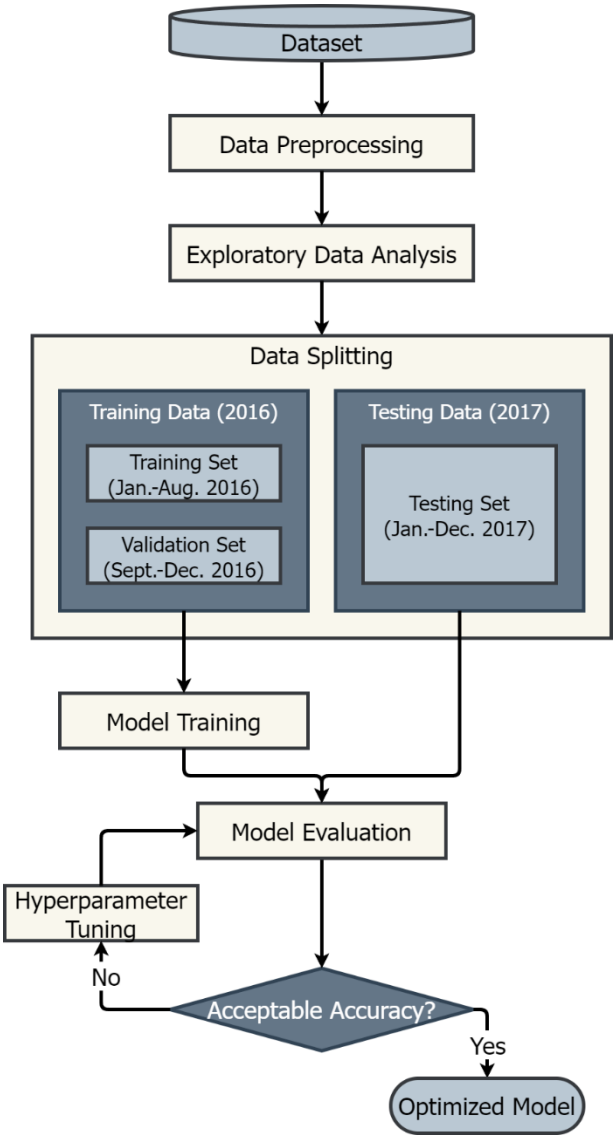

FIGURE 3. Flowchart for developing the proposed interpretable machine learning model.

3.b Proposed Model for Predicting Bearing Temperature

As discussed in the previous section, the first step is to predict the bearing temperature using the five input variables. **Figure 3** shows the flowchart of proposed methodology for predicting bearing temperature using machine learning algorithms.

Selection of Regression Algorithms

In this project a number of machine learning algorithms have been considered for developing a predictive model. These included:

- Linear Models – Linear Regression (LR), Lasso, Ridge, and Bayesian Ridge Regression
- Tree-based Models – Decision Trees, Random Forest (RF)
- Boosting Models – AdaBoost, XGBoost and LGBoost
- Support Vector Regression (SVR)

Out of these, four algorithms – Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost – have been shortlisted for further testing. These algorithms are commonly used and are known as supervised learning algorithms, a subcategory of machine learning algorithms [Mahesh 2020].

Data Preprocessing

Data preprocessing is an important step of any machine learning model. This is because raw data is typically created, processed, and stored by a mix of humans and business processes, often resulting in imperfections like vague, inconsistent, irrational, duplicate or missing values. These imperfections need to be corrected for the algorithms to work properly. Hence, an important step in preprocessing is to identify and handle (often remove) outliers. The outliers are removed only from the training and evaluation data so that the models can be trained and evaluated on healthy turbine operation data. This improves the models' capability to detect anomalies in the test data.

Exploratory Data Analysis (EDA)

Exploratory data analysis is used to analyze and investigate the data set and summarize the main characteristics by employing data visualization methods. Common methods include the use of Pearson, Kendall, or Spearman correlation matrices. These matrices depict the correlation between all the possible pairs of values and is a powerful tool to identify and visualize patterns in data.

Data Splitting – Training, Validation and Testing Data

In supervised machine learning tasks, best practice is to split data into three independent data sets:

1. **Training set** – This is the data set that is fed to the model in order for it to learn relationships and recognize patterns in the data.
2. **Validation set** – The validation data set is used to test model performance and configuration of hyperparameters. This data set needs to be independent from the training data set so that the model does not overfit and fail to generalize.
3. **Testing set** – After the validation data set is used to determine algorithm and parameter choices, the test data set is used to understand the model's performance on unseen data.

Model Training

Model training is the process of teaching a machine learning model to make predictions or perform a specific task by exposing it to a labeled data set. The goal of model training is to enable the model to learn patterns, relationships, and rules from the training data so that it can generalize its knowledge to make accurate predictions on unseen or future data.

Model Evaluation

In order to select the best performing algorithm out of the four, some criteria for evaluation need to be applied. These criteria should be able to judge a model's performance regarding (a) accuracy of prediction, (b) compatibility with interpretable machine learning tools, (c) time usage for carrying out the calculations, and (d) simplicity. The selection of the best model is based on an overall assessment of all the criteria.

To evaluate the accuracy of prediction, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2) have been used.

Hyperparameter Tuning

Many machine learning algorithms require hyperparameters that need to be defined before running them. First-level model parameters are decided during training, but the second-level tuning parameters need to be tuned to optimize the performance. Typically, this is done by performing cross-validation or evaluating predictions on a separate test set [Probst, Boulesteix et al. 2019].

In this analysis, hyperparameter tuning is performed using grid search [Bergstra and Bengio 2012] and hyperparameter values suggested by Probst, Boulesteix et al. [2019]. This method runs through all possible combinations of the parameters within their search ranges forming a grid. It is performed using the scikit-learn library for python programming language. The grid search finally ranks all the combinations by their mean RMSE score across the same cross-validation folds used for model evaluation. Results from the grid search are used to select the optimal values for the hyperparameters.

Besides grid search there are additional hyperparameter tuning methods such as random search and Bayesian optimization. Grid search is selected due to its transparency and reproducibility, as well as its robustness against local optima. By evaluating all possible combinations, it reduces the risk of getting stuck in suboptimal regions of the hyperparameter space, and hence it increases the likelihood of finding the best set of hyperparameters for a given problem.

3.c Model Interpretation Using SHAP

Once the model has been tuned using optimal hyperparameters, it is ready to be interpreted. SHAP has been used to interpret outputs of the best performing machine learning model and quantifying impact of each features to predictions. A negative SHAP value indicates a negative impact that decreases the value of the model output, whereas a positive SHAP value indicates a positive impact that increases the value of the model output. Although a SHAP analysis does not explicitly imply causalities, it helps in interpreting how each feature contributes to the model output and helps to identify importance of a feature in a model prediction.

4 CASE STUDY – MODEL FOR PREDICTING BEARING TEMPERATURE

4.a SCADA Data

To demonstrate the feasibility of the proposed methodology, SCADA data made available by the energy company EDP [2017] from four horizontal axis wind turbines located off the western coast of Africa has been used. The data has been recorded over a period of 2 years (2016 and 2017) at a 10-minute averaging interval. The datasets contain values of 76 parameters. Besides this, associated datasets about meteorological conditions have also been provided for the same time instances. Failure logs containing timestamp, damaged component and associated remarks are also available. For this work, Turbine Number 7 ("T07") has been selected because its failure log has recorded generator bearing failure. For Turbine Number 7, the total number of instances are 52445 and 52294 for 2016 and 2017, respectively.

The generator uses two bearings, one on the drive-end and one on the driven end. The failure log records damage of generator bearings on August 20, 2017, at 08:08:00, and damage of generator shortly afterwards on August 21, 2017, at 16:47:00. The downtime caused by the generator failures is highlighted in green in **Figure 4** and lasts from August 20, 2017, at 08:10:00 until August 28, 2017, at 21:50:00. The model shall attempt to predict these failures.

TABLE 1. Selected features and target for developing the model.

Variable	Description	Units
Timestamp	10-minute resolution	
Features		
Gen_RPM	Generator shaft / bearing rotational speed	rpm
Gen_Phase_Temp	SCADA dataset gives the average temperature inside generator in stator windings Phase 1, 2 and 3. Since the temperatures are nearly the same, Gen_Phase_Temp is an average temperature of the three temperatures	ºC
Wind_Speed	Ambient wind speed	m/s
Humidity	Relative nacelle air humidity	%
Nac_Temp	Nacelle temperature	ºC
Target		
Gen_Bear_Temp	Temperature in generator bearing 1 (Driven End)	ºC

TABLE 2. Failure log for Turbine Number 7 ("T07").

Timestamp	Component	Remarks
August 20, 2017, 08:08:00	Generator bearing	Generator bearings damaged
August 21, 2017, 16:47:00	Generator	Generator damaged

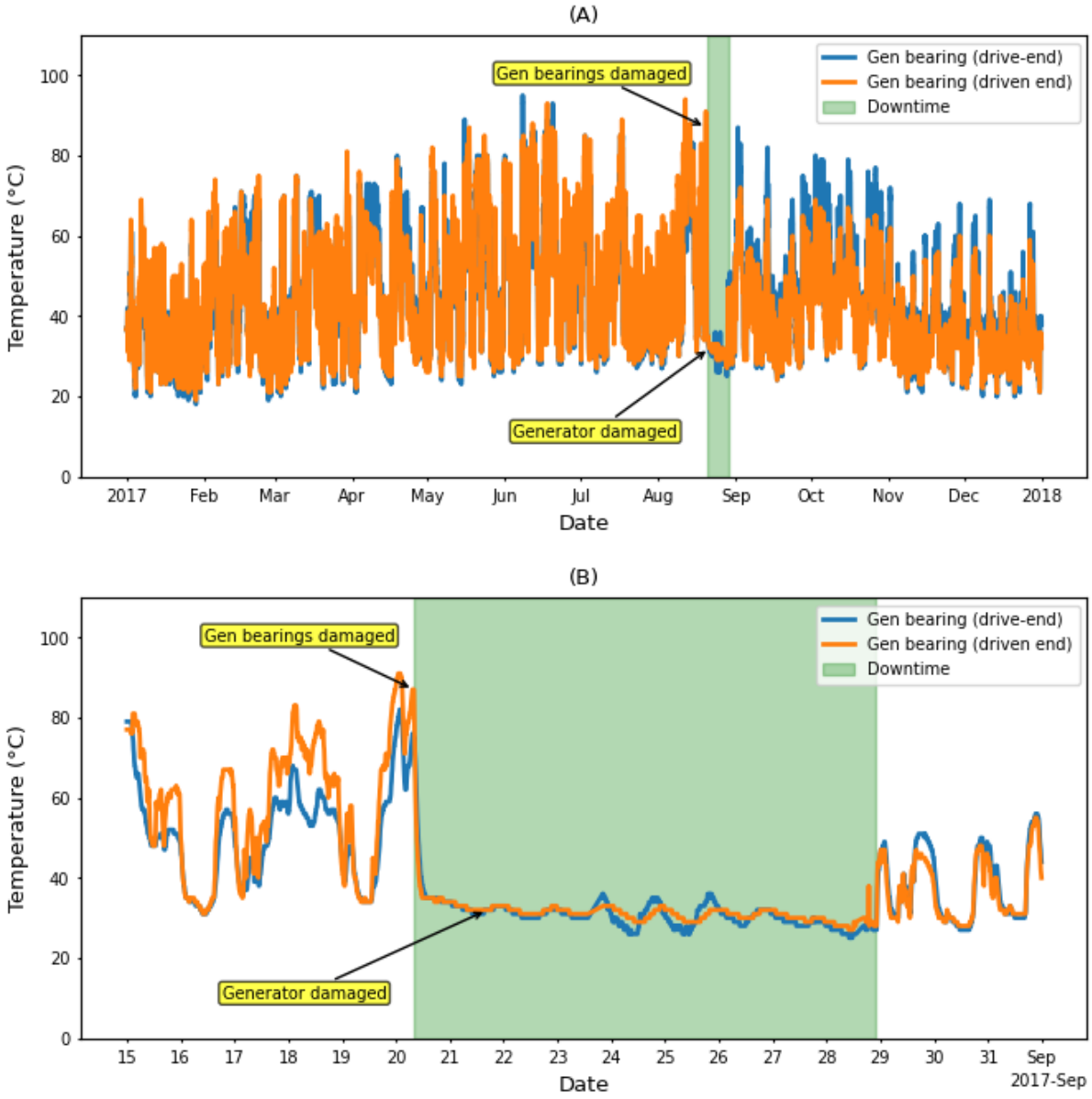

FIGURE 4. Bearings temperature during the bearing and generator failures in (A) 2017 and (B) August 2017.

4.b Data Preprocessing

Identification of Data Outliers

Quite often SCADA data contains outliers that arise due to imperfections in the SCADA system and do not reflect the actual condition of process, environment, or component. For the development of a predictive model, it is important to remove these outliers because their presence can lead to biases in the model.

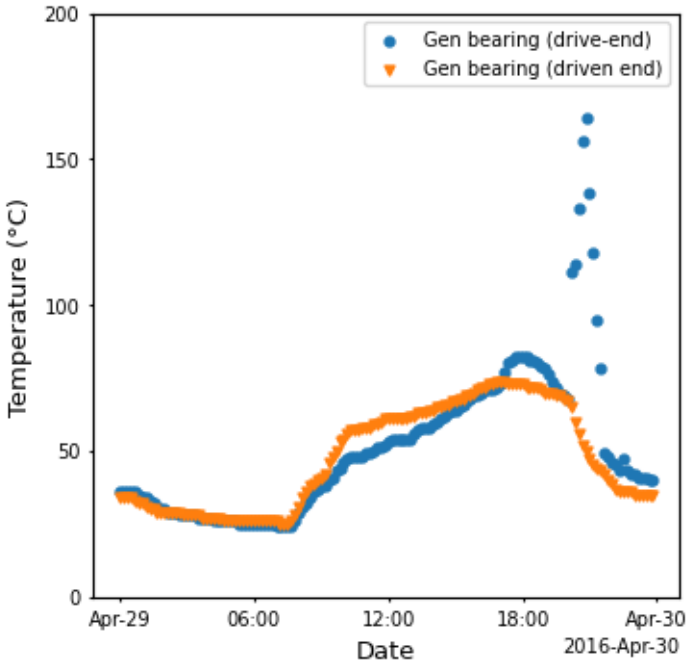

FIGURE 5. Effect of faulty sensors on recorded temperature of bearings.

One common reason for outliers in the data is the inputs from faulty sensors. Since health prognosis of a bearing relies heavily on the data collected by the sensors, the reliability of analysis thus depends upon the reliability of the collected data. Hence, the reliability of results from the proposed methodology also depends upon the quality of data used for the analysis.

Figure 5 shows plots of the temperature data versus selected periods of the two bearings. Sudden spike in the recoded temperatures can only be due to errors in the data collection, possibly arising due to the faulty sensor. This is justified by the record showing that the sensor was replaced on 2016-04-30 12:40 after recording *High temperature in generator bearing 1*. Outliers like those shown in the figure need to be handled during the data preprocessing.

In this model outliers have been identified by the use of box plots, shown in **Figure 6**. In a box plot, the lower limit of the whisker marks the minimum value, excluding outliers, whereas the upper limit of the whisker marks the maximum value, excluding outliers. The lower limit of the box is the first quartile (Q1 or the 25th percentile), whereas the upper point of the box is the third quartile (Q3 or the 75th percentile). All values within the box between Q1 and Q3, also called the interquartile range (IQR), are calculated using Equation (1). The horizontal red line in the box is the median value. An outlier in this case is defined as a value outside 1,5 times the IQR above Q3 or below Q1.

$$IQR = Q3 - Q1 \tag{1}$$

Where:
IQR = Interquartile range
$Q1$ = the first quartile, or the 25th percentile
$Q3$ = the third quartile, or the 75th percentile

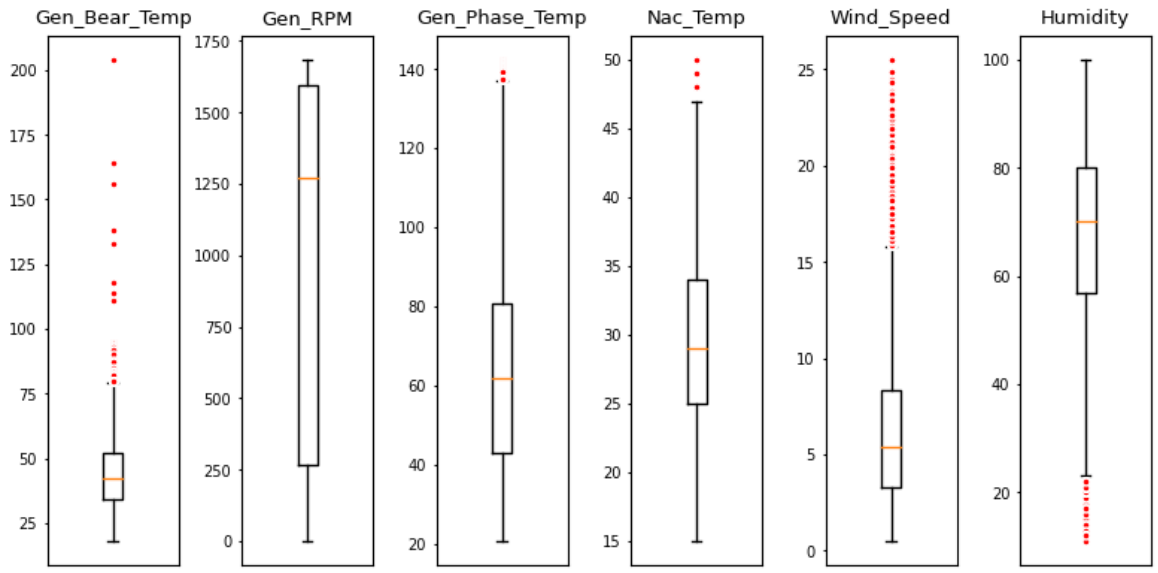

FIGURE 6. Box plot of SCADA signals.

Data Cleaning

Depending upon the characteristics of specific variables, rules for identification and handling of outliers have also been adopted. For example, a threshold of 100℃ has been set for the generator bearing temperature and all values higher than this have been removed. Similarly, relative humidity values are missing in the period January 3, 2017, to May 6, 2017, and this gap has been filled with values from the previous year.

Further cleaning has been performed using DBSCAN [Ester, Kriegel et al. 1996]. DBSCAN is a density-based clustering algorithm that works on the assumption that clusters are dense regions in space separated by regions of lower density. 'Densely clustered' data points are gathered into a single cluster.

The results before and after cleaning are shown in **Figure 7**. **Figure 7A** shows the presence of a significant number of outliers which indicate that either the turbine is not operating despite the blowing wind, or the sensors are not working properly. Additionally, there are many instances of the turbine not operating at its maximum potential. **Figure 7B** shows the plot after the removal of the most significant outliers and the remaining data points sufficiently fit the theoretical power curve.

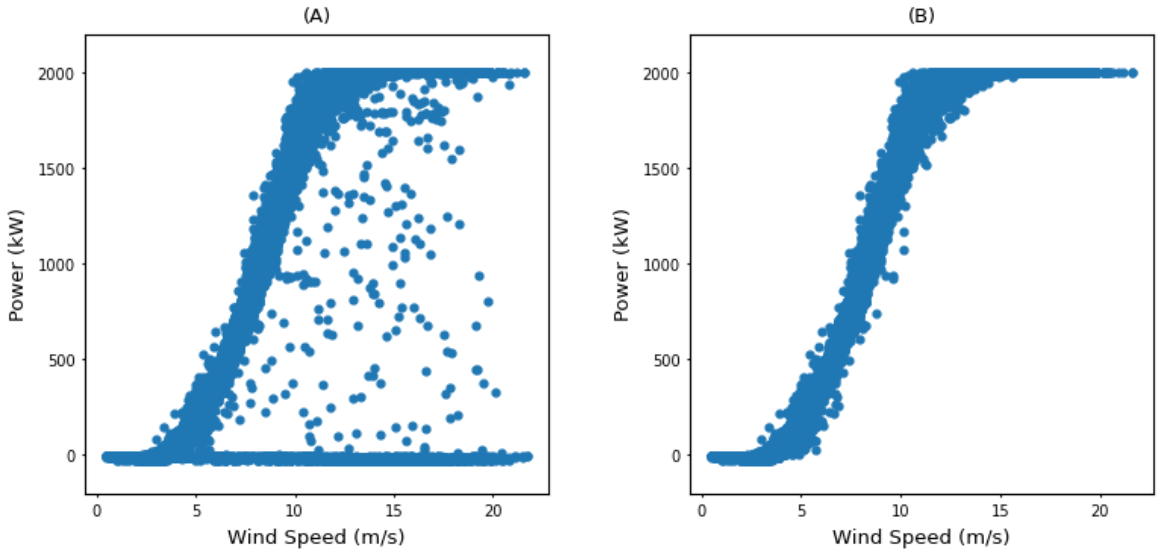

FIGURE 7. Plot of power generated versus wind speed using data of training period (A) Using raw. (B) Using data after cleaning outliers.

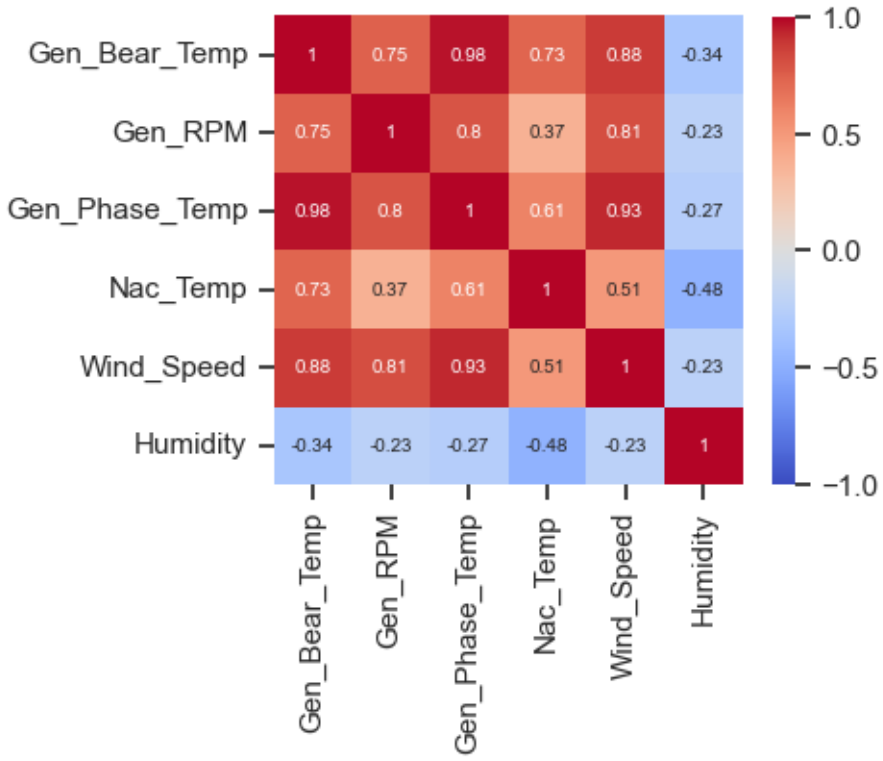

FIGURE 8. Pearson correlation matrix of the input features.

4.c Exploratory Data Analysis (EDA)

Figure 8 shows the Pearson correlation matrix of the input features and target. Some signals are highly correlated, for example (a) wind speed and generator rotational speed, (b) wind speed and generator phase temperature, and (c) generator phase temperature and bearing temperature. The matrix shows that the selected features are significantly relevant to the target variable.

To further understand the correlation between the features and target, pairwise relationships between them in the training set have been plotted (**Figure 9)**. The marginal histograms have been prepared by dividing signal values into 25 bins.

Effect of Generator Shaft / Bearing Rotational Speed on Bearing Temperature

The time averaged wear rate of a bearing can be given as [Gupta 2011]:

$$W(T) = \frac{1}{T}\frac{K}{H}\int_{o}^{T} Q(t)u(t)\,dt \tag{2}$$

Where:
W = Time-averaged wear rate over the time interval (T)
K = Wear coefficient
H = Hardness of the material being subjected to wear
Q = The time-dependent load at a given interaction
u = Sliding velocity as a function of time

The equation shows the dependence of wear on the parameters Q and u, which in turn are dependent upon the rotational speed. Thus, the wear rate increases with an increase in the rotational speed. Corresponding to the increase in wear, the heat generated due to friction also increases with the increase in the rotational speed. This increase in heat generation manifests itself as an increase in the temperature.

Figure 9 shows the bearing temperature (*Gen_Bear_Temp*) is a function of the rotational speed of generator shaft / bearing (*Gen_RPM*).

Effect of Generator Temperature on Bearing Temperature

In a generator, heat is produced in the windings of the stators due to the passage of electricity through the electric wiring (Joule Heating). This heat is dissipated to the surrounding through conduction and convection. A part of dissipated heat also increases the temperature of the generator bearings.

Figure 9 shows the approximately linear relationship between the generator temperature (*Gen_Phase_Temp*) and the bearing temperature (*Gen_Bear_Temp*).

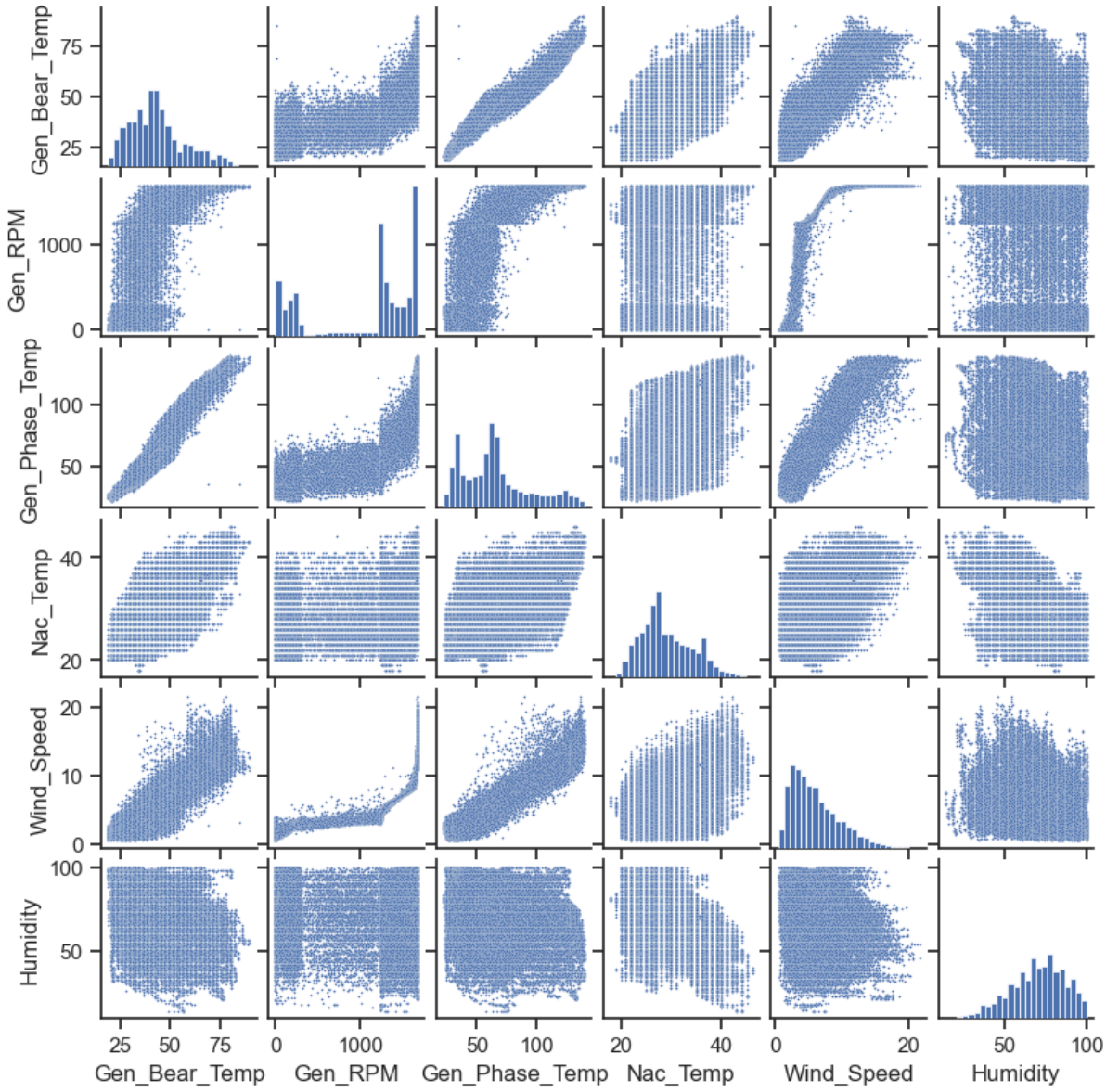

FIGURE 9. Pairwise relationships between input features.

Effect of Wind Speed on Bearing Temperature

Wind speed has two opposing effects on the bearing temperature. On the one hand, an increase in wind speed increases the rotational speed of bearing resulting in increase in temperature due to friction. On the other hand, wind speed also increases air circulation within the nacelle, thereby increasing the convective heat transfer coefficient and subsequently heat loss from the bearing.

Figure 9 shows that there is a net increase in bearing temperature (*Gen_Bear_Temp*) with an increase in wind speed (*Wind_Speed*).

Effect of Nacelle Air Humidity on Bearing Temperature

Since the specific heat capacity of humid air increases with an increase in the relative humidity of air, expectedly an increase in relative humidity increases the convective heat transfer coefficient and subsequently increases heat loss from the bearing.

Figure 9 shows a weak correlation between the relative humidity of air (*Humidity*) and the bearing temperature (*Gen_Bear_Temp*).

Effect of Nacelle Temperature on Bearing Temperature

The ambient temperature in the nacelle follows an annual cycle, whereby the temperature is lower during winters and higher during summers. Since the convective heat transfer is proportional to the temperature difference between a bearing's surface temperature and the ambient temperature, this variation in the ambient temperature has an effect on the heat dissipation from bearing to the environment.

Figure 9 shows an increase in the bearing temperature (*Gen_Bear_Temp*) with an increase in ambient temperature inside nacelle (*Nac_Temp*).

4.d Data Splitting – Training, Validation and Test Data

The data from 2016, after the removal of outliers, has been used for training the model in two steps. In the first step, the clean 2016 data is split into two parts – training data and validation data. The data from the first eight months is used to train the algorithms, while the data from the last four months is used to evaluate (validate) the algorithms. Four month-long validation data can be considered sufficient to cover different parts of the time series such as trends and seasonality patterns. The validation data has been divided into four folds, each lasting for nearly a month. The initial part of the validation set is correlated with the last part of the training set. In order to increase independence between training and validation, a gap of 24 hours is removed from the end of the training set close to the validation set.

In the second step, the best performing model has been trained on all data in 2016 in order to capture any seasonal variations.

Thus, the complete dataset has been split into training data (33%), validation data (17%) and test data (50%). The dataset contains over 100 000 timestamps, and hence using only 33% (in the first step) and 50% (in the second step) of the data for training is sufficient. Holding out 17% of the data for validation is in the recommended range [Belyadi and Haghighat 2021].

4.e Model Training

The four shortlisted algorithms – Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost – are trained using the training data set. For the algorithms to be evaluated on equal terms, all algorithm parameters are set to their default values during initial training.

4.f Model Evaluation

In the first step, performance of the four algorithms – Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost – have been evaluated. **Table 3** presents the RMSE scores for the four algorithm from the cross validation. The table shows that Support Vector Regression (SVR) has the best RMSE mean score whereas Linear Regression (LR) has

the worst. The existence of almost equal RMSE values across different folds signifies that the data is evenly distributed over the time period.

Table 4 presents the results of the evaluation of the four models on the whole one-year test set (2017). There is a noticeable difference in the RMSE scores when the models predict a whole year compared to only the folds in the cross validation. This is due to the test set containing faulty turbine operational data whereas the cross validation set consists of only healthy turbine operational data similar to the training set used to learn the model. The evaluation results suggest that:

- **Linear Regression (LR)** – This has a decent score and shortest fit and prediction time.
- **Random Forest (RF)** – This has a good score but somewhat long fit time.
- **Support Vector Regression (SVR)** – This goes from top performing algorithm on the validation data to worst performing on the test data in almost all parameters, highest RMSE and longest fit and predict time.
- **XGBoost** – This scores on top while having an acceptable fit and predict time.

TABLE 3. Cross validation RMSE scores

Model	Fold 0	Fold 1	Fold 2	Fold 3	Mean
LR	1,61	1,74	1,62	1,57	1,64
RF	1,53	1,68	1,57	1,58	1,59
SVR	1,48	1,55	1,46	1,31	1,45
XGBoost	1,48	1,74	1,48	1,51	1,55

TABLE 4. Performance of models with default parameters

Model	MAE	MAPE	MSE	RMSE	R^2	Fit time [s]	Predict time [s]
LR	1,569	0,039	4,436	2,106	0,980	0,011	0,005
RF	1,479	0,035	3,888	1,972	0,982	18,104	0,889
SVR	1,521	0,037	4,887	2,211	0,978	90,701	188,590
XGBoost	1,436	0,034	3,824	1,955	0,983	1,266	0,019

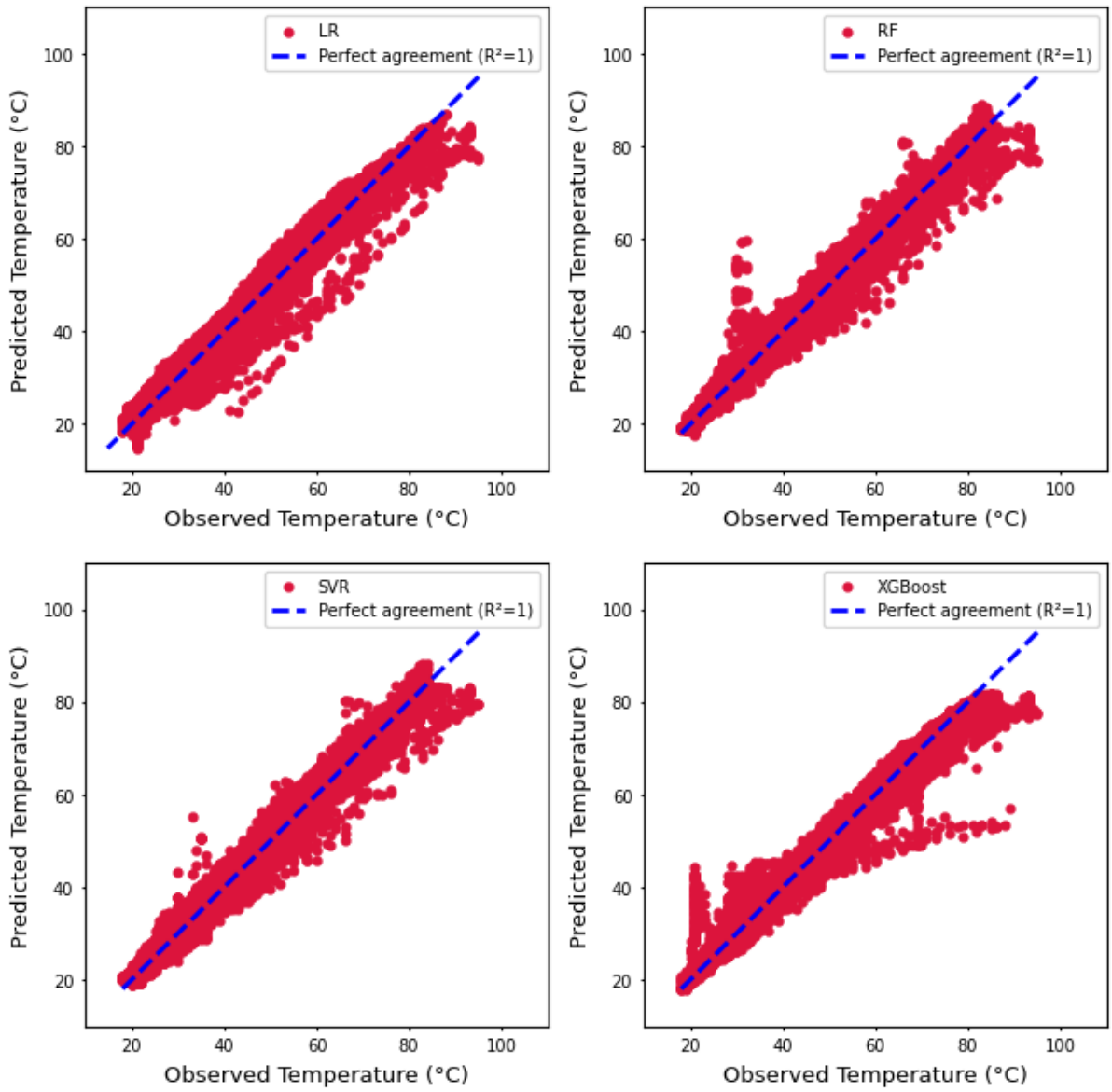

FIGURE 10. Predicted and observed temperatures for all models

To visualize the performance of the algorithms, plots of the predicted temperatures versus observed temperatures are shown in **Figure 10**.

- **Linear Regression (LR)** – This tends to predict rather low values
- **Random Forest (RF)** – Along with XGBoost this appears to give the best fit
- **Support Vector Regression (SVR)** – This predicts high values for some low bearing temperatures and low values for some high bearing temperatures.
- **XGBoost** – This appears to be the most accurate model, even though at times it predicts high values for some low bearing temperatures

Based on the detailed evaluation, XGBoost can be considered the most suitable algorithm for the model and has been fine-tuned using hyperparameter tuning techniques.

4.g Hyperparameter Tuning

As described in the previous section, the XGBoost model has been selected as the most suitable model for further analysis. An important part of machine learning optimization is the tweaking and tuning of hyperparameters. Hyperparameter tuning is performed in the XGBoost model to enhance the model's accuracy before trying it on the test data set. The selected hyperparameters and their suggested ranges [Probst, Boulesteix et al. 2019] for tuning are presented in **Table 5**. In addition to the parameters in **Table 5**, the parameters *colsample_bytree* and *colsample_bylevel* have been set to 0,6. In order to determine the optimal combination of hyperparameters grid search with cross validation strategy has been performed.

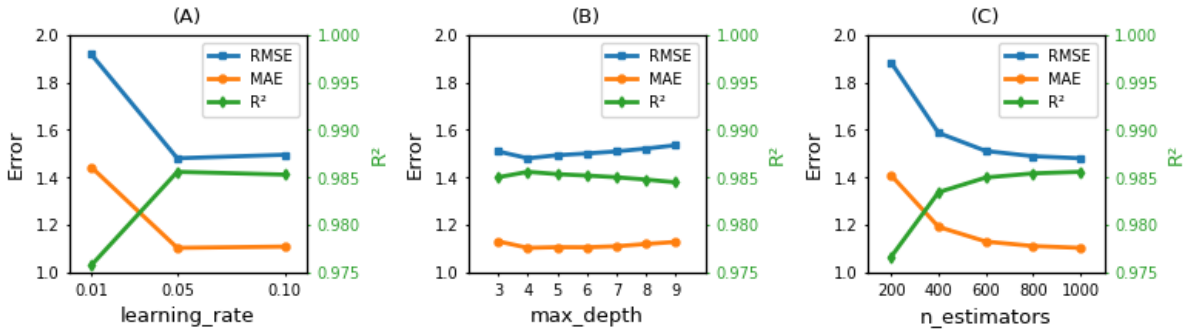

FIGURE 11. Model impact changing (A) *learning_rate*, (B) *max_depth* and (C) *n_estimators*

TABLE 5. Hyperparameter search range

Hyperparameter	Search range	Optimal value
n_estimators	[200, 400, 600, 800, 1000]	1000
max_depth	[3, 4, 5, 6, 7, 8, 9]	4
learning_rate	[0,1 , 0,05 , 0,01]	0,05

TABLE 6. Optimized XGBoost performance on test data and validation data

Test Data Performance					
Model	**MAE**	**MAPE**	**MSE**	**RMSE**	**R^2**
XGBoost	1,436	0,034	3,824	1,955	0,983
Optimized XGBoost	1,389	0,033	3,354	1,832	0,985
Change [%]	3,272	2,941	12,291	6,292	0,203
Validation Data Performance [RMSE]					
Model	**Fold 0**	**Fold 1**	**Fold 2**	**Fold 3**	**Mean**
XGBoost	1,48	1,74	1,48	1,51	1,55
Optimized XGBoost	1,41	1,65	1,44	1,40	1,48
Change [%]	4,73	5,17	2,70	7,29	4,52

Results from the grid search are displayed in **Figure 11**. The figure shows that as compared to *max_depth*, *learning_rate* and *n_estimators* have more effect on performance of the algorithm in terms of RMSE, MAE and R^2. The optimal values of these parameters are given in **Table 5**.

Table 6 shows the performance of XGBoost algorithm after hyperparameter tuning using the optimized parameter values given in **Table 5**. As shown, there is an improvement in the performance of the algorithm after hyperparameter tuning.

4.h Prediction of Generator Bearing Temperature

The optimized XGBoost algorithm-based model (**Figure 3**) has been used to predict bearing temperature using the Testing Data (2017).

Figure 12 shows the plots of the actual and predicted values for the period January 1 to January 15, 2017, the curves of which are for:

- actual temperature
- predicted temperature
- predicted plus/minus 2 standard deviation temperature

The figure shows that the actual temperature remains within the (predicted ± 2 standard deviation) temperature range.

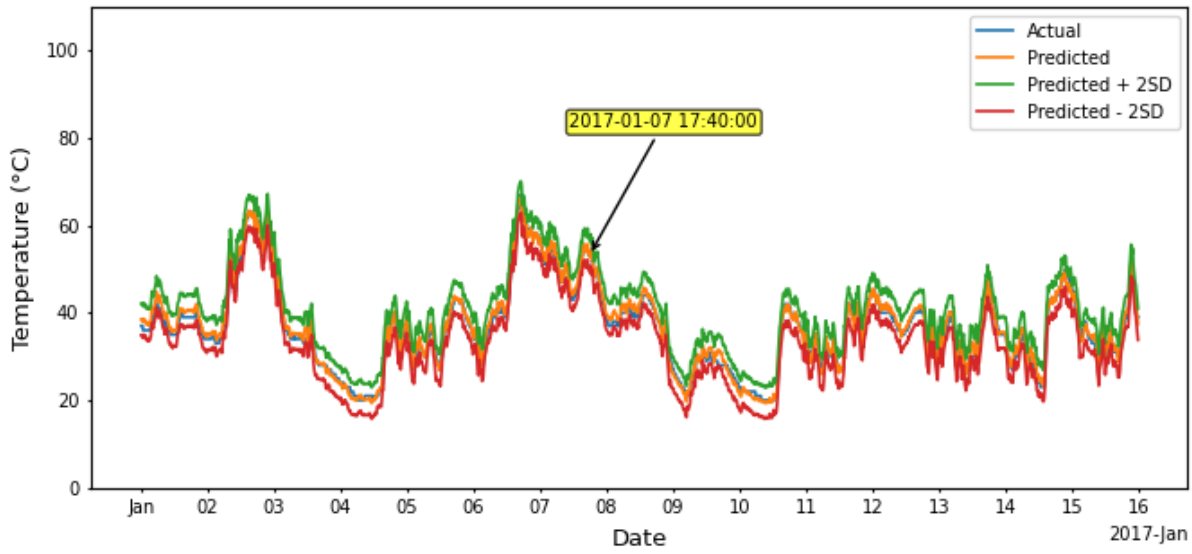

FIGURE 12. Actual and predicted temperatures of generator bearing for the period January 1 to January 15, 2017.

4.i Sources of Error

Inaccuracies in the output results may arise due to:

- The high correlations between feature and target variables may impact how the machine learning model learns. This risk is partly mitigated by using hyperparameters *colsample_bytree* and *colsample_bylevel*.
- Faulty sensors
- Wrong calibration or drift in calibration of sensors

In the case study there may be additional sources of errors, including:

- Replacing the missing humidity data with the values from the previous year

4.j Fault Detection and Recommendation for Rescheduling Maintenance Plan

Figure 13 shows the plots of the actual and predicted values for the period from June 7 to June 23, 2017. During this period there are times when the actual bearing temperature exceeds the predicted value by more than two standard deviations over significantly long periods, and this is highlighted in green. For example, on June 7, 2017, the actual value reaches 95°C whereas the model prediction is 76°C, a difference of 19°C.

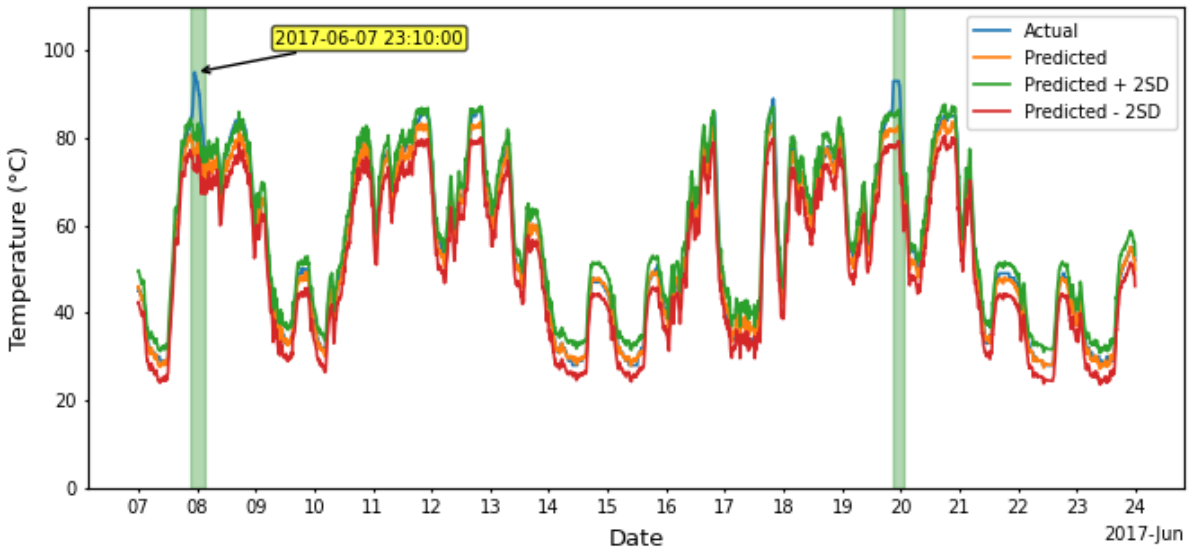

FIGURE 13. Actual and predicted temperatures of generator bearing for the period June 7 to June 23, 2017.

After June 7, 2017, there is a tendency for the actual bearing temperature to be higher than the predicted bearing temperature. At times it often crosses the two standard deviation limit. This indicates two possibilities:

- Malfunctioning of the bearing sensor.
- Possibility that the bearing is getting hotter than expected perhaps due to increased friction. The increased friction could be either because of increased wear or improper lubrication. Both of these possibilities warrant special inspection and monitoring activity.

Based on the detection of faulty bearing, recommendation may be made for scheduling maintenance activities at the earliest opportunity. This recommendation is justified by the fact that the bearing breaks down two months later on August 20, 2017.

5 MODEL INTERPRETATION USING SHAP

The XGBoost algorithm-based model used for the case study gives reasonably good predictions for the temperature of a generator bearing. The model needs to be further evaluated to interpret it's working. Since XGBoost is a tree-based model, the Tree SHAP algorithm proposed by Lundberg, Erion et al. [2018] for tree ensembles can be used to calculate the SHAP values that could be used for the interpretation of the working.

5.a Global Explanations

Figure 14A shows the mean absolute SHAP values for the used features. The figure shows that:

- The generator phase temperature has by far the highest impact on the model predictions. This is reasonable due to the adjacent location of the bearing and generator.
- Nacelle temperature and wind speed have moderate average impact on the model predictions, which should be expected since the convective heat loss from bearing is directly proportional to the difference in temperature between the bearing and the nacelle temperature. Wind speed affects not only the rotational speed but also the convective heat loss.
- Generator or bearing rotational speed and relative humidity have low impact.

Figure 14B shows the changes in the SHAP value for changes in the feature value. For all features except the humidity, a higher feature value has a positive impact on the model prediction, and a low feature value has a negative impact on the model output. As is to be expected, the humidity has the opposite impact for its feature values, because increase in humidity increases the specific heat capacity of air resulting in higher convective heat loss from the bearing and a decrease in temperature.

SHAP treats each feature as a "player", hence there are interaction effects between features. The SHAP main effect plots in **Figure 15** remove all interaction effects between features and thus display the raw impact of each feature.

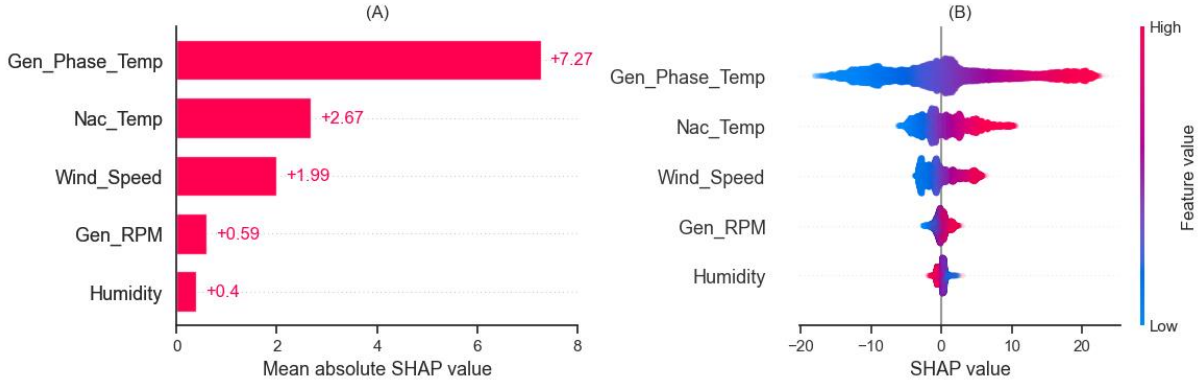

FIGURE 14. (A) Mean absolute SHAP value per feature. (B) Matrix plot of SHAP values for different features.

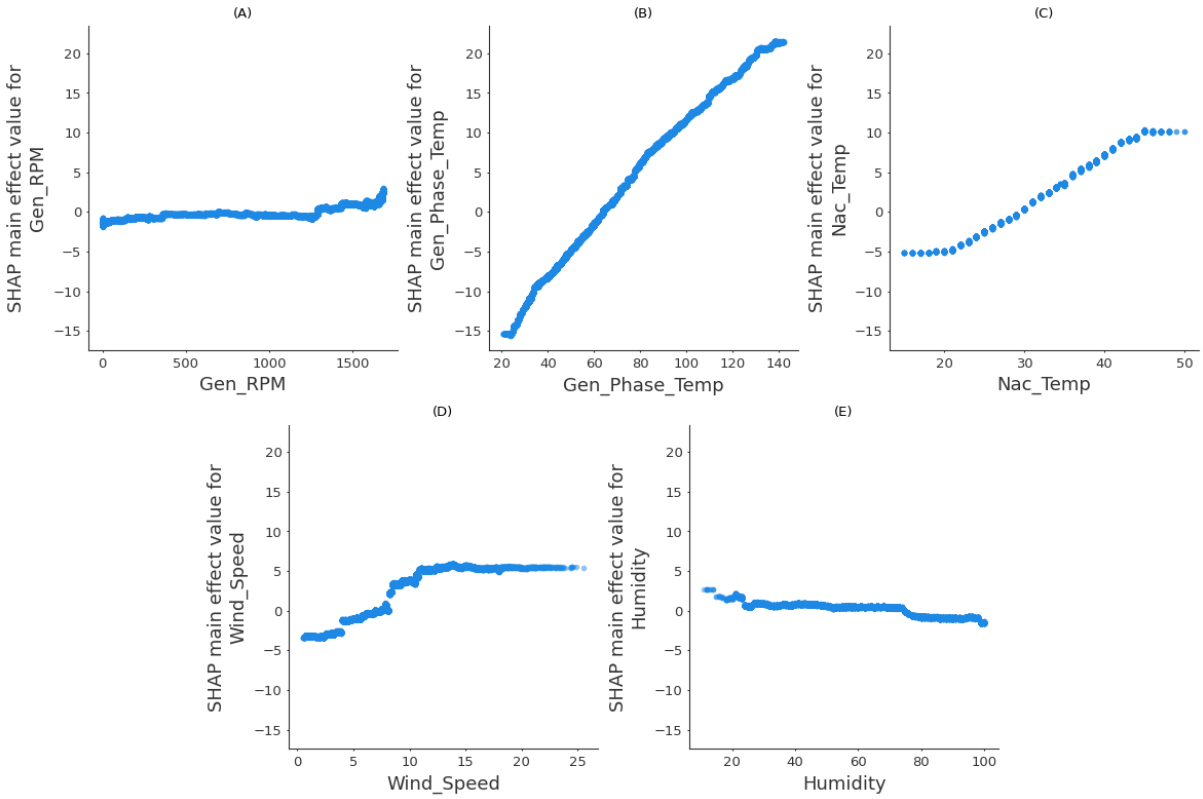

FIGURE 15. SHAP main effects plot for (A) generator rpm, (B) generator phase temperature, (C) nacelle temperature, (D) wind speed and (E) humidity.

- **Generator Shaft / Bearing Rotational Speed** – Generator rotational speed has a low impact with a small positive spike near its max rotation speed.
- **Generator Temperature** – The generator phase temperature has a dominant and nearly linear impact on the model output.
- **Wind Speed** – At the cut-in wind speed of 4m/s, there is a marked increase in the impact of wind speed. It increases up until the rated wind speed of 12m/s and from there on stays constant.
- **Nacelle Air Humidity** – The impact of humidity is rather weak and decreases slowly across its range.
- **Nacelle Temperature** – Nacelle temperature has an increased positive impact in the temperature range 20-45°C.

5.b Local Explanations

SHAP waterfall plots are used for explaining individual predictions. Starting from the expected value of the model output (the average prediction of the model on the training data) at the bottom of the waterfall plot, each row shows the contribution of each feature to the model output for a prediction. A positive (red) contribution moves the initial output value higher whereas a negative (blue) contribution moves the initial output value lower.

Explanation of Prediction for January 7, 2017

Figure 12 shows the plots of the actual and predicted values for the period of January 1 to January 15, 2017. During this period all predicted values are within two standard deviations of the actual value, indicating a possibility that the bearing is operating normally. From this period, an instance (January 7, 2017, 17:40:00) has been randomly selected for local explanation.

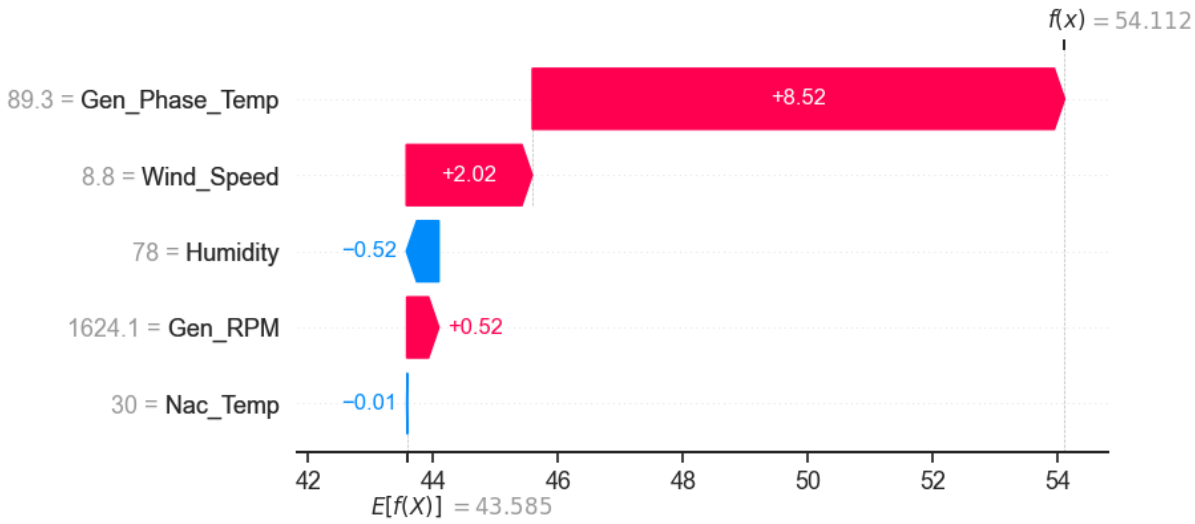

FIGURE 16. Local explanation on January 7, 2017, 17:40:00 by waterfall plot.

According to **Figure 15**, the temperature of bearing is influenced most by the generator temperature because of its high temperature and proximity to the bearing. This is followed by the nacelle temperature and wind speed. The generator rotational speed and humidity have relatively minor effect.

On January 7, 2017, at 17:40:00 the actual generator bearing temperature is 53℃. The SHAP waterfall plot in **Figure 16** explains how the XGBoost model arrived at a prediction of 54℃.

- **Generator Shaft / Bearing Rotational Speed** – Rotational speed has minor effect on the predicted temperature value, hence the net heating effect on the predicted bearing temperature (+0.52 ^{o}C) is relatively small.
- **Generator Temperature** – The high generator phase temperature (89.3^{o}C) has by far the most significant positive influence (+8.52^{o}C) on the bearing temperature.
- **Wind Speed** – Wind speed makes relatively small positive effect (+2.02^{o}C) on the predicted value. Wind speed has two opposing effects – increase in temperature due to increased friction and decrease in temperature due to increased convective heat loss. In this case the rotational speed has small effect (+0.52^{o}C) and hence a greater positive effect may be due to the interaction between the wind speed, the generator temperature and the bearing temperature.
- **Nacelle Air Humidity** – The high relative humidity (78%) also does not significantly (-0.52^{o}C) affect the predicted temperature value, because relative humidity itself does not have any significant role.
- **Nacelle Temperature** – The nacelle temperature (30^{o}C) is close to the average annual temperature, ranging between 15-50^{o}C, and hence does not play a significant role (-0.01^{o}C) in the fall of temperature on predicted value.

Explanation of Prediction for June 7, 2017

Figure 13 shows the plots of the actual and predicted values for the period June 7 to June 23, 2017. On June 7, 2017 (Summer), the environmental and operating temperatures are quite different from those of January 7, 2017 (Winter). Based on the SHAP waterfall plot (**Figure 17**), an attempt is made to explain the working of the model.

- **Generator Shaft / Bearing Rotational Speed** – As in the previous case (January 7, 2017), the rotational speed has a minor effect on the predicted temperature value, and hence the net heating effect on the predicted bearing temperature (+1.48^{o}C) is relatively small. The small increase could be due to the small positive spike that appears near its max rotation speed (**Figure 14A**).
- **Generator Temperature** – The generator temperature is very high (137.3^{o}C) and this significantly (+20.95^{o}C) raises the temperature of the bearing.
- **Wind Speed** – Compared to the previous case, wind speed gives relatively higher positive effect (+4.43^{o}C) on the predicted value. This may be because of higher interaction between the wind speed, the generator temperature, and the bearing temperature.
- **Nacelle Air Humidity** – As in the previous case, nacelle relative humidity has negligible (-0.12^{o}C) effect on the predicted temperature value.
- **Nacelle Temperature** – Compared to the previous case, the nacelle temperature (39^{o}C) is 9^{o}C higher than the previous case, and hence there is significantly (+5.86^{o}C) higher effect on the predicted temperature.

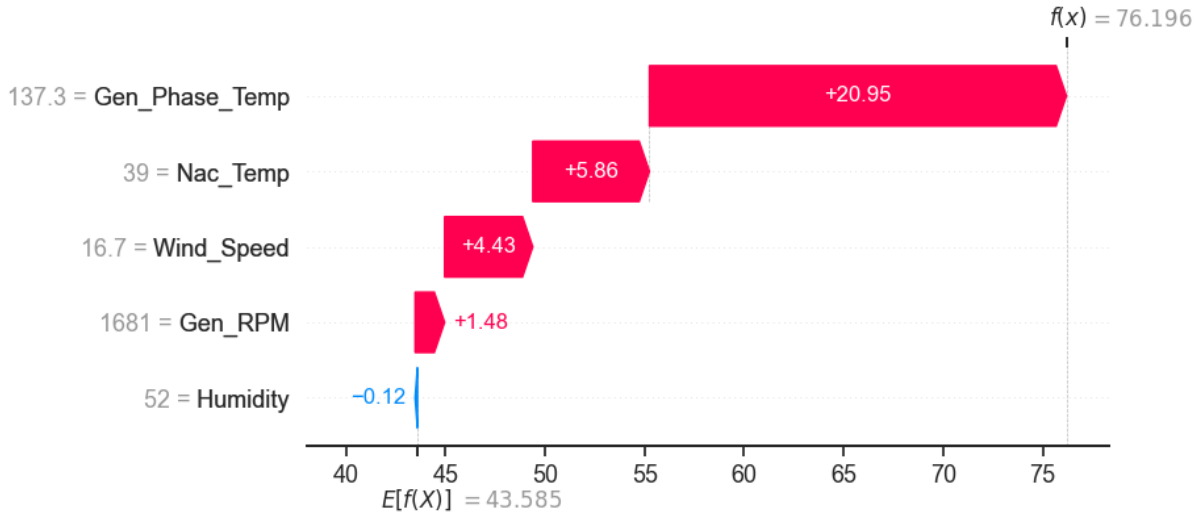

FIGURE 17. Local explanation on June 7, 2017, 23:10:00 by waterfall plot.

The analysis provides a reasonable explanation for the predicted bearing temperature. A high generator temperature (137°C) increases the predicted bearing temperature significantly (+20.95°C) and the remaining features also contribute to bringing the predicted bearing temperature to 76.2°C.

6 CONCLUSIONS

This paper presents a simple and robust methodology for making a machine learning based model for detecting faults in wind turbine generator bearing. In this model, the predicted bearing temperature is compared against the actual bearing temperature and a significant difference between the two indicates a possibility of fault(s) in the bearing or its lubrication. Either of these may result in failure. As a case study, the idea has been demonstrated on a generator bearing, using real-life SCADA data. The results show that it is possible to detect potential failure well in advance. This knowledge can be used for planning maintenance.

Four different machine learning algorithms, Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost, have been evaluated and XGBoost has been found to be the most suitable algorithm for the task.

The paper also examines the role of five features, generator shaft / bearing rotational speed, generator temperature, wind speed, nacelle air humidity, and nacelle temperature, on the predicted bearing temperature. Out of these, the generator temperature has been found to play the major role, followed by the wind speed and nacelle temperature. Bearing rotational speed and relative humidity of nacelle air play minor roles.

To take the research work further, the following tasks have been identified:

(a) analysis of data from different wind turbines,
(b) testing of other machine learning / artificial intelligence algorithms, like artificial neural networks,
(c) consideration of the impact of more features,
(d) use of other interpretable machine learning tools such as Individual Conditional Expectation (ICE) plots [Goldstein, Kapelner et al. 2015] and LIME (Local interpretable model-agnostic explanations (LIME) [Ribeiro, Singh et al. 2016],
(e) expanding the scope from component to system level.

REFERENCES

Adadi, A. and Berrada, M. (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI), IEEE Access, 6, pp. 52138-52160. doi: 10.1109/ACCESS.2018.2870052.

Arabian-Hoseynabadi, H. and Oraee, H. and Tavner, P. J. (2010). Failure Modes and Effects Analysis (FMEA) for wind turbines, *International Journal of Electrical Power and Energy Systems*, 32 (7). pp. 817-824.

Belyadi, H. and Haghighat, A. (2021) Machine Learning Guide for Oil and Gas Using Python: A Step-by-Step Breakdown with Data, Algorithms, Codes, and Applications. Gulf Professional Publishing.

Bergstra, J. and Bengio, Y. (2012) Random search for hyper-parameter optimization, Journal of machine learning research, 13(2). Available at: https://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

Boukhriss, M., Khalifa, Z. and Ghribi, R. (2013) Study of thermophysical properties of a solar desalination system using solar energy, Desalination and Water Treatment, 51, pp. 1290-1295. doi: 10.1080/19443994.2012.714925.

EDP (2017) Data. Available at: https://www.edp.com/en/innovation/open-data/data (Accessed: 14.06.2023).

Ekanayake, I., Meddage, D. and Rathnayake, U. (2022) A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP), Case Studies in Construction Materials, 16. doi: 10.1016/j.cscm.2022.e01059.

Ester, M. et al. (1996) A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Second International Conference on Knowledge Discovery and Data Mining (KDD'96). Proceedings of a conference held August 2-4. pp. 226-231.

Goldstein, A., A. Kapelner, J. Bleich and E. Pitkin (2015). "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation." journal of Computational and Graphical Statistics 24(1): 44-65.

Gupta, P.K. (2013). Analytical Modeling of Rolling Bearings. In: Wang, Q.J., Chung, YW. (eds) Encyclopedia of Tribology. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-92897-5_741

IEA (2021). Net Zero by 2050. Paris. Available at: https://www.iea.org/reports/net-zero-by-2050 (Accessed: 26.09.2022).

ISO. (2017). "ISO 15243:2017 Rolling bearings — Damage and failures — Terms, characteristics and causes." Retrieved 13.06.2023, from https://www.iso.org/standard/59619.html.

Kahrobaee, S. and Asgarpoor, S. (2011). Risk-Based Failure Mode and Effect Analysis for Wind Turbines (RB-FMEA), Faculty Publications from the Department of Electrical and Computer Engineering. 172.

Kandukuri, S. T., Klausen, A., Karimi, H. R. and Robbersmyr, K. G. (2016). A Review of Diagnostics and Prognostics of Low-Speed Machinery Towards Wind Turbine Farm-Level Health Management, *Renewable and Sustainable Energy Reviews*, 53, pp. 697-708.

Koukoura, S., Scheu, M. N. and Kolios, A. (2021). Influence of Extended Potential-To-Functional Failure Intervals Through Condition Monitoring Systems on Offshore Wind Turbine Availability, *Reliability Engineering and System Safety*, 208, 107404.

Luengo, M. M. and Kolios, A. (2015). Failure Mode Identification and End of Life Scenarios of Offshore Wind Turbines: A Review, *Energies*, 8, 8339-8354

Lundberg, S. M., G. G. Erion and S.-I. Lee (2018). "Consistent individualized feature attribution for tree ensembles." arXiv e-prints.

Lundberg, S. M. and Lee, S.-I. (2017) A unified approach to interpreting model predictions, Advances in neural information processing systems, 30. doi: 10.48550/arXiv.1705.07874.

Mahesh, B. (2020) Machine Learning Algorithms - A review, International Journal of Science and Research (IJSR), 9(1), pp. 381-386. Available at: https://www.ijsr.net/archive/v9i1/ART20203995.pdf.

Molnar, C., Casalicchio, G. and Bischl, B. (2020) Interpretable Machine Learning – A Brief History, State-of-the-Art and Challenges, in Koprinska, I., et al. (ed.) ECML PKDD 2020 Workshops, Cham. Springer International Publishing, pp. 417-431.

Nilsson, J. and Bertling, L. (2007) Maintenance management of wind power systems using condition monitoring systems—life cycle cost analysis for two case studies, IEEE Transactions on energy conversion, 22(1), pp. 223-229.

Probst, P., Boulesteix, A.-L. and Bischl, B. (2019) Tunability: Importance of hyperparameters of machine learning algorithms, The Journal of Machine Learning Research, 20(1), pp. 1934-1965. doi: 10.48550/arXiv.1802.09596.

Ribeiro, M. T., S. Singh and C. Guestrin (2016). " Why should i trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. A. f. C. Machinery. San Francisco, CA (United States), Association for Computing Machinery: 1135-1144.

Sankar S., Nataraj M. and Prabhu Raja V. (2012). Failure Analysis of Bearing in Wind Turbine Generator Gearbox, *Journal of Information Systems and Communication*, 3 (1), pp.-302-309.

Shafiee, M. and Dinmohammadi, F. (2014). An FMEA-Based Risk Assessment Approach for Wind Turbine Systems: A Comparative Study of Onshore and Offshore, *Energies*, 7, pp. 619-642.

Stehly, T., Beiter, P. and Duffy, P. (2020) 2019 cost of wind energy review. (NREL/TP-5000-78471). Golden, CO (United States): National Renewable Energy Laboratory. Available at: https://www.nrel.gov/docs/fy21osti/78471.pdf.

Stetco, A. et al. (2019) Machine learning methods for wind turbine condition monitoring: A review, Renewable Energy, 133, pp. 620-635. doi: 10.1016/j.renene.2018.10.047.

Tchakoua, P., Wamkeue, R., Ouhrouche, M., Slaoui-Hasnaoui, F., Tameghe, T. A., and Ekemb, G. (2014). Wind Turbine Condition Monitoring: State-of-the-Art Review, New Trends, and Future Challenges, *Energies*, 7, pp. 2595-2630.

Vilone, G. and Longo, L. (2020) Explainable artificial intelligence: a systematic review, arXiv preprint. doi: 10.48550/arXiv.2006.00093.

Wiggelinkhuizen, E., L. Rademakers, T. Verbruggen, S. Watson, J. Xiang, G. Giebel, E. Norton, M. Tipluica, A. Christensen and E. Becker (2007). Conmow Final Report, Energy research Centre of the Netherlands.

Zhang, X., Sun, L., Sun, H., Guo, Q. and Bai, X. (2016). Floating Offshore Wind Turbine Reliability Analysis Based on System Grading and Dynamic FTA, *Journal of Wind Engineering and Industrial Aerodynamics*, pp. 15421–33.

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: