frontiers | Frontiers in Energy Research

TYPE Original Research
PUBLISHED 13 December 2023
DOI 10.3389/fenrg.2023.1284676

OPEN ACCESS

EDITED BY
Juan P. Amezquita-Sanchez,
Autonomous University of Queretaro,
Mexico

REVIEWED BY
Ryad Zemouri,
Conservatoire National des Arts et
Métiers (CNAM), France
Marianne Rodgers,
Wind Energy Institute of Canada, Canada

*CORRESPONDENCE
Maneesh Singh,
✉ maneesh.singh@hvl.no

RECEIVED 28 August 2023
ACCEPTED 29 November 2023
PUBLISHED 13 December 2023

CITATION
Bindingsbø OT, Singh M, Øvsthus K and Keprate A (2023), Fault detection of a wind turbine generator bearing using interpretable machine learning.
Front. Energy Res. 11:1284676.
doi: 10.3389/fenrg.2023.1284676

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

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

[1]Department of Mechanical and Marine Engineering, Western Norway University of Applied Sciences, Bergen, Norway, [2]Department of Mechanical, Electrical and Chemical Engineering, Oslo Metropolitan University, Oslo, Norway

Introduction: 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 6-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.

Methods: 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.

Results: Consistent abnormal differences between predicted and actual temperatures may be attributed to the degradation and presence of a fault in the bearing.

Discussion: 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 et al., 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 and Asgarpoor 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.

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 et al., 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 (Mahesh, 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 et al., 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 et al., 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 et al., 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 Proposed fault detection methodology

2.1 Description of the 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 and cracking (ISO, 2017), thereby resulting in the deterioration of the components and ultimately failure (Sankar et al., 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.

2.2 Feature selection

As discussed in the previous section, temperature is a commonly measured parameter to monitor the health of a bearing, because it is

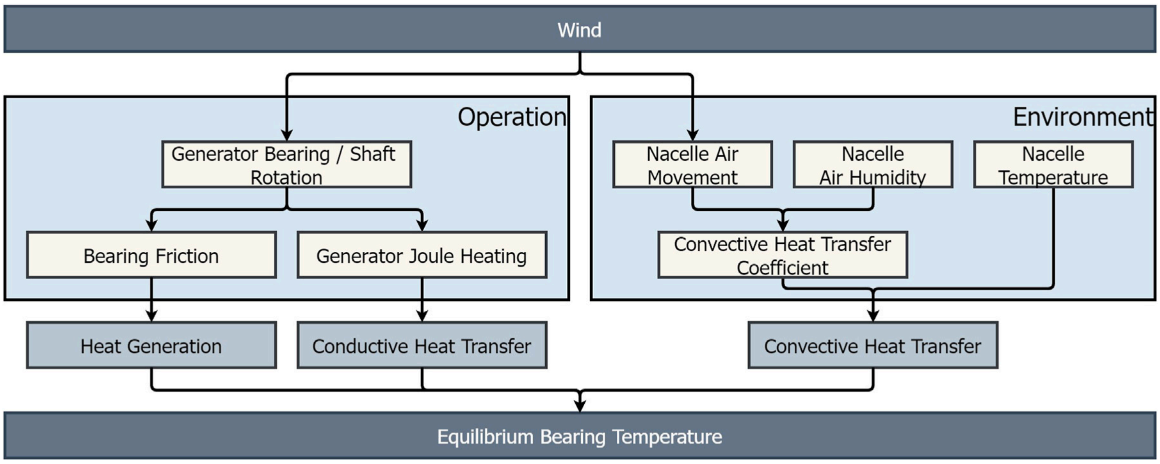

FIGURE 1
Flowchart showing the heat transfers taking place in bearings.

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 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.

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.

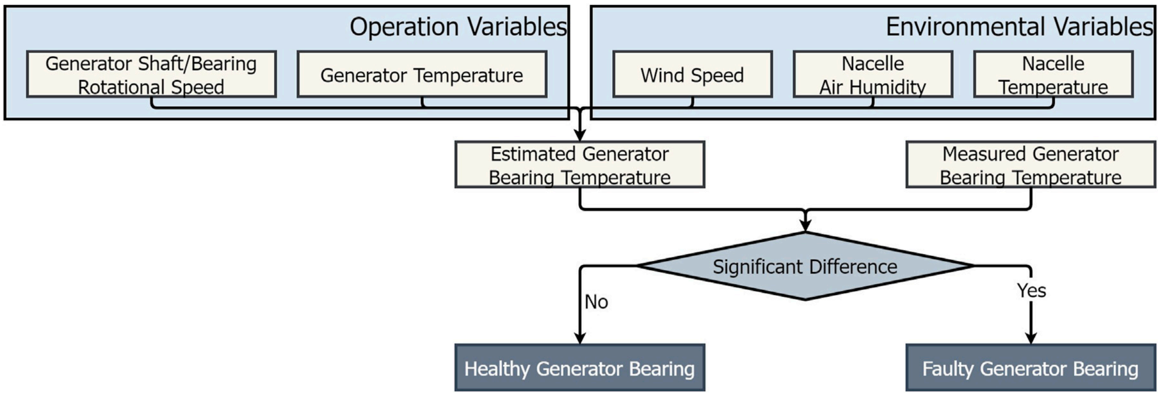

FIGURE 2
Flowchart showing the proposed fault detection methodology.

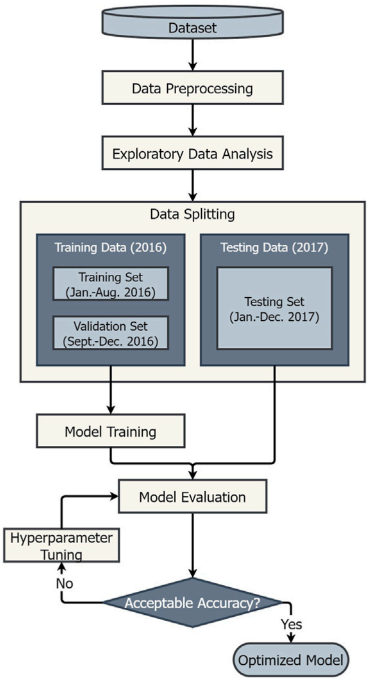

FIGURE 3
Flowchart for developing the proposed interpretable machine learning model.

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.

2.3 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.

2.3.1 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)

The short-listing of suitable algorithms have been carried out based on two key criteria.

- Firstly, the algorithms that demonstrate high compatibility with interpretable machine learning tools (example, SHAP) have been prioritized. This consideration is crucial as it ensures that the developed models are not just black boxes, rather their decision-making processes can be understood and explained. This aspect is particularly important for applications where transparency and trust in the model's predictions are paramount.
- Secondly, one representative algorithm from each of the aforementioned categories—linear models, tree-based models, boosting models, and support vector machines have been deliberately selected. This enables comparison regarding their behavior and strengths.

These selection criteria help to identify the most effective algorithm that not only delivers high accuracy but also aligns with the interpretability and applicability requirements of our project. Thus, out of the above mentioned algorithms, four algorithms—Linear Regression (LR), Random Forest (RF), Support Vector Regression (SVR) and XGBoost—have been shortlisted for further testing.

2.3.2 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.

2.3.3 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 metrics. These metrics depict the correlation between all the possible pairs of values and is a powerful tool to identify and visualize patterns in data.

2.3.4 Data splitting—training, validation and testing data

In supervised machine learning tasks, best practice is to split data into three independent data sets: training set, validation set and test set.

2.3.5 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.

2.3.6 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.

2.3.7 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 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.

2.4 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.

3 Illustrative case study

3.1 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-min 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 52,445 and 52,294 for 2016 and 2017, respectively. Table 1 shows the selected features and target used for developing the model.

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 20 August 2017, at 08:08:00, and damage of generator shortly afterwards on 21 August 2017, at 16:47:00 (Table 2). The downtime caused by the generator failures is highlighted in green in

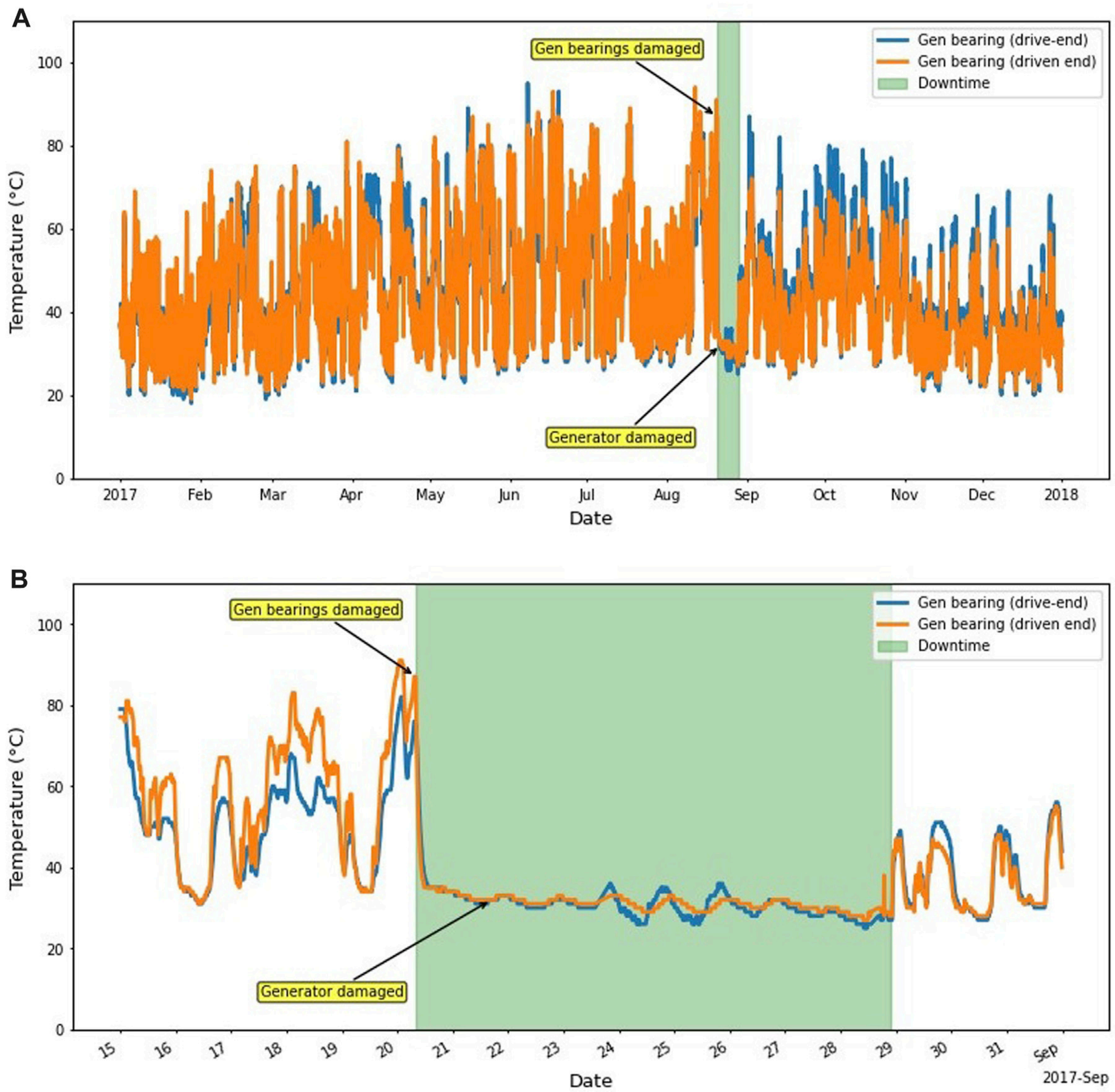

FIGURE 4
Bearings temperature during the bearing and generator failures in **(A)** 2017 and **(B)** August 2017.

Figure 4 and lasts from 20 August 2017, at 08:10:00 until 28 August 2017, at 21:50:00. The model shall attempt to predict these failures.

3.2 Data preprocessing

3.2.1 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 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 30 April 2016 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

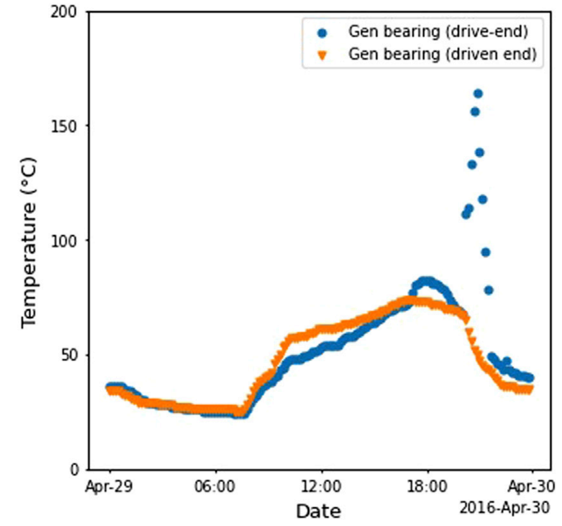

FIGURE 5
Effect of faulty sensors on recorded temperature of bearings.

Q1 and Q3, also called the interquartile range (IQR), are calculated using Eq. 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.

$$IQR = Q3 - Q1 \quad (1)$$

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

3.2.2 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°C 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 3 January 2017 to 6 May 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 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.

3.3 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.

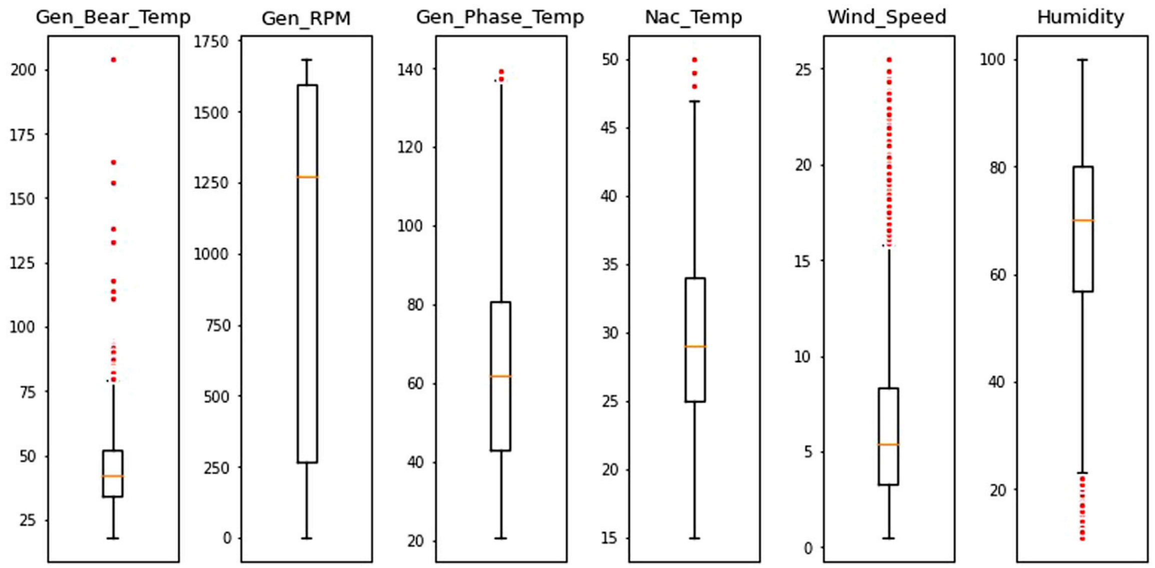

FIGURE 6
Box plot of SCADA signals.

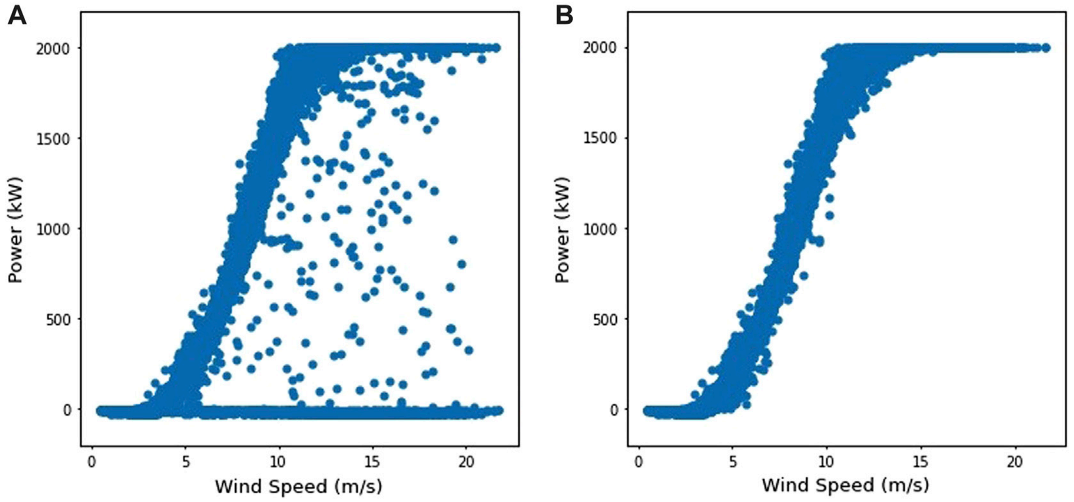

FIGURE 7
Plot of power generated versus wind speed using data of training period **(A)** Using raw. **(B)** Using data after cleaning outliers.

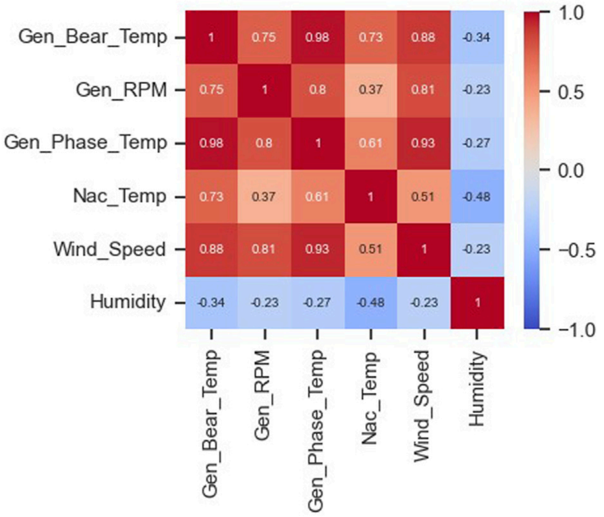

FIGURE 8
Pearson correlation matrix of the input features.

3.3.1 Effect of generator shaft/bearing rotational speed on bearing temperature

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

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

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*).

3.3.2 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*).

3.3.3 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*).

3.3.4 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

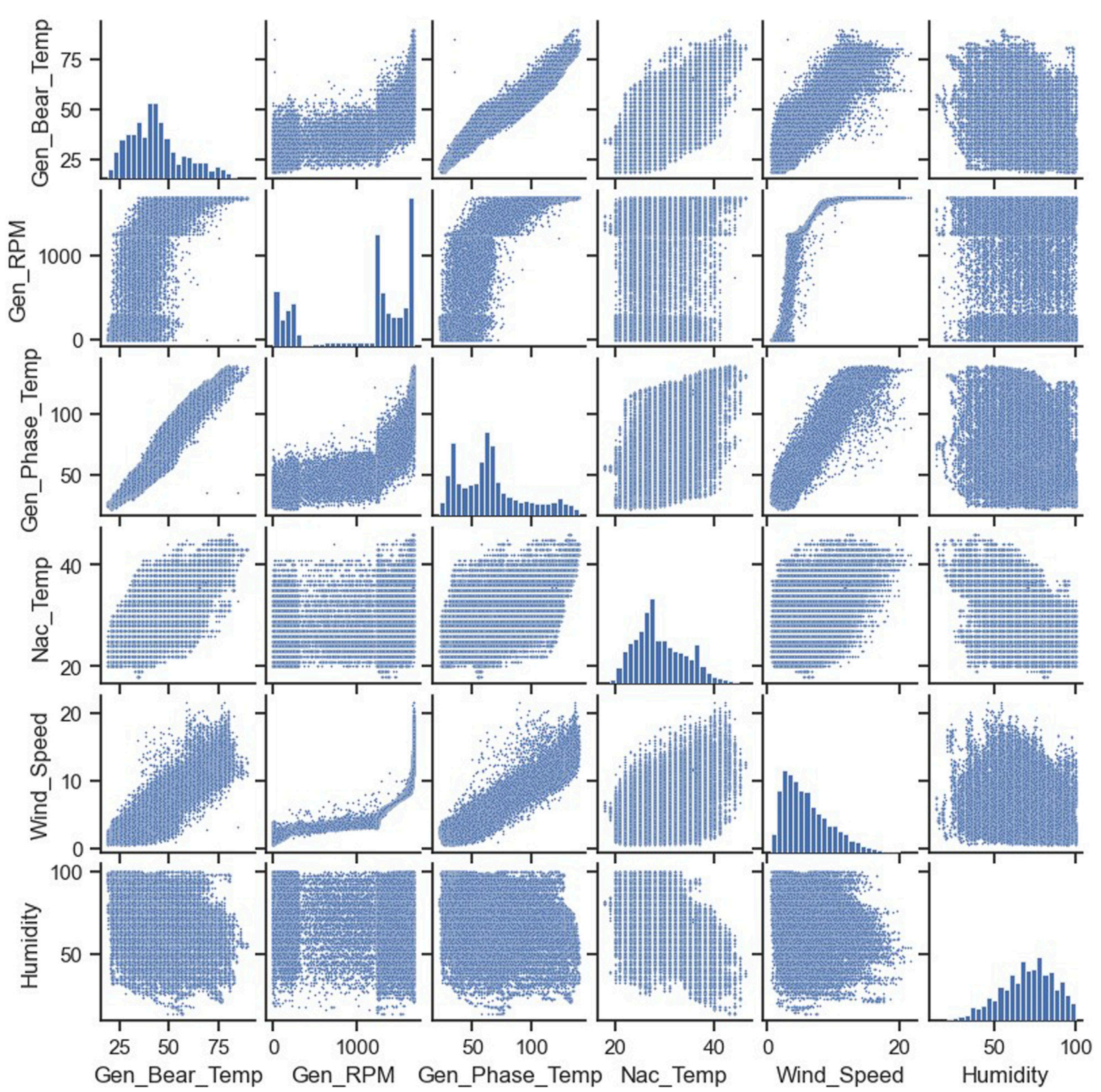

FIGURE 9
Pairwise relationships between input features.

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*).

3.3.5 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*).

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

TABLE 1 Selected features and target for developing the model.

Variable	Description	Units
Timestamp	10-min 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	°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 2 Failure log for Turbine Number 7 ("T07").

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

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

2016 data is split into two parts—training data and validation data. The data from the first 8 months is used to train the algorithms, while the data from the last 4 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 h 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).

3.5 Model training

The four shortlisted Ralgorithms—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.

3.6 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 1-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 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

TABLE 5 Hyperparameter search range.

Hyperparameter	Search range	Optimal value
n_estimators	[200, 400, 600, 800, 1,000]	1,000
max_depth	[3, 4, 5, 6, 7, 8, 9]	4
learning_rate	[0.1, 0.05, 0.01]	0.05

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

While the SVR shows good performance in scoring metrics, it is important to note that the algorithm demands significantly more time for model fitting and prediction compared to XGBoost. This increased computational time, especially while dealing with large datasets or in real-time analysis, often makes SVR unsuitable. In contrast, XGBoost with its efficient handling of large data and faster execution emerges as a more practical choice.

Upon detailed evaluation, XGBoost has been identified as the most suitable algorithm because it strikes an optimal balance between accuracy and computational efficiency. Furthermore, this algorithm can be fine-tuned using hyperparameter tuning techniques, thereby, enhancing its performance. This process involved systematically adjusting the algorithm's parameters to find the combination that yields the best results in terms of prediction accuracy and processing speed. The fine-tuned XGBoost model demonstrates a marked improvement in performance, confirming its suitability for the required predictive modeling tasks.

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 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 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.

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.

3.8 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 1 January to 15 January 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 or approximately 3.5°C) temperature range.

3.9 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

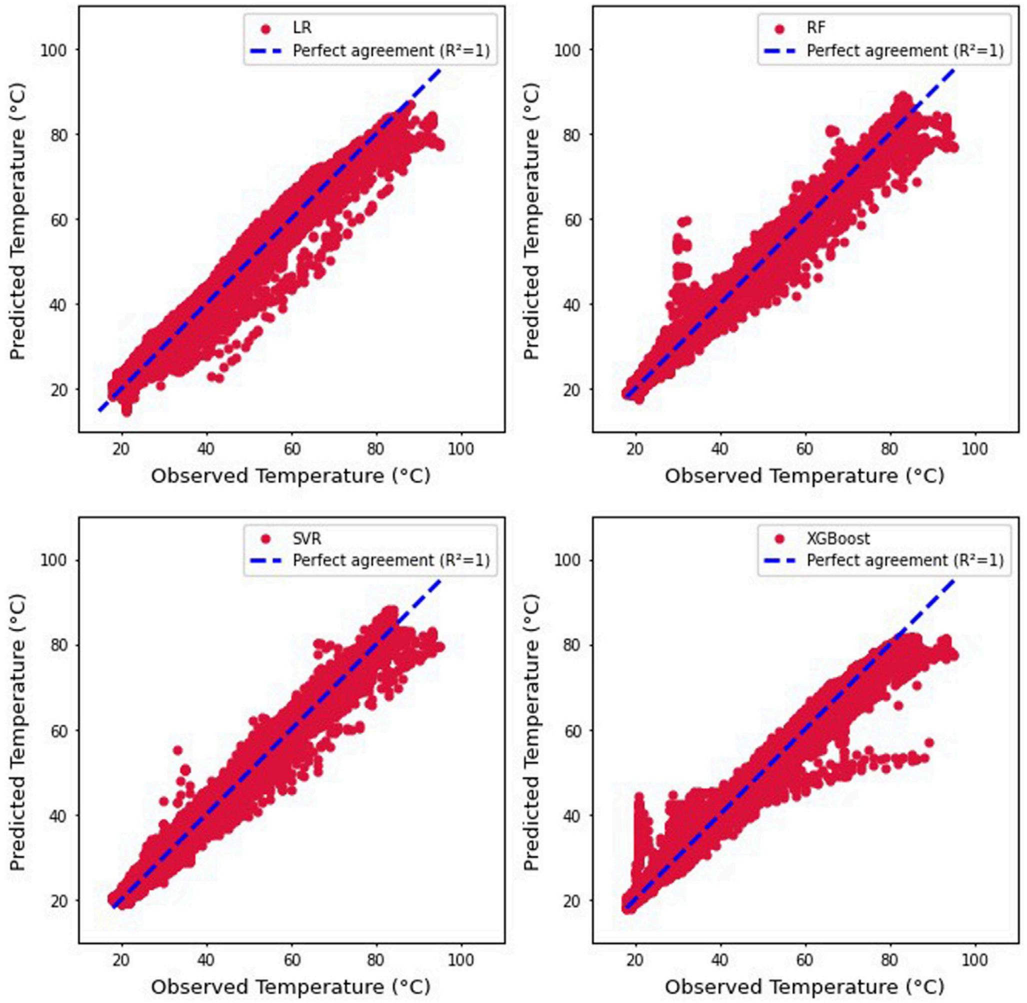

FIGURE 10
Predicted and observed temperatures for all models.

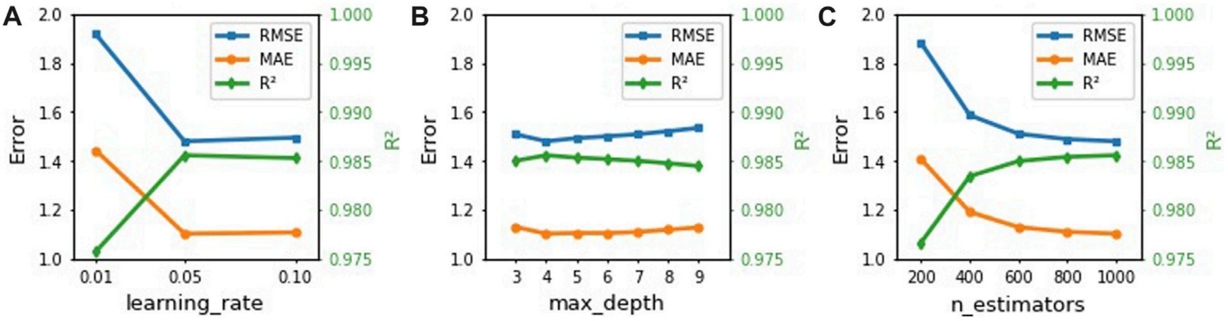

FIGURE 11
Model impact changing **(A)** *learning_rate*, **(B)** *max_depth* and **(C)** *n_estimators*.

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

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

3.10 Fault detection and recommendation for rescheduling maintenance plan

Figure 13 shows the plots of the actual and predicted values for the period from 7 June to 23 June 2017. During this period there are

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

times when the actual bearing temperature exceeds the predicted value by more than two standard deviations (approximately 3.5°C) over significantly long periods, and this is highlighted in green. For example, on 7 June 2017, the actual value reaches 95°C whereas the model prediction is 76°C, a difference of 19°C.

After 7 June 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 2 months later on 20 August 2017.

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 it is working. Since XGBoost is a tree-based model, the Tree SHAP algorithm proposed by Lundberg et al. (2018) for tree ensembles can be used to calculate the SHAP values that could be used for the interpretation of the working.

4.1 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

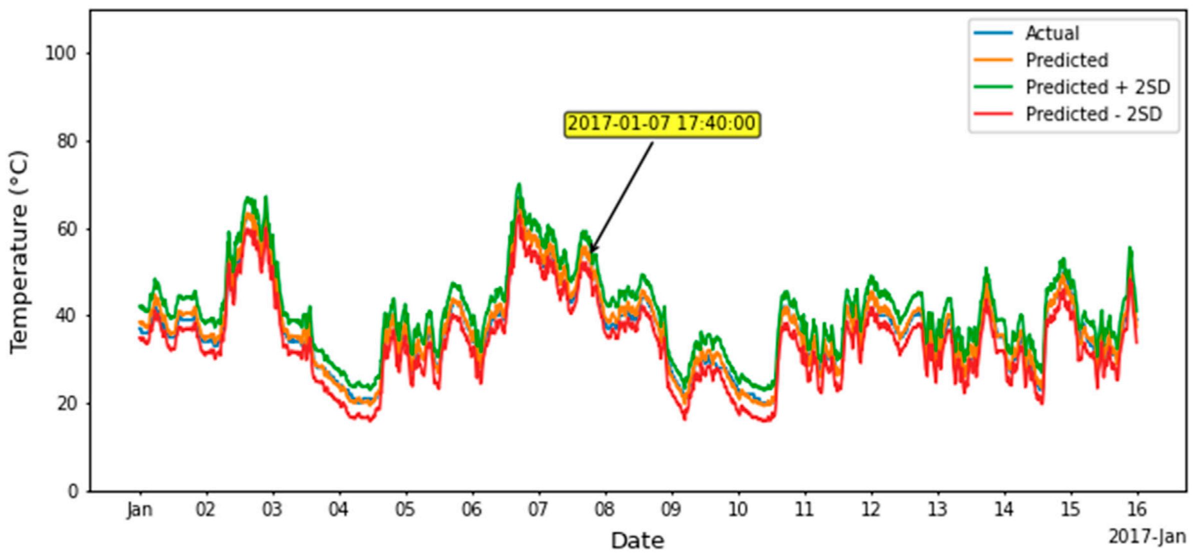

FIGURE 12
Actual and predicted temperatures of generator bearing for the period January 1 to 15 January 2017.

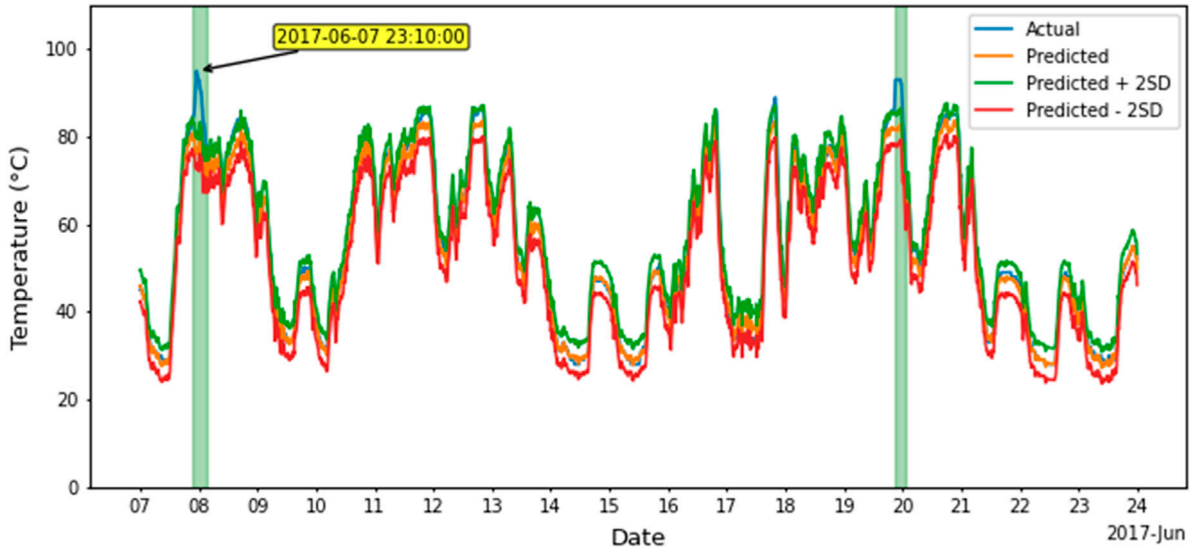

FIGURE 13
Actual and predicted temperatures of generator bearing for the period 7 June to 23 June 2017.

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.

- 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 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.
- Nacelle Temperature—Nacelle temperature has an increased positive impact in the temperature range 20°C–45°C.

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.

4.2.1 Explanation of prediction for 7 January 2017

Figure 12 shows the plots of the actual and predicted values for the period of 1 January to 15 January 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 (7 January 2017, 17:40:00) has been randomly selected for local explanation.

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

- 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°C) is relatively small.
- Generator Temperature—The high generator phase temperature (89.3°C) has by far the most significant positive influence (+8.52°C) on the bearing temperature.
- Wind Speed—Wind speed makes relatively small positive effect (+2.02°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°C) and hence a greater

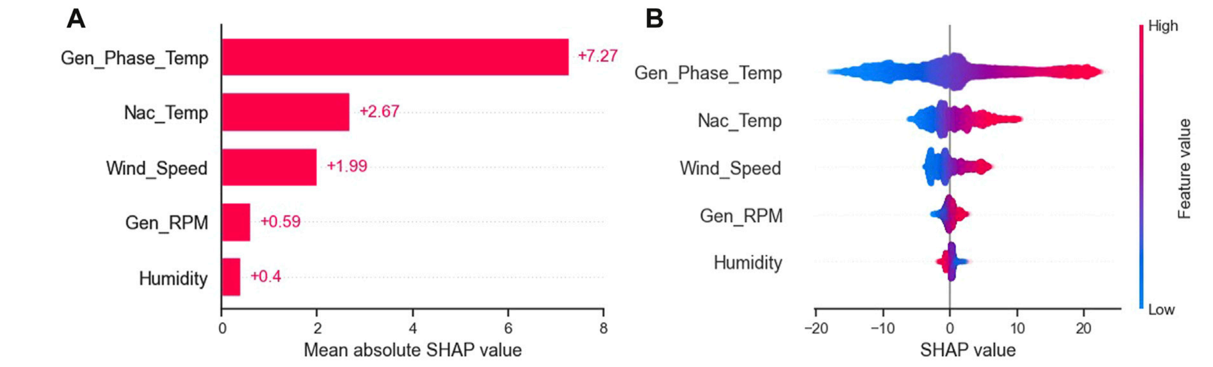

FIGURE 14
(A) Mean absolute SHAP value per feature. **(B)** Matrix plot of SHAP values for different features.

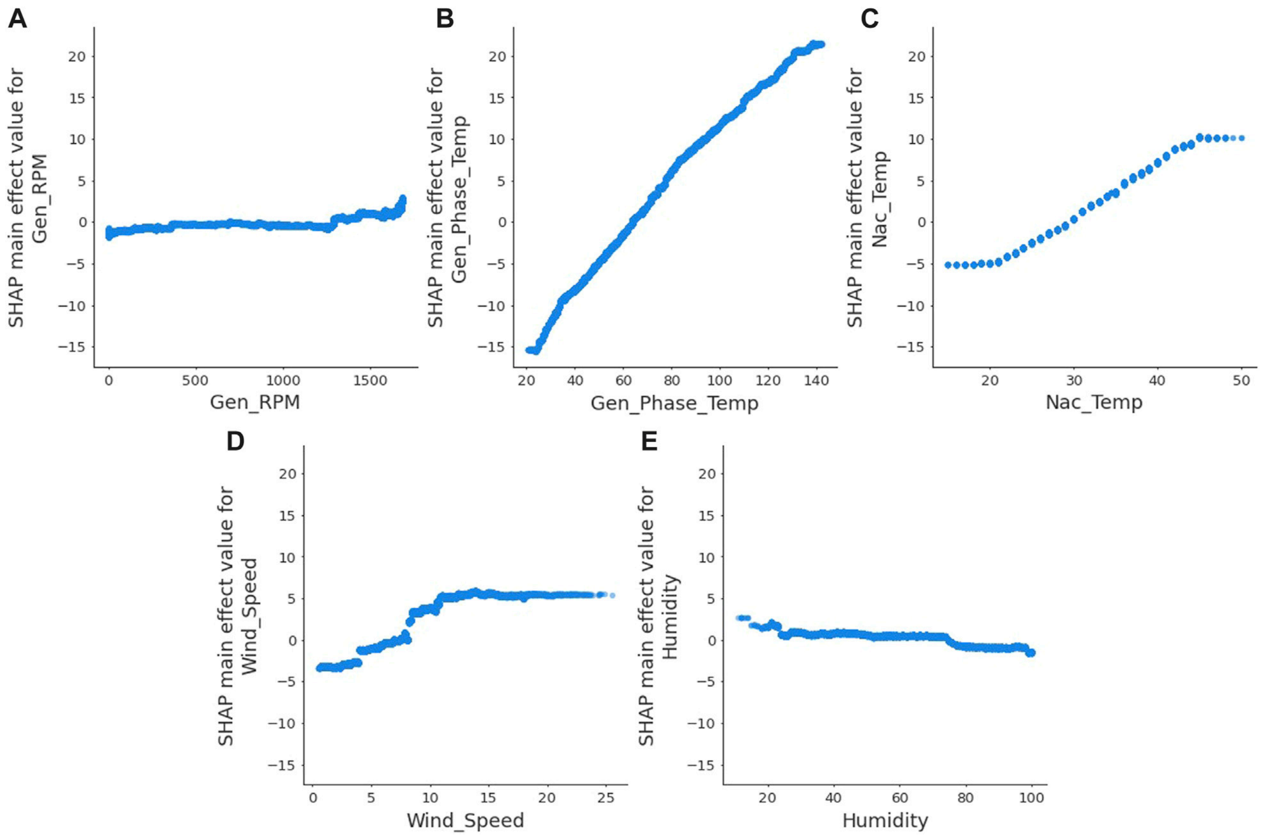

FIGURE 15
SHAP main effects plot for **(A)** generator rpm, **(B)** generator phase temperature, **(C)** nacelle temperature, **(D)** wind speed and **(E)** humidity.

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°C) affect the predicted temperature value, because relative humidity itself does not have any significant role.
- Nacelle Temperature—The nacelle temperature (30°C) is close to the average annual temperature, ranging between 15°C and 50°C, and hence does not play a significant role (−0.01°C) in the fall of temperature on predicted value.

4.2.2 Explanation of prediction for 7 June 2017

Figure 13 shows the plots of the actual and predicted values for the period 7 June to 23 June 2017. On 7 June 2017 (Summer), the environmental and operating temperatures are quite different

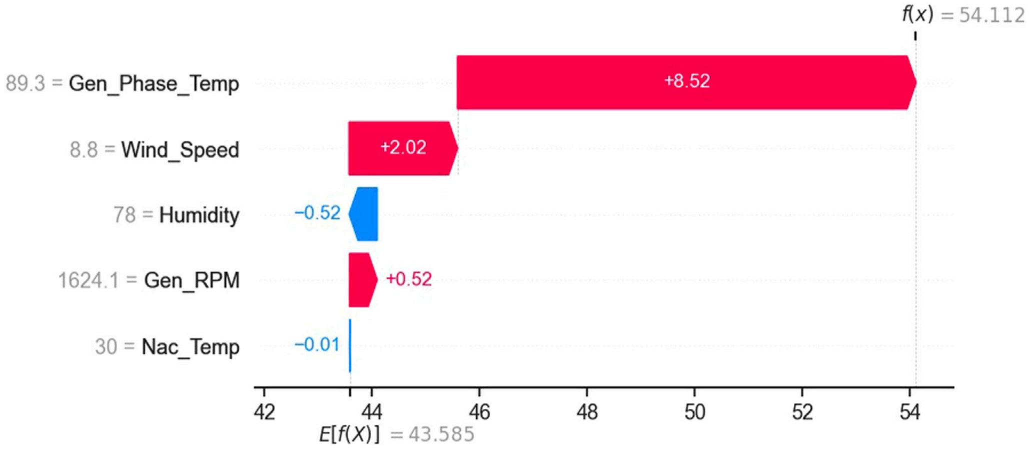

FIGURE 16
Local explanation on 7 January 2017, 17:40:00 by waterfall plot.

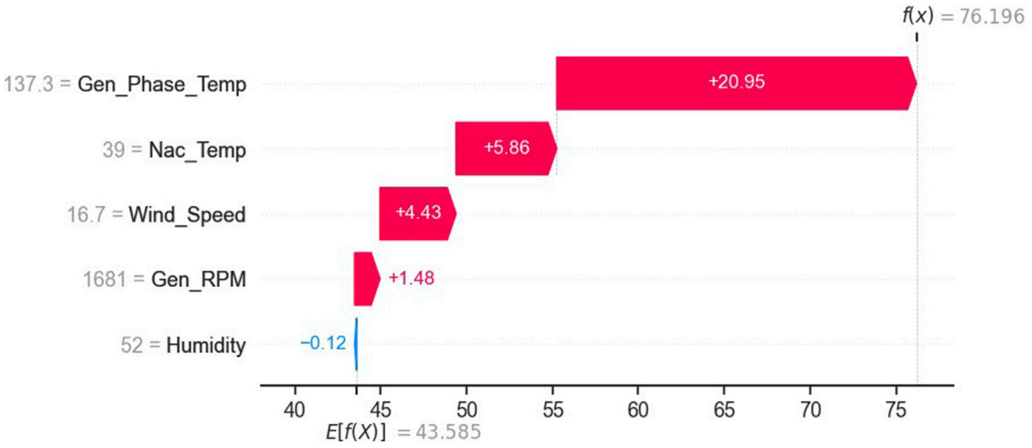

FIGURE 17
Local explanation on 7 June 2017, 23:10:00 by waterfall plot.

from those of 7 January 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 (7 January 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°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°C) and this significantly (+20.95°C) raises the temperature of the bearing.
- Wind Speed—Compared to the previous case, wind speed gives relatively higher positive effect (+4.43°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°C) effect on the predicted temperature value.
- Nacelle Temperature—Compared to the previous case, the nacelle temperature (39°C) is 9°C higher than the previous case, and hence there is significantly (+5.86°C) higher effect on the predicted temperature.

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.

5 Conclusion

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 et al., 2015) and LIME (Local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016),
(e) expanding the scope from component to system level.

Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found below: For training the link is: https://www.edp.com/en/wind-turbine-scada-signals-2016, For testing the link is: https://www.edp.com/en/innovation/open-data/wind-turbine-scada-signals-2017.

Author contributions

OB: Writing–original draft. MS: Writing–review and editing. KØ: Writing–review and editing. AK: Writing–review and editing.

Funding

The authors declare that no financial support was received for the research, authorship, and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

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

Arabian-Hoseynabadi, H., Oraee, H., and Tavner, P. J. (2010). Failure modes and effects analysis (FMEA) for wind turbines. *Int. J. Electr. Power Energy Syst.* 32 (7), 817–824. doi:10.1016/j.ijepes.2010.01.019

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.* Houston: Gulf Professional Publishing.

Bergstra, J., and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* 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 Water Treat.* 51, 1290–1295. doi:10.1080/19443994.2012.714925

EDP (2017). Data. Available at: https://www.edp.com/en/innovation/open-data/data (Accessed June 14, 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 Stud. Constr. Mater.* 16, e01059. doi:10.1016/j.cscm.2022.e01059

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). "A density-based algorithm for discovering clusters in large spatial databases with noise," in Second International Conference on Knowledge Discovery and Data Mining (KDD'96). Proceedings of a conference held, August 2-4, 226–231.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. *J. Comput. Graph. Statistics* 24 (1), 44–65. doi:10.1080/10618600.2014.907095

Gupta, P. K. (2013). "Analytical modeling of rolling bearings," in *Encyclopedia of tribology*. Editors Q. J. Wang and Y. W. Chung (Boston, MA: Springer). doi: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 August 26, 2022).

ISO (2017). ISO 15243:2017 Rolling bearings — damage and failures — terms, characteristics and causes. Available at: https://www.iso.org/standard/59619.html (Accessed June 13, 2022).

Kahrobaee, S., and Asgarpoor, S. (2011). "Risk-based failure mode and effect analysis for wind turbines (RB-FMEA)," in 2011 North American Power Symposium, Boston, MA, USA, 04-06 August 2011.

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. *Renew. Sustain. Energy Rev.* 53, 697–708. doi:10.1016/j.rser.2015.08.061

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. *Reliab. Eng. Syst. Saf.* 208, 107404. doi:10.1016/j.ress.2020.107404

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. doi:10.3390/en8088339

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

Lundberg, S. M., and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Adv. neural Inf. Process. Syst.* 30. doi:10.48550/arXiv.1705.07874

Mahesh, B. (2020). Machine learning algorithms - a review. *Int. J. Sci. Res. (IJSR)* 9 (1), 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 *ECML PKDD 2020 workshops*. Editor I. Koprinska (Cham: Springer International Publishing), 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 Trans. energy Convers.* 22 (1), 223–229. doi:10.1109/tec.2006.889623

Probst, P., Boulesteix, A.-L., and Bischl, B. (2019). Tunability: importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.* 20 (1), 1934–1965. doi:10.48550/arXiv.1802.09596

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why should i trust you?," in *Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. Editor 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. *J. Inf. Syst. Commun.* 3 (1), 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, 619–642. doi:10.3390/en7020619

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., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., et al. (2019). Machine learning methods for wind turbine condition monitoring: a review. *Renew. Energy* 133, 620–635. doi:10.1016/j.renene.2018.10.047

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

Wiggelinkhuizen, E., Rademakers, L., Verbruggen, T., Watson, S., Xiang, J., Giebel, G., et al. (2007). *Conmow final report*. Netherlands: 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. *J. Wind Eng. Industrial Aerodynamics* 154, 15421–15433. doi:10.1016/j.jweia.2016.04.005