



Enhanced Bearing Fault Diagnosis Through Trees Ensemble Method and Feature Importance Analysis

Amir Alhams¹ · Ahmed Abdelhadi² · Yousif Badri³ · Sadok Sassi¹ · Jamil Renno¹

Received: 16 February 2024 / Revised: 18 April 2024 / Accepted: 23 April 2024
© The Author(s) 2024

Abstract

Purpose This research introduces a groundbreaking method for bearing defect detection. It leverages ensemble machine learning (ML) models and conducts comprehensive feature importance analysis. The key innovation is the training and benchmarking of three tree ensemble models—Decision Tree (DT), Random Forest (RF), and Extreme Gradient Boosting (XGBoost)—on an extensive experimental dataset (QU-DMBF) collected from bearing tests with seeded defects of varying sizes on the inner and outer raceways under different operating conditions.

Method The dataset was meticulously prepared with categorical variable encoding and Min–Max data normalization to ensure consistent class distribution and model accuracy. Implementing the ML models involved a grid search method for hyperparameter tuning, focusing on reporting the models' accuracy. The study also explores applying ensemble methods and using supervised and unsupervised learning algorithms for bearing fault detection. It underscores the value of feature importance analysis in understanding the contributions of specific inputs to the model's performance. The research compares the ML models to traditional methods and discusses their potential for advanced fault diagnosis in bearing systems.

Results and Conclusions The XGBoost model, trained on data from actual bearing tests, outperformed the others, achieving 92% accuracy in detecting bearing health and fault location. However, a deeper analysis of feature importance reveals that the models weigh certain experimental conditions differently—such as sensor location and motor speed. This research's primary novelties and contributions are comparative evaluation, experimental validation, accuracy benchmarking, and interpretable feature importance analysis. This comprehensive methodology advances the bearing health monitoring field and has significant practical implications for condition-based maintenance, potentially leading to substantial cost savings and improved operational efficiency.

Keywords Bearing fault diagnosis · Machine learning · Decision trees · Random forest · XGBoost · Vibration signals

Introduction

Ball bearings are essential elements in a rotating machine. They enable rotational or linear motion while minimizing friction and supporting dynamic mechanical loads. Defects and faults inside these bearings may lead to disastrous failures. Some defects in bearing appear gradually over time,

while others occur suddenly with little warning. Therefore, early identification and diagnosis of defects in ball bearings can contribute to increasing machinery uptime and form a part of a preventative maintenance strategy. For example, Frosini et al. mentioned that 40–50% of induction motor failures in the industry happened due to damages occurring in bearings [1]. Therefore, condition monitoring (CM) of ball bearings plays a significant role in early fault detection and is considered an integral part of any preventative maintenance plan.

The persistent rotational movement of bearings during extended periods of operation causes friction, elevated temperatures, and increased vibrations, leading to localized and distributed primary defect categories. Localized defects are defined as single-point faults, such as cracks, pits, spalls, and even small particles in the lubrication fluid [2]. These

✉ Yousif Badri
ybad223@aucklanduni.ac.nz

¹ Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, Qatar

² Department of Engineering Science, University of Auckland, Auckland, New Zealand

³ Mechanical and Mechatronics Engineering Department, University of Auckland, Auckland, New Zealand

defects are usually identified according to their locations (on the outer race, inner race, or the rolling element itself) [3]. The second type of bearing faults are called extended or distributed defects [4–6]. They refer to imperfections that are spread out over the surface of any of the bearing components. These defects, usually related to manufacturing or installation mistakes, can include surface roughness and waviness [6]. They can also arise due to the progression of minor localized defects. Surface defects (localized or distributed) generate undesirable frequencies during the rotational motion of their supporting mechanisms and can also excite them at one of their resonances. However, they are usually hard to diagnose if represented only in the frequency domain. The presence of multiple simultaneous defects at various locations in the bearing may bring more complications when interrogating the spectrum. Therefore, investigating and studying these faults and how to detect them is especially important.

Different methods were developed to detect and diagnose defects in bearings. Among these techniques are vibration, acoustic emissions (AE), or motor's current variation [7, 8]. Bearings tend to generate vibration and noise due to the presence of defects, which works as an obstacle that impedes the smooth motion of the bearing balls. However, even if the geometry of the bearing is perfect, it generates vibration due to continuous changes in the total stiffness. This change in stiffness is due to the finite number of elements that carry the total load [9]. Singh et al. [10] stated that the generated vibration signals due to defects are caused by the restressing of the rolling elements. Consequently, when the ball hits the end of a defect, the impulse generates vibration signals and enlarges the defect's size. In contrast, other explanations define the short force impulse vibration as the ball's compression between the inner and outer races [11, 12]. In either case, the spectrum of a defect-free ball bearing differs from that of a ball bearing with (one or more) defect(s).

Different attempts were made to develop numerical or mathematical models that could predict the dynamic behavior of damaged bearings. However, these models always require validation compared to experimental results. Therefore, damaged bearings with specific types and sizes of damage are always needed. Using manufacturing techniques, such as Electric Discharge Machining (EDM) or punching, the damage could be intentionally seeded on the inner or outer raceways of bearings. The defective bearings would then be mounted on an experimental test rig to study their responses. One of the fault-testing approaches is run-to-failure, which consists of running the bearing under abnormal conditions, such as over-loading, over-speeding, or poor lubrication, until defects occur [9]. Vibration data is periodically or continuously recorded to study the time evolution of the vibration signal. Another approach is to seed several defects on different bearings [13] and test them

separately under the same operating conditions to compare their readings with signals extracted from healthy bearings. The latter approach is adopted in this study. Subsequently, the vibration analysis could be done in the time domain, the frequency domain, or the time–frequency domain [9, 14, 15].

Recently, machine learning (ML) was also adopted in both the diagnosis and prognosis of bearings' faults. ML could be based on supervised or unsupervised learning algorithms. Supervised learning (SL) is the process in which the machine will learn with the help of given data containing features and labeled data. Features are the independent parameters, while dependent parameters are called labels. In contrast, unsupervised learning (UL) is about labeling unlabeled data using algorithms and then running processes to produce analyzed data. ML models usually use a considerable percentage of data to train the system and then use the rest to test the prediction accuracy before conducting the prediction on foreign data [16]. Multiple ML techniques have been employed in the last two decades to detect defects in rolling element bearings, using distinct test data parameters to train the algorithms.

Artificial neural networks (ANN) are usually adopted when dealing with complicated problems with many trainable parameters. Several types of ANN were adopted in bearings troubleshooting and condition-based maintenance (CBM), such as convolution neural networks (CNN) and recurrent neural networks (RNN). Eren, L. [17] presented a one-dimensional CNN model to monitor bearings health using a single-learning body model and achieved 97% fault detection accuracy. Hoang and Kang [18] used a novel CNN model that transforms 1D signals into 2D ones and approached 100% accuracy in defect detection using the Case Western Reserve University (CWRU) public bearing data set. RNN is another branch of ANN that recurrently processes the data instead of the feed-forward behavior, allowing outputs to be processed as inputs while having unknown status in the hidden layers [19, 20]. However, one of the disadvantages of RNN is the possible gradient's exploding and vanishing problem in time series during the backpropagation process [21]. Therefore, long-short-term memory (LSTM) could be used. LSTM is a time-recurrent neural network block used to solve the vanishing gradient problem [22] since it is suitable for processing and predicting data with gaps and delays in a time series based on complex historical fault data. Liu et al. [23] proposed their model of Gated Recurrent Unit-based denoising autoencoder, which outweighed other classifiers with more than 99.5% diagnosis accuracy. Several works have adopted the K-Nearest Neighbors' Classifier [24–26] and used it in the fault detection of bearings as a different type of ANN classifier. Moreover, Multi-Layer Perceptron (MLP), the classical supplement of feed-forward neural networks, is regularly used

in CBM of bearing [27–29]. Many other ML approaches were implemented in the CBM of bearings, such as Neural Fuzzy Networks [30], Generative Adversarial Networks [31], and Naive Bayes Classifier algorithm [32]. Regression techniques were also adopted in different bearing CBM models [33–35].

On the other hand, Trees Ensembles (TE) methods such as Decision Trees (DT), Extra Tree Classifier (ETC), Random Forest (RF), and Gradient Boost (GBoost) are increasingly adopted by researchers in classification and regression problems of bearing fault-detection [36–38] due to their low computational demands. In addition, enhancing the TE in classifying the defect is conducted with the use of other classification algorithms such as Support Vector Machine (SVM) [39], fuzzy classifiers [40], envelope signal-based feature extraction [41], and 2D-discrete wavelet transform [42, 43]. For example, Patil and Phalle [44] have found that adopting ETC to classify bearing faults achieved an accuracy of 98.12%. Moreover, Nistane and Harsha [45] used ETC supported with a stationary wavelet transform algorithm to compare this technique's performance with RF and MLP regression. Furthermore, RF is another algorithm that combines multiple decision trees into one single decision tree, making it robust for regression and classification tasks; however, the training time increases in this case. The main difference between RF and ETC is that RF chooses the best node to split, while ETC randomly splits nodes to sample the training data, reducing bias, variance, and training time. Some researchers proposed ML models using RF [40, 46]. In contrast, others [47] employed a refined composite multi-scale reverse dispersion entropy technique with RFC, achieving 100% maximum prediction accuracy and a 97.3% average classification accuracy.

In addition to ETC and RF, Boosting is a general ensemble method that creates a robust classifier from several weak classifiers. One of the boosting techniques is GradientBoosting (GBoost), which minimizes the prediction error of the next model by choosing the best outcome of the last model based on the DT model. It gained much attention in ML due to its efficiency in making predictions and its ability to handle large and sparse data. However, Extreme Gradient-Boosting (XGBoost) has a computational advantage over GBoost, where training progresses slowly. XGBoost is a SL algorithm used to solve classification and regression problems [48] and is commonly used in bearing fault diagnosis [46, 49, 50]. Qi et al. [51] included different classifiers in their models and achieved 90.42%, 95.76%, and 97.21% prediction accuracies when using DT, XGBoost, and Weighted Extreme Gradient Boosting (WXGB), respectively. Xia et al. [52] later adopted XGB in their Federated Learning model using the Privacy-Preserving technique. Some works have included combining Adaboost and EMD [53, 54], while in [55], researchers used the DT classifier followed by

Adaboost to compare it with SVM to achieve 96% and 92% maximum testing accuracy, respectively. One more classifier is the Light Gradient-Boosting Machine (LightGBM), which was implemented in bearing fault detection in [56–58] and for the same contribution but supported with CNN in [59–62].

In predictive maintenance and condition monitoring, machine learning techniques have gained significant traction for their ability to extract valuable insights from complex industrial data. Recent advancements in tree-based ensemble methods, such as Decision Trees, Random Forests, and XGBoost, have demonstrated remarkable performance in diagnosing faults in rolling element bearings using vibration signals [63]. Complementing these approaches, deep learning architectures have emerged as powerful tools for time series forecasting, potentially enabling proactive maintenance strategies [64]. However, industrial data's growing complexity and scale pose challenges regarding computational efficiency and privacy concerns. In this regard, distributed frameworks for training XGBoost models have been proposed, leveraging parallel computing to handle large-scale datasets [65]. Additionally, federated learning techniques have been explored for collaborative bearing fault diagnosis, enabling privacy-preserving data sharing across multiple parties [66]. Recognizing the diversity of machine learning approaches, comprehensive reviews have been conducted to evaluate traditional and deep learning methods for fault diagnosis in rotating machinery [67]. These studies provide valuable insights into the strengths and limitations of various techniques, guiding practitioners in selecting appropriate methods for their specific application scenarios. Beyond the realm of rotating machinery, machine learning has also found applications in monitoring and diagnosing faults in actuators, which are critical components of many industrial systems [68]. Integrating intelligent algorithms into actuator systems can enhance their reliability and performance, further underscoring the pervasive impact of machine learning in industrial operations. While significant progress has been made, implementing machine learning algorithms in real-world industrial environments remains challenging [69]. Data quality, sensor reliability, and system complexity must be carefully considered to ensure accurate and robust condition monitoring solutions.

The models developed in the literature using the ensembles method and ANN algorithms were assessed based only on overall accuracy. However, those models usually make classification depending on the part of the independent feature data set while ignoring features with lower contribution scores to the classification model. However, a generalized multi-parameter bearing fault detection classification model requires mapping between every signal feature with the target defect. Therefore, this paper proposes and compares three tree ensemble machine learning

models (Decision Tree, Random Forest, and XGBoost) for diagnosing and prognosing faults in roller element bearings using vibration data. It utilizes 17 time-domain statistical features extracted from the vibration signals to the machine learning models as input features. A thorough feature importance analysis is conducted to understand which vibration signal features contribute most significantly to the performance of each machine learning model in detecting bearing faults and their locations.

Experimental Testing, Results, and Data Setup

This study proposes and compares three tree ensembles of ML models that can diagnose and prognose faults of roller element bearings. Each of the three models can detect the defect's existence and location, whether it occurred on the outer or inner rings. Moreover, four more extra bearings data were experimentally collected on the same testing machine used in QU-DMBF [62], which were used to train the system and test it afterward. The introduced models enhance the CBM decisions of rotating equipment and assess the importance of time-domain signal parameters used to define the defective bearing. In this paper, we examine each model and the features it deems essential to make a decision, referred to as feature importance. This aspect is often overlooked in prior studies. Many researchers have gravitated toward benchmarking and frequently fixating on accuracy scores, which usually leads to ignoring the exploration of the importance of features. Regarding predictive modeling, prior papers may have focused primarily on accuracy scores and model performance rather than usually reporting on individual features and their contribution to the model performance cross-referenced with the problem domain. While it acknowledges the importance of accuracy as a vital component, this study explores the importance of features in model performance and domain: experimental Testing, Results, and Data Setup.

This paper utilizes benchmark experimental data to measure vibrations and validate defect diagnosis and prognosis functionalities using multiple time domain parameters. This section introduces the experimental test rigs and the set of bearings employed to gather vibrational data, along with incorporating these bearings into the existing QU-DMBF dataset. Subsequently, the time domain parameters essential for bearing fault analysis will be elaborated upon, detailing all the time domain parameters employed. The section concludes by discussing the data setup and model training criteria.

Qatar University Dual-Machine Bearing Fault Benchmark (QU-DMBF) Dataset

Figure 1 shows the test rig used in this research. This rig was originally a Machinery Fault Simulator from SpectraQuest Inc., USA. The rig includes a DC motor (0.5 HP, 90 VDC, 5 A) with a maximum rotational speed of 2500 RPM. The original shaft of the machine was removed and replaced by a new and bigger one to accommodate the tested bearings (type NSK-6208). For the sake of resistance and stability, other supporting mechanical components were also re-designed. The machine has an approximate weight of 5kg and overall dimensions of (100 × 63 × 53) cm.

Initially, 19 different bearing configurations were considered in the investigation: one healthy, nine with a defect on the outer ring, and nine with a defect on the inner ring. The defect sizes vary between 0.35 mm and 2.35 mm. However, three more healthy bearings were added to the QU-DMBF published data set to balance healthy to faulty samples. One extra bearing with a 0.5 mm defect located on the outer race was exclusively used for testing. Hence, 23 bearings in total (same size and same brand) were employed in generating the dataset.

Six ICP® accelerometers (PCB Piezotronics, Model No. 352C33) were used to extract vibrational signals. These accelerometers were fixed on the motor and in different locations on the machine's mounting base, and they were also fixed in various orientations and distances from the bearing rotation point at the end of the shaft (Fig. 2).

Two four-channel NI-9234 sound and vibration input modules controlled the readings at a sampling frequency of 4.096 kHz. Signal recording for each bearing was taken for 30 s at five different speeds, which are 240, 360, 480, 720, and 1020 RPM. Hence, the total recording time of faulty bearings was 16,200 s (18 bearings × 6 accelerometers × 5 speeds × 30 s). The recorded time for each healthy bearing was increased to 270 seconds to ensure proper balance in the training data. The procedure consisted of recording 32,400 s

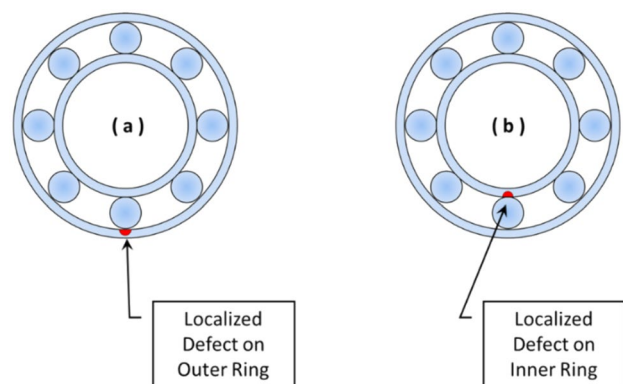
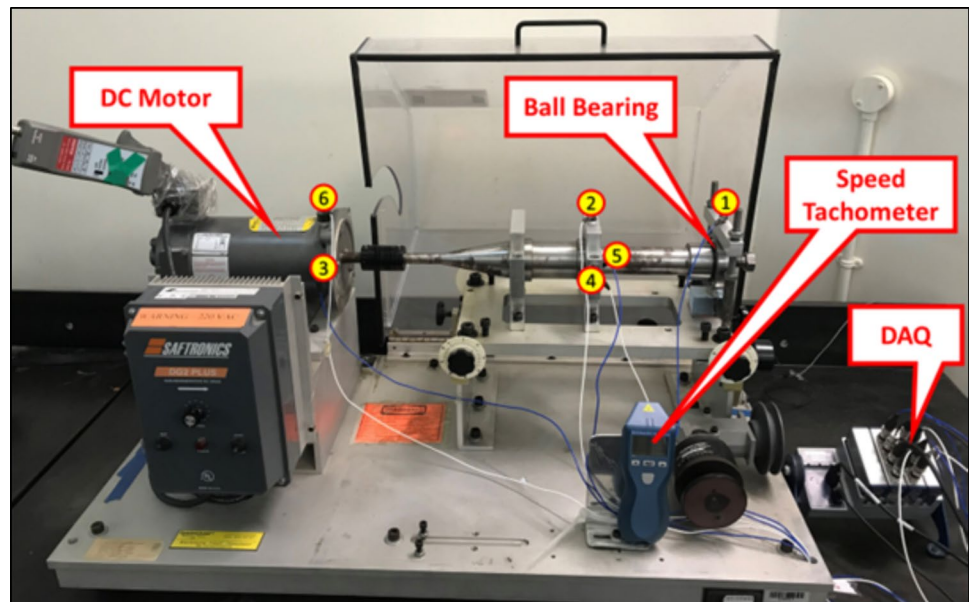


Fig. 1 Defect insertions: **a** on the outer ring; **b** on the inner ring

Fig. 2 Experimental testing machine where numbers 1–6 refer to accelerometers' locations



(4 bearings \times 6 accelerometers \times 5 speeds \times 270 s), similarly for all the healthy bearings.

Time Domain Analysis

Vibration responses due to defective bearings could be studied using different approaches, such as the time domain, frequency domain, or time–frequency domain [9, 15]. Several statistical parameters could be extracted from the time domain signals. These indicators could be training parameters in fault detection models for bearings CBM. Different mathematical operators can impact the statistical characteristics of the signals, and a change in the mathematical operator can either increase or decrease the informational value of the signals. For example, the Root Mean Square (RMS), also known as the quadratic mean or the square root of the arithmetic mean of the squares of the values, is related to the vibration energy of a signal in the time domain [71]. Furthermore, the Crest Factor (CRSF) is the dimensionless waveform metric that displays the ratio of the peak amplitude divided by the RMS value of the signal. It usually indicates how extreme the peaks are in a waveform. Signal spikiness affects how sensitive the crest factor is. Its sensitivity to signal spikiness can provide an early indication of significant changes in vibration readings. The CRSF increases when the signal has distinct spikes or peaks. A waveform containing random or periodic spikes scattered throughout the signal would have a greater CRSF than a pure sinusoidal wave. Researchers observed that the CRSF is more sensitive than skewness to the effects of radial load on bearing vibration.

One more parameter is the Kurtosis (KU), which represents a measure of the "tailedness" of the probability distribution of the random variable. In probability theory

and statistics, KU is defined as the fourth-order normalized moment concerning the square of the variance of a time series signal. It is more sensitive to impacts and degradation than CRSF and is the most sensitive parameter that can be used to identify faults in rolling element bearings [72, 73]. A KU of 3 usually represents the normal behavior of a healthy machine. At the same time, a kurtosis higher than 3 indicates advanced states of fault progression (a distribution with heavier tails than a normal one) [74]. KU is a valuable instrument for keeping track of the health of rotating machinery, including gears, bearings, and other components.

Time domain indicators have proved their effectiveness in training ML models to detect bearing failures [75]. Different time domain indicators were adopted during the last two decades to train ML models, such as intrinsic mode functions (IMFs) [76] and Zero Crossing features (ZC) [77]. Many of these time domain parameters were obtained from the data and implemented in the proposed model. Of all time domain indicators in literature [78–81], seventeen (17) were used in training and testing the current work's model. Their formulas and abbreviations are displayed in Table 1.

Experimental Data Setup

Bearing fault detection is considered a classification problem. The objective is to determine the bearing's health and the location of the defect if the bearing is defective. Hence, each class was given a label, as shown in Table 2. To develop the ML model, 80% of the dataset was used for training, while the remaining 20% was used for testing. As explained in "Qatar University Dual-Machine Bearing Fault

Table 1 Time domain indicators used to train and test the proposed model

Equation name	Equation formula
Peak	$PEAK = Max(x)$ (1)
Peak to peak (p2p)	$p2p = Max(x) - Min(x)$ (2)
Mean square value (MSV)	$MSV = \frac{1}{N} \sum_{i=1}^N x_i^2$ (3)
Root square mean (RMS)	$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$ (4)
Crest factor (CRSF)	$CRSF = \frac{PEAK}{RMS}$ (5)
Standard deviation (STD)	$STD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$ (6)
Variance (VAR)	$VAR = \frac{1}{N-1} \sum_{i=1}^N x_i - \bar{x} ^2$ (7)
MEAN (\bar{x})	$MEAN = \bar{x} = \frac{1}{N} \sum_{i=1}^N x(i)$ (8)
Root sum of squares (RSSQ)	$RSSQ = \sqrt{\sum_{i=1}^N x_i ^2}$ (9)
Entropy	$Entropy = - \sum_{i=1}^N y(x_i) \log_2 y(x_i)$ (10)
Square root amplitude value (SRAV)	$SRAV = \left(\frac{1}{N} \sum_{i=1}^N \sqrt{ y(x_i) } \right)^2$ (11)
Kurtosis (KURT)	$KURT = \frac{N \sum_{i=1}^N (x_i - \bar{x})^4}{\left(\sum_{i=1}^N (x_i - \bar{x})^2 \right)^2}$ (12)
Shape factor (SF)	$SF = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i)^2}}{\frac{1}{N} \sum_{i=1}^N x_i }$ (13)
Clearance factor (CLF)	$CLF = \frac{\max(x_i)}{\left(\frac{1}{N} \sum_{i=1}^N \sqrt{ x_i } \right)^2}$ (14)
Impulse indicator/factor (IMPF)	$IMPF = \frac{\max(x_i)}{\frac{1}{N} \sum_{i=1}^N x_i }$ (15)
Fourth order normalized power (NP4)	$NP4 = \frac{1}{N} \sum_{i=1}^N \left(\frac{y(x_i - \bar{y})}{\sigma} \right)^4 - 3$ (16)
Skewness (SKEW)	$SKEW = \frac{N \cdot \sum_{i=1}^N (x_i - \bar{x})^3}{\left[\sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \right]^3}$ (17)

Table 2 Defect classes and corresponding labels

Class	Label
No-defect	0
Inner-defect	1
Outer-defect	2

Machine Learning Models

In this section, we benchmark the dataset obtained from the experiments using three ML models: Decision Tree (DT), Random Forest (RF), and extreme GradientBoosting (XGBoost). These three models use the fundamentals of the DT algorithm. Thus, a feature importance analysis will be conducted, which provides insights into how influential a particular input is in the predictions of each model.

Benchmark (QU-DMBF) Dataset” section, the dataset was stratified to ensure a balanced presence of examples for each class label.

Methodology

Before presenting the modeling, several assumptions need to be considered. Firstly, the data is assumed to follow a particular distribution, which can impact the models' performance. Additionally, the models assume that the features used for training are independent of each other, which might only sometimes hold in real-world scenarios. The assumption of balanced classes is also crucial for the models to learn effectively from the data. Furthermore, the models assume that the features selected are relevant and contribute significantly to the prediction task.

Moreover, it is worth remembering that the dataset consists of vibration signals obtained from accelerometer readings on bearings with different defect conditions (healthy, inner race defect, outer race defect) collected at Qatar University, with details on the experimental setup and data collection procedure provided. The study used time-domain statistical features extracted from the vibration signals as input features for the machine learning models, employing 17 different time-domain features. The models were treated as classification problems to predict the bearing condition (healthy, inner race defect, outer race defect) based on the input features. The dataset was split into training (80%) and testing (20%) sets in a stratified manner to ensure class balance.

The metrics used to validate and evaluate the models performance are precision, recall and F1-score. The precision score reflects how well the model has predicted the true positives in fraction to the overall predicted positives. On the other hand, the recall score shows how fit is the model to predict all true positives in fraction to the sum of true positives and false negatives. Finally, the F1-score gives an assessment on both, the precision and recall, producing a single measure while taking into account both the false positives and false negatives.

Figure 3 illustrates this paper's proposed approach for implementing machine learning models. It comprises three essential parts. Firstly, categorical variables were encoded into a numerical format during data preparation. Then, the Min–Max data normalization technique is introduced to the data to ensure all features have the same scale. Finally, a stratified split was implemented to provide a consistent target class distribution in the training and testing sets. This strategic division ensures against data imbalance and potential model bias stemming from the underrepresentation or the overrepresentation of specific target classes. The second part involves model implementation, where the grid search method is applied to fine-tune the model's hyperparameters through an iterative process of building and testing. The process starts by setting a grid and values range to the model's hyperparameters; each combination of the model's grid will be evaluated. Once the validation set has achieved optimal

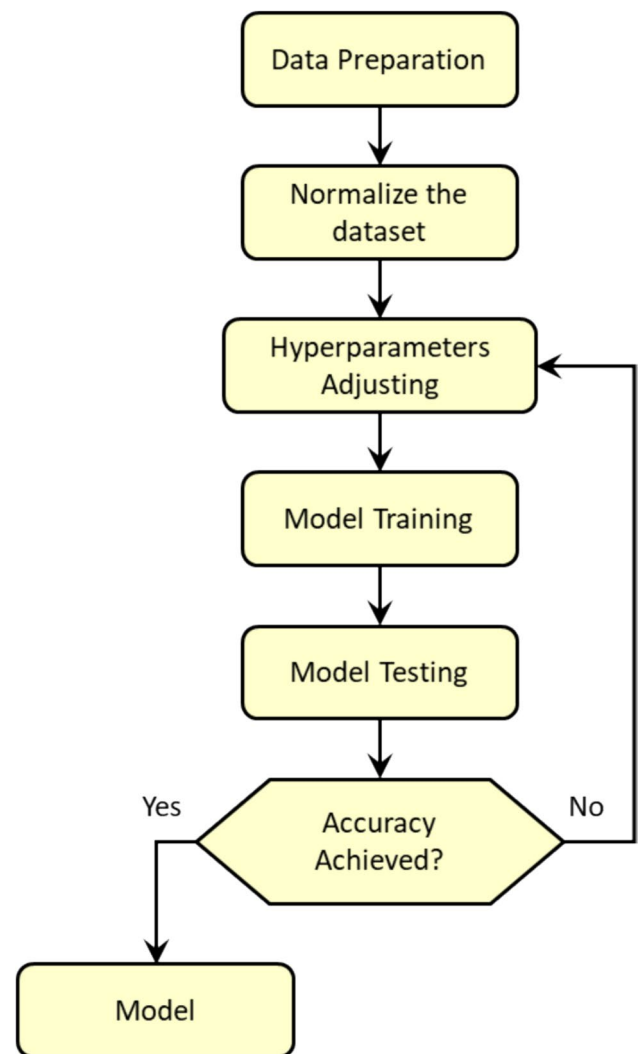


Fig. 3 Methodology flow chart

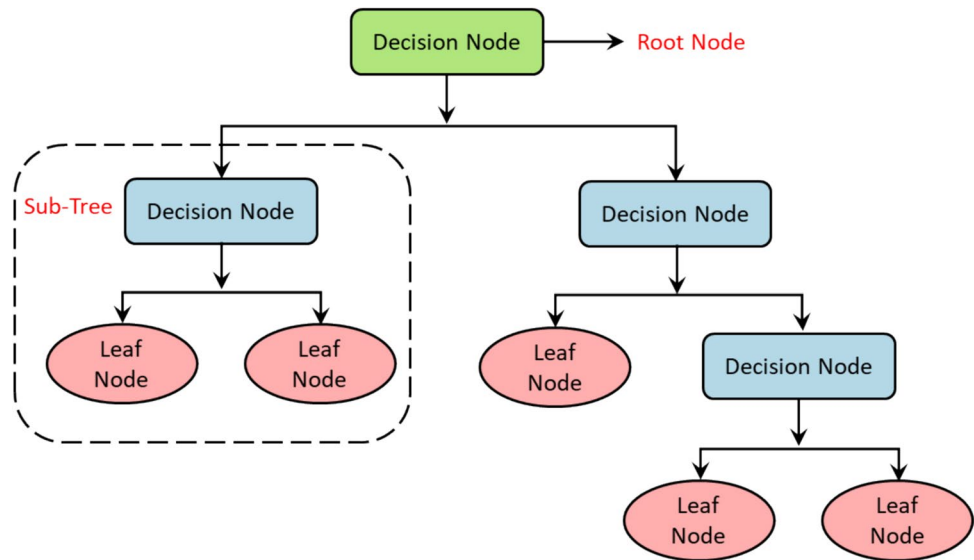
accuracy, the grid search reports the hyperparameter values. Lastly, the concluding phase involves implementing the model on the entire dataset and reporting the accuracy report.

Ensemble Methods

Decision Tree (DT)

A decision tree algorithm is a versatile supervised machine-learning algorithm for classification and regression problems. It creates a flowchart-like tree structure (Fig. 4), where each internal node denotes a feature, branches denote rules, and leaf nodes denote the algorithm's result. The decision tree algorithm works by recursively making new decision trees to maximize the homogeneity of the target variable in each subset of the dataset until a stage is reached where

Fig. 4 Decision tree algorithm



further classification is not possible, and the final node is called a leaf node.

The primary challenge in decision tree implementation is identifying which attributes to consider as the root node at each level, known as attribute selection. The decision tree learning process employs a divide-and-conquer strategy by conducting a greedy search to identify the optimal split points within a tree. In the context of classification problems, to measure the degree of impurity in a subset of the tree, the Gini impurity $Gini(D)$ is used, as shown in Eq. (18). For the information gain, $IG(D)$, which measures the reduction in disorder [36] yield by splitting the data based on a feature (F), Eq. (19) is used, where entropy or disorder $E(D)$ is shown in Eqs. (20) [36].

$$Gini(D) = 1 - \sum_{i=1}^N p_i^2 \quad (18)$$

$$IG(D) = E(D) - \sum_{v \in Values(F)} \frac{|D_v|}{|D|} E(D_v) \quad (19)$$

$$E(D) = - \sum_{i=1}^N p_i \log_2(p_i) \quad (20)$$

where p_i is the probability of class i occurring in the node, and N is the total number of classes, and D_v is a subset of the collected samples set D [36].

However, DTs are prone to overfitting and underfitting problems. Hence, to obtain a model with high classification performance, the hyperparameter of the network must be optimized. Thus, the grid-search optimization technique was introduced to the model, resulting in the following values for the hyperparameters: entropy [70], to measure the quality of

the split, the ideal depth of the tree is found to be 9, with a minimum three samples to create a leaf, and a minimum of 7 samples to make a split. The model achieved an accuracy of 82% on the testing dataset and a 94% accuracy on the training dataset. Table 3 shows the model's precision, recall, and F1 score, and Fig. 5 shows the confusion matrix.

Random Forest (RF)

Random forest (RF) is an ensemble supervised learning method for classification, regression, and other tasks that operates by constructing a forest of decision trees and combining their outputs (Fig. 6).

Breiman and Cutler first introduced it [82] and trademarked it. The algorithm grows a forest of trees, where each tree is trained on a randomly chosen subspace of the training data and introduces variation among the trees. The decision at each node is selected by a randomized procedure rather than a deterministic optimization. The RF algorithm is an extension of the bagging method and utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Feature randomness generates a random subset of features, which ensures low correlation among decision trees. The algorithm has three primary hyperparameters that must be set before training: node size, the number

Table 3 Performance metrics of the DT model on the testing data

Label	Precision	Recall	F1-score
No-defect	0.96	0.96	0.96
Inner-defect	0.74	0.85	0.79
Outer-defect	0.85	0.72	0.78
Average	0.85	0.84	0.84

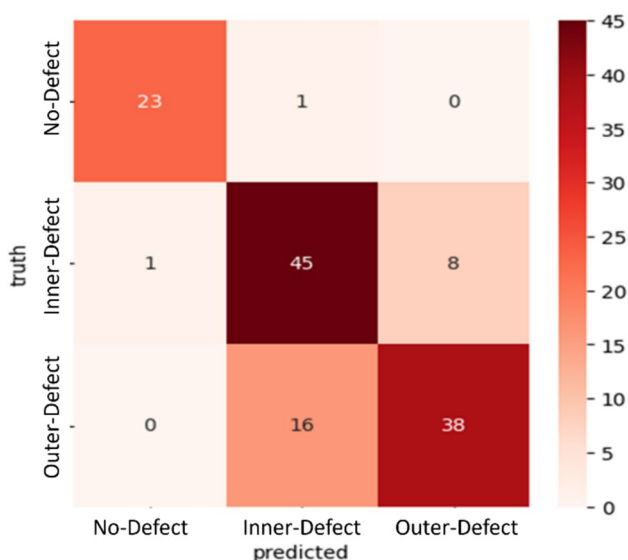
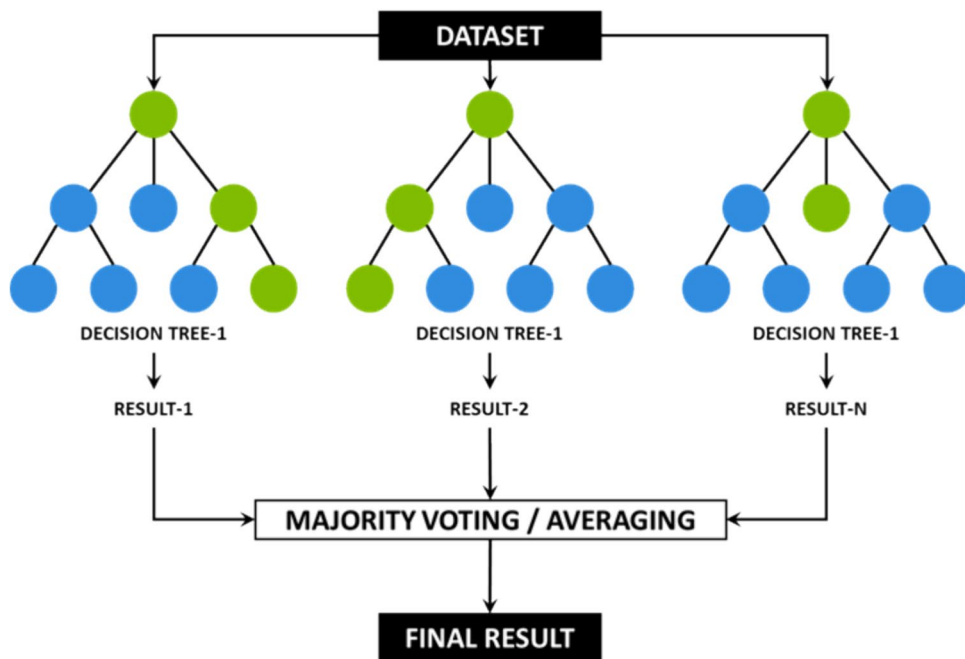


Fig. 5 Confusion matrix of the DT model on the testing data

of trees, and the number of features sampled. In the case of Random Forest for classification, each decision tree in the ensemble is often built using Gini impurity $Gini(D)$, like the earlier decision tree. In summary, RF is a popular machine learning algorithm for classification and regression problems in various societal and industrial sectors. It builds multiple decision trees via randomness injection and takes a vote among the trees to make robust and accurate predictions. Among the key advantages of this method are its ease of use, flexibility, and the ability to handle missing

Fig. 6 Random forest algorithm



values. Additionally, random noise in the data doesn't affect the accuracy of RF models. While there isn't a single equation that encapsulates random forests, the algorithm involves creating multiple decision trees, each trained on a different bootstrap sample of the data, and making predictions by aggregating the results of these trees.

A grid search was applied to tune the RF model's hyperparameters and optimize the dataset's results. After several grid searches, it was found that the optimal number of trees in the forest is 60 trees, the max depth of the tree is 10, and the minimum number of samples mandated to split a node is 2. The classification report shows that the model achieved an accuracy of 91% on the testing data, as displayed in Table 4. Moreover, the confusion matrix, displayed in Fig. 7, shows a recall score of 100% for the No-Defect class.

XGBoost

XGBoost, short-term for Extreme Gradient Boosting, is a software library that implements optimized distributed

Table 4 Performance metrics of the RF model on the testing data

Label	Precision	Recall	F1-score
No-defect	0.9	1.00	0.96
Inner-defect	0.89	0.91	0.90
Outer-defect	0.92	0.87	0.90
Average	0.91	0.93	0.92

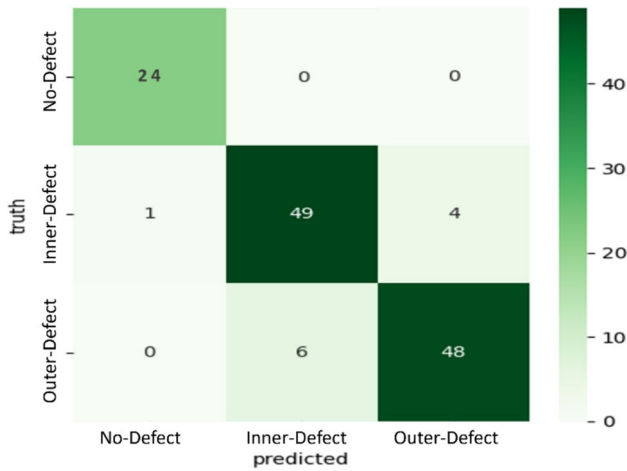


Fig. 7 Confusion matrix of the RF model on the testing

gradient boosting machine learning algorithms under the Gradient Boosting framework. Like other boosting methods, it builds models sequentially, where each model learns and improves upon the previous model. However, XGBoost utilizes more accurate approximations when computing splits. This prevents overfitting and enhances speed. XGBoost uses gradient boosting, which minimizes loss(L) when adding new models. This is done using gradients in the loss function $L(y_i, y'_i)$ to estimate the descent direction. New models predict the residuals or errors of prior models and gradually boost the predictions. For multiclass classification problems, the loss function is shown in Eq. (21) [50]. Where N is the number of samples and M is the number of classes, the $y_{i,j}$ represent the true label for the i th example and the j th class, and $p_{i,j}$ represent the predicted probability of the

sample in question to the class. XGBoost models are made up of decision trees like random forests. However, while random forest trains trees in parallel, gradient boosting trains trees sequentially. XGBoost is particularly popular due to its ability to handle large datasets, achieve state-of-the-art performance in many machine learning tasks such as classification and regression, and efficiently handle missing values without requiring significant pre-processing. It also has built-in support for parallel processing, making it possible to train models on large datasets quickly. Since its introduction, XGBoost has become the machine learning algorithm of choice for data scientists and machine learning engineers. It is known for its speed, ease of use, and performance on large datasets. It does not require optimization of parameters or tuning, allowing it to be used immediately after installation without further configuration (Fig. 8). Additionally, XGBoost counters the overfitting and underfitting problems in RF and DT models by incorporating regularization techniques such as feature subsampling and learning rate. The regularization term $\Omega(f)$ is defined as shown in Eq. (22), where.

$$L(y_i, y'_i) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j}) \tag{22}$$

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T \omega_j^2 \tag{23}$$

The regularization term to control the model complexity [49], shown in Eq. (22), consists of two parts: γT , and $\frac{1}{2} \lambda \sum_{j=1}^T \omega_j^2$. In the first part, γ refers to the regularization parameter for tree complexity, while T is the number of terminal nodes (leaves) in the tree. Combined, they aim to

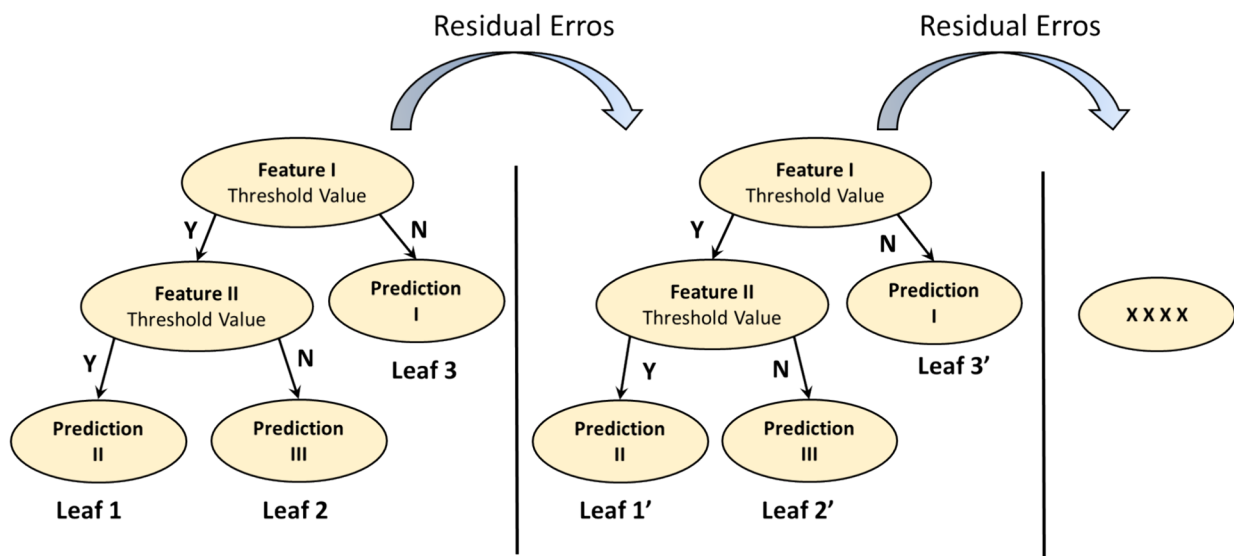


Fig. 8 XGBoost Algorithm

penalize the complexity of the tree based on the number of terminal nodes. The second term represents the L2 regularization penalty applied to the weights of the terminal nodes. Where λ is the L2 regularization parameter that controls the strength of the penalty.

However, to capture complex relationships and an appropriately generalized model, optimizing the XGBoost is fruitful as it utilizes the model to its best performance. Hence, using the grid-search method for hyperparameter tuning with a fivefold cross-validation, which is determined as the optimal choice, balancing computational cost and performance accuracy, helps mitigate overfitting. As a result, the model's optimal hyperparameters are 0.1 for eta, which is an alias for

learning rate, 3 as the ideal depth of the tree, and 250 for the number of runs to learn. Figure 9 shows the log loss function with the number of epochs. As observed, the model's performance stops improving around the 180th iteration, corresponding to Fig. 10, the classification error for the test set plateaus around the 180th iteration. Hence, a learning rate of 0.1 is suitable for the dataset. It is worth noting that early stoppage has been set to 15 iterations. Consequently, if after 15 iterations, the accuracy does not improve, the training shall stop.

As expected beforehand and illustrated in Table 5, the XGBoost outperformed all the other models with an accuracy score of 92% on the testing dataset.

Fig. 9 XGBoost log loss function versus the number of epochs

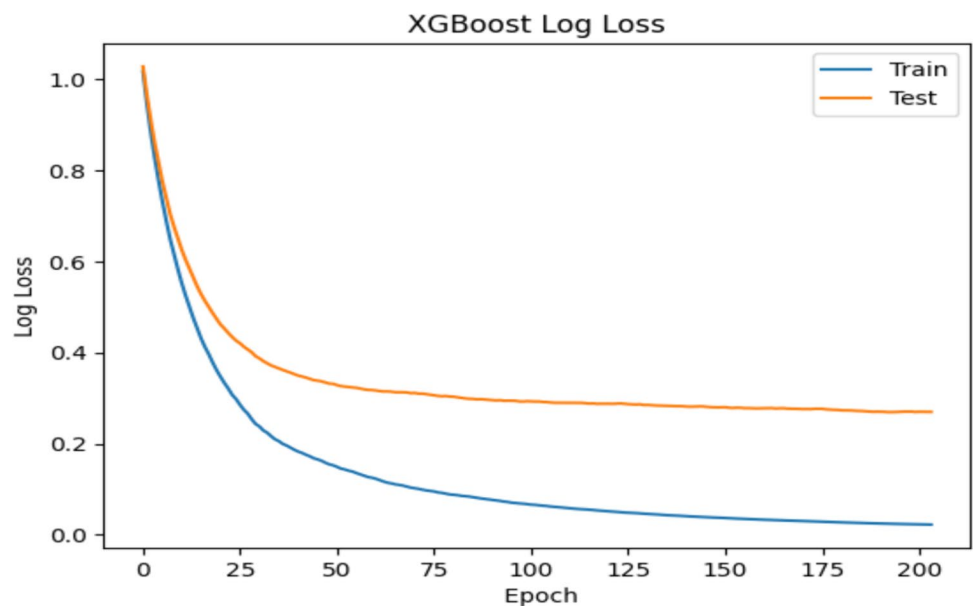


Fig. 10 XGBoost classification error figure shows the classification error value on the y-axis and the number of epochs on the x-axis

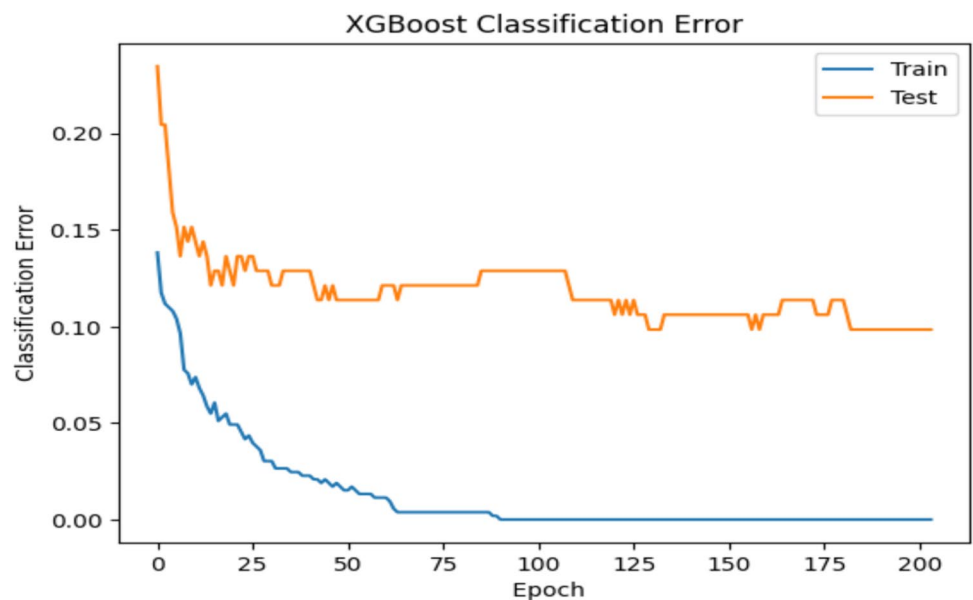
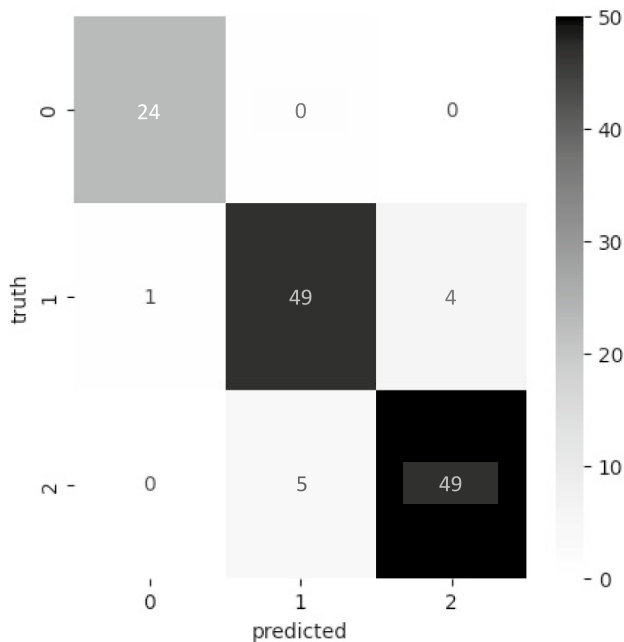


Table 5 Performance metrics of the XGBoost model on the testing data

Label	Precision	Recall	F1-score
No-defect	0.96	1.00	0.98
Inner-defect	0.91	0.91	0.91
Outer-defect	0.92	0.91	0.92
Average	0.93	0.94	0.93

**Fig. 11** Confusion matrix of the testing data on the XGBoost model

The confusion matrix, shown in Fig. 11, indicates the performance of the XGBoost; like the Random Forest model, it predicts the No Defect class at 100% accuracy.

Feature Importance

The Ensemble trees models, Random Forest and XGBoost, build multiple decision trees to drive the output. The importance of the dataset's features in these models is derived from the collective behavior of these trees. In random forests, the Gini importance is typically used to calculate how often a feature is used to split the dataset across all the trees created by the model. Features that are used more often result in higher feature importance scores. However, XGBoost models utilize the Gain importance method, where features are ranked based on the resulting gain in accuracy introduced by the feature. The contribution of each feature is aggregated over all the trees created by the model; hence, the gain is calculated at each split. Finally, the Decision Tree model utilizes the Gini importance as well; however, since

DT builds only a single tree, the importance of each feature comes from its contribution to the reduction in impurity or entropy over the single tree.

On the other hand, the precision, recall, and F1 score values only report the models' performance without providing insights into feature selection or the driving factors contributing to that performance. In other words, for the data under study, each ensemble method selects its essential feature list in mapping between them and the targeted output. Therefore, a high-accuracy model is not the only target to achieve. In addition, the extent to which the ensemble algorithm considers the entire training feature in detecting the defect on the bearing is also a necessary fact to consider. After training the model, a feature-important analysis is conducted to illustrate each feature's contribution to the targeted prediction, as shown in Fig. 12.

In Fig. 12a, the DT approach has identified the SF as the main feature that dominates the bearing defect behavior at around 35% of the mapping process. Other input parameters, such as CRSF, CLF, or RSSQ, are essential but have scores of less importance. Furthermore, the algorithm neglects the accelerometer location by allocating a zero-importance score. Accelerometer location is a crucial feature that contributes to the signal time-histogram intensity.

Therefore, the model might suffer from collinearity, where other features are highly correlated, and the model is utilizing them to make up for the accelerometer location feature. Meanwhile, RF allocates significant importance to each input feature without neglecting them. Nevertheless, two primary experimental conditions (accelerometer location and motor speed) were unimportant. Eventually, the XGboost boosting technique showed more consideration for the accelerometer location and the motor speed. Although four distinctive features were neglected, their values were severely dependent on other features, such as RMS, STD, etc., already included in the model's essential features.

Based on the previous discussion, one can state that the proposed method utilizes tree ensemble machine learning models such as Decision Trees, Random Forests, and XGBoost for condition monitoring and predictive maintenance of rotating machinery, precisely diagnosing faults in rolling element bearings. These models offer various applications, including early detection and diagnosis of bearing faults by identifying defects and pinpointing their location within the bearing using vibration signal data. Additionally, they support condition-based maintenance (CBM) of bearings by monitoring vibration features and enabling proactive maintenance scheduling to prevent catastrophic failures. The models also facilitate the fault prognosis of bearings, allowing for the tracking of defect progression over time. Furthermore, they can be integrated into automated condition monitoring systems to continuously monitor bearing health in industrial machinery. The study provides a comparative

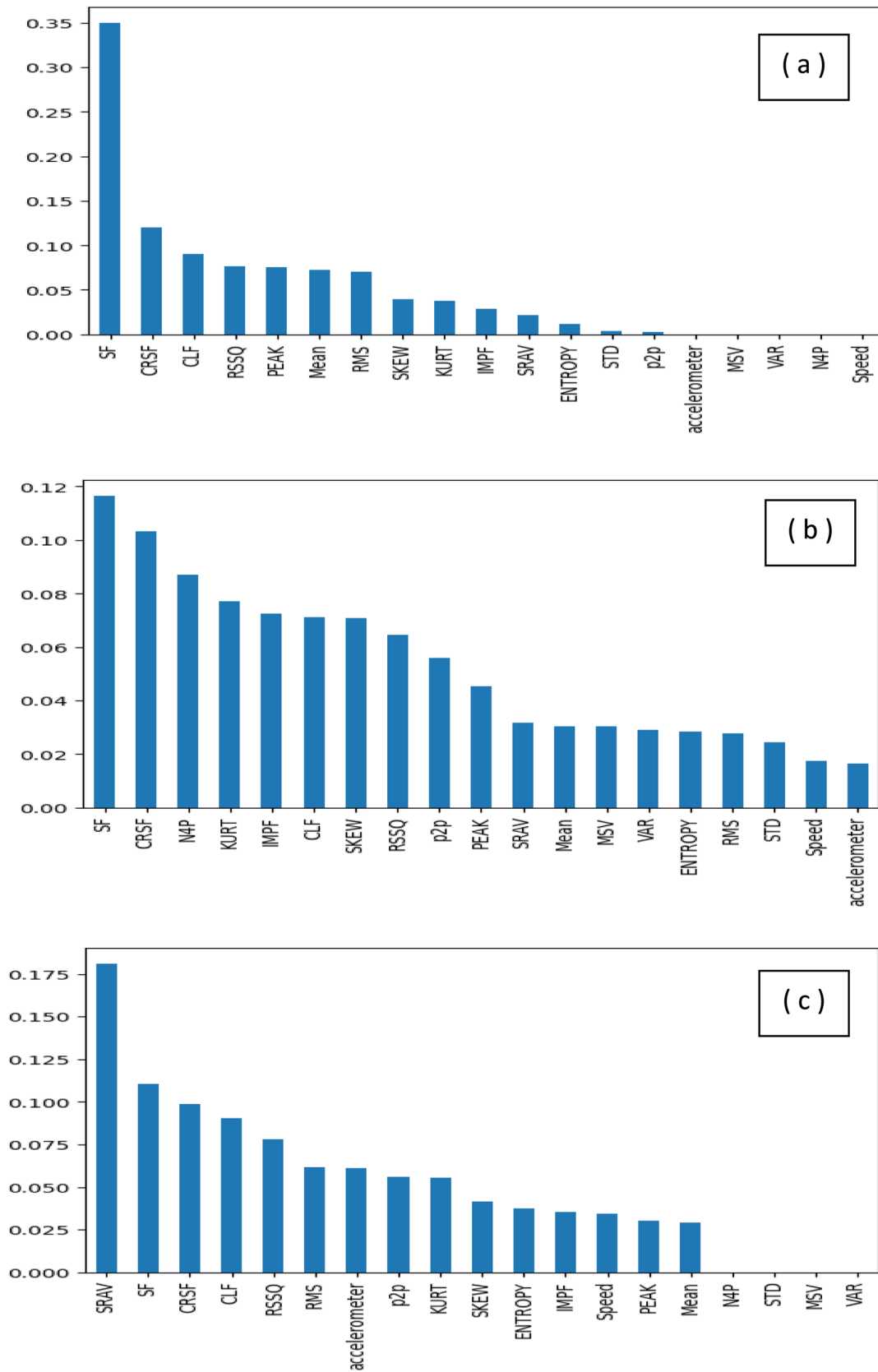


Fig. 12 Relative importance analysis results for the three models, **a** decision tree, **B** random forest, and **C** XGBoost

benchmark of different tree-based machine-learning techniques for bearing fault diagnosis, emphasizing the primary application of leveraging machine learning on vibration data for reliable and early fault detection and diagnosis in rolling element bearings of rotating equipment across various industrial sectors. This approach enables timely maintenance and helps prevent unexpected failures that could result in costly downtime.

Moreover, the research reveals that the Decision Tree method identifies specific features that predominantly influence defect behavior, potentially reflecting the localized impact of faults within the bearing structure. In contrast, Random Forest assigns considerable importance to all input features, suggesting a holistic approach to fault detection that considers the collective influence of various parameters. XGBoost, with its balanced consideration of essential features such as accelerometer location and motor speed, showcases a nuanced understanding of the interplay between different variables in determining bearing health and fault location.

The research delves into the importance of various time domain parameters like Root Mean Square (RMS), Crest Factor (CRSF), and Kurtosis (KU) in analyzing bearing faults. These parameters offer a tangible understanding of the vibration energy, signal spikiness, and distribution characteristics, which are crucial for fault detection in rotating machinery. The models, including Decision Tree, Random Forest, and XGBoost, showcase high accuracy rates and highlight the significance of specific features in predicting bearing defects. Furthermore, the feature importance analysis reveals how different models prioritize input features. For instance, Decision Trees emphasize features like SF, while Random Forests assign significant importance to all input features without neglecting any. On the other hand, XGBoost demonstrates a balanced consideration for features like accelerometer location and motor speed, showcasing its ability to handle complex relationships and optimize model performance effectively.

Conclusion

This research paper provides a detailed analysis of the experimental testing, results, and data setup related to bearing fault diagnosis and prognosis functionalities. The study utilizes the Qatar University Dual-Machine Bearing Fault Benchmark (QU-DMBF) Dataset, which includes various bearing configurations with defects on the outer and inner rings. Vibrational signals were extracted using accelerometers, and time domain analysis was conducted to study defective bearings. Three machine learning models were benchmarked using this dataset: Decision Tree (DT), Random Forest (RF), and Extreme Gradient Boosting (XGBoost). These models

were evaluated based on their performance metrics, such as precision, recall, and F1-score, with XGBoost outperforming the other models with an accuracy score of 92% on the testing dataset. The Decision Tree model achieved an accuracy of 82% on the testing dataset, with precision, recall, and F1-score values reported for each defect class. On the other hand, Random Forest achieved an accuracy of 91% on the testing data, with a recall score of 100% for the No-Defect class. XGBoost, known for its optimized distributed gradient boosting algorithms, demonstrated superior performance with an accuracy score of 92% on the testing dataset. The dataset preparation in the study involved several vital steps. Firstly, categorical variables were encoded into a numerical format. This is a common practice in machine learning to ensure that the algorithms can work with the data effectively. Secondly, the Min–Max data normalization technique was applied to the data. This technique ensures that all features have the same scale, essential for many machine learning algorithms to perform optimally. Finally, a stratified split was implemented to ensure a consistent class distribution in the training and testing sets. This strategic division helps to prevent data imbalance and potential model bias that could arise from the underrepresentation or overrepresentation of specific target classes. The study highlights that while Decision Trees and Random Forests offer simplicity and interpretability, they often fall short in predictive accuracy compared to XGBoost, especially when dealing with noisy, high-dimensional, and multicollinear data. XGBoost's ability to balance bias and variance effectively, incorporate regularization techniques, and fine-tune hyperparameters is pivotal in mitigating overfitting and enhancing defect location predictions. Moreover, XGBoost outshines its counterparts by providing more accurate feature importance scores, enabling it to iteratively build robust ensemble models by focusing on previous mistakes and refining feature importance scores with each boosting round.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflict of interest The authors declare no potential conflicts of interest concerning the research presented in this article, authorship, and/or publication of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will

need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Frosini L, Bassi E (2010) Stator current and motor efficiency as indicators for different types of bearing faults in induction motors. *IEEE Trans Industr Electron* 57(1):244–251. <https://doi.org/10.1109/TIE.2009.2026770>
- Mishra C, Samantaray AK, Chakraborty G (2017) Ball bearing defect models: a study of simulated and experimental fault signatures. *J Sound Vib* 400:86–112. <https://doi.org/10.1016/j.jsv.2017.04.010>
- Rao BKN, Srinivasa Pai P, Nagabhushana TN (2012) Failure diagnosis and prognosis of rolling—element bearings using artificial neural networks: a critical overview. *J Phys Conf Ser*. <https://doi.org/10.1088/1742-6596/364/1/012023>
- Ghazaly NM, Stojanovic N, Abd El-Jaber GT (2019) Study various defects of ball bearings through different vibration techniques. *Am J Mech Eng* 1:1
- Cheng H, Zhang Y, Lu W, Yang Z (2019) Research on ball bearing model based on local defects. *SN Appl Sci*. <https://doi.org/10.1007/s42452-019-1251-4>
- Patil AP, Mishra BK, Harsha SP (2021) Fault diagnosis of rolling element bearing using autonomous harmonic product spectrum method. *Proc Inst Mech Eng Part K J Multi Body Dyn* 235(3):396–411. <https://doi.org/10.1177/1464419321994986>
- Imaouchen Y, Alkama R, Thomas M (2015) Bearing fault detection using motor current signal analysis based on wavelet packet decomposition and Hilbert envelope. *MATEC Web Conf EDP Sci*. <https://doi.org/10.1051/mateconf/20152003002>
- Meziani S, Zarour D, Thomas M (2023) Experimental study for early detection of bearing defects by vibration and acoustic emission (Online). Available: <https://hal.archives-ouvertes.fr/hal-03465557>
- Tandon N, Choudhury A (1999) A review of vibration and acoustic measurement methods for detecting defects in rolling element bearings (Online). Available: www.elsevier.com/locate/triboint
- Singh S, Köpke UG, Howard CQ, Petersen D (2014) Analyses of contact forces and vibration response for a defective rolling element bearing using an explicit dynamics finite element model. *J Sound Vib* 333(21):5356–5377. <https://doi.org/10.1016/j.jsv.2014.05.011>
- Sawalhi N, Randall RB (2011) Vibration response of spalled rolling element bearings: observations, simulations and signal processing techniques to track the spall size. *Mech Syst Signal Process* 25(3):846–870. <https://doi.org/10.1016/j.ymssp.2010.09.009>
- Singh S, Howard CQ, Hansen CH (2015) An extensive review of vibration modelling of rolling element bearings with localized and extended defects. *J Sound Vib* 357:300–330. <https://doi.org/10.1016/j.jsv.2015.04.037>
- Salem A, Aly A, Sassi S, Renno J (2018) Time-domain based quantification of surface degradation for better monitoring of the health condition of ball bearings. *Vibration* 1(1):172–191. <https://doi.org/10.3390/vibration1010013>
- Gupta P, Pradhan MK (2017) Fault detection analysis in rolling element bearing: a review (Online). Available: www.sciencedirect.com/www.materialstoday.com/proceedings
- Liu J, Shao Y (2018) Overview of dynamic modelling and analysis of rolling element bearings with localized and distributed faults. *Nonlinear Dyn* 93(4):1765–1798. <https://doi.org/10.1007/s11071-018-4314-y>
- Keller NJ (2020) Condition monitoring systems for axial piston pumps: mobile applications. Purdue University Graduate School, Thesis. <https://doi.org/10.25394/PGS.12202811.v1>
- Eren L (2017) Bearing fault detection by one-dimensional convolutional neural networks. *Math Probl Eng*. <https://doi.org/10.1155/2017/8617315>
- Hoang DT, Kang HJ (2019) Rolling element bearing fault diagnosis using convolutional neural network and vibration image. *Cogn Syst Res* 53:42–50. <https://doi.org/10.1016/j.cogsys.2018.03.002>
- Zhang S, Zhang S, Wang B, Habetler TG (Jan 2019) Machine learning and deep learning algorithms for bearing fault diagnostics—a comprehensive review. <https://doi.org/10.1109/ACCESS.2020.2972859>
- Neupane D, Seok J (2020) Bearing fault detection and diagnosis using case western reserve university dataset with deep learning approaches: a review. *IEEE Access* 8:93155–93178. <https://doi.org/10.1109/ACCESS.2020.2990528>
- Barcelos AS, Marques Cardoso AJ (2021) Current-based bearing fault diagnosis using deep learning algorithms. *Energies* (Basel). <https://doi.org/10.3390/en14092509>
- Munir HS, Ren S, Mustafa M, Siddique CN, Qayyum S (2021) Attention based GRU-LSTM for software defect prediction. *PLoS ONE*. <https://doi.org/10.1371/journal.pone.0247444>
- Liu H, Zhou J, Zheng Y, Jiang W, Zhang Y (2018) Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders. *ISA Trans* 77:167–178. <https://doi.org/10.1016/j.isatra.2018.04.005>
- SCAD College of Engineering and Technology and Institute of Electrical and Electronics Engineers (2018) Proceedings of the international conference on trends in electronics and informatics (ICOEI 2018): 11–12, May 2018
- Jamil MA, Khan MAA, Khanam S (2021) Feature-based performance of SVM and KNN classifiers for diagnosis of rolling element bearing faults. *Vibroeng Proc Extra*. <https://doi.org/10.21595/vp.2021.22307>
- Lu J, Qian W, Li S, Cui R (2021) Enhanced *k*-nearest neighbor for intelligent fault diagnosis of rotating machinery. *Appl Sci* (Switzerland) 11(3):1–15. <https://doi.org/10.3390/app11030919>
- Xie S, Li Y, Tan H, Liu R, Zhang F (2022) Multi-scale and multi-layer perceptron hybrid method for bearings fault diagnosis. *Int J Mech Sci*. <https://doi.org/10.1016/j.ijmecsci.2022.107708>
- de Almeida LF, Bizarria JWP, Bizarria FCP, Mathias MH (2015) Condition-based monitoring system for rolling element bearing using a generic multi-layer perceptron. *JVC/J Vib Control* 21(16):3456–3464. <https://doi.org/10.1177/1077546314524260>
- Rafiee J, Arvani F, Harifi A, Sadeghi MH (2007) Intelligent condition monitoring of a gearbox using artificial neural network. *Mech Syst Signal Process* 21(4):1746–1754. <https://doi.org/10.1016/j.ymssp.2006.08.005>
- Nandi S, Toliyat HA, Li X (2005) Condition monitoring and fault diagnosis of electrical motors—a review. *IEEE Trans Energy Convers* 20(4):719–729. <https://doi.org/10.1109/TEC.2005.847955>
- Dai J, Wang J, Huang W, Shi J, Zhu Z (2020) Machinery health monitoring based on unsupervised feature learning via generative adversarial networks. *IEEE/ASME Trans Mechatron* 25(5):2252–2263. <https://doi.org/10.1109/TMECH.2020.3012179>
- Pandarakone SE, Gunasekaran S, Mizuno Y, Nakamura H (Oct. 2018) Application of Naive Bayes classifier theorem in detecting induction motor bearing failure. In: Proceedings—2018 23rd international conference on electrical machines, ICEM 2018. Institute of Electrical and Electronics Engineers Inc., pp 1761–1767. <https://doi.org/10.1109/ICELMACH.2018.8506836>
- Xu Q, Fan Z, Jia W, Jiang C (2019) Quantile regression neural network-based fault detection scheme for wind turbines with

- application to monitoring a bearing. *Wind Energy* 22(10):1390–1401. <https://doi.org/10.1002/we.2375>
34. Soualhi A, Medjaher K, Zerhouni N (2015) Bearing health monitoring based on hilbert-huang transform, support vector machine, and regression. *IEEE Trans Instrum Meas* 64(1):52–62. <https://doi.org/10.1109/TIM.2014.2330494>
 35. Huang X, Wen G, Dong S, Zhou H, Lei Z, Zhang Z, Chen X (2021) Memory residual regression autoencoder for bearing fault detection. *IEEE Trans Instrum Meas*. <https://doi.org/10.1109/TIM.2021.3072131>
 36. Amarnath M, Sugumaran V, Kumar H (2013) Exploiting sound signals for fault diagnosis of bearings using decision tree. *Measurement (Lond)* 46(3):1250–1256. <https://doi.org/10.1016/j.measurement.2012.11.011>
 37. Nguyen N-T, Lee H-H (2008) Decision tree with optimal feature selection for bearing fault detection. *J Power Electron* 8(1):101–107 (uci: **G704-001582.2008.8.1.010**)
 38. Euldji R, Boumahdi M, Bachene M (2021) Decision-making based on decision tree for ball bearing monitoring. In: 2020 2nd international workshop on human-centric smart environments for health and well-being (IHSH), pp 171–175. <https://doi.org/10.1109/IHSH51661.2021.9378734>
 39. Sugumaran V, Muralidharan V, Ramachandran KI (2007) Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. *Mech Syst Signal Process* 21(2):930–942. <https://doi.org/10.1016/j.ymsp.2006.05.004>
 40. Sugumaran V, Ramachandran KI (2007) Automatic rule learning using decision tree for fuzzy classifier in fault diagnosis of roller bearing. *Mech Syst Signal Process* 21(5):2237–2247. <https://doi.org/10.1016/j.ymsp.2006.09.007>
 41. Senanayaka JSL, van Khang H, Robbersmyr KG (2017) Towards online bearing fault detection using envelope analysis of vibration signal and decision tree classification algorithm. In: 2017 20th international conference on electrical machines and systems (ICEMS), pp 1–6. <https://doi.org/10.1109/ICEMS.2017.8056146>
 42. Choudhary A, Goyal D, Letha SS (2021) Infrared thermography-based fault diagnosis of induction motor bearings using machine learning. *IEEE Sens J* 21(2):1727–1734. <https://doi.org/10.1109/JSEN.2020.3015868>
 43. Li Q, Li H, Hu W, Sun S, Qin Z, Chu F (2024) Transparent operator network: a fully interpretable network incorporating learnable wavelet operator for intelligent fault diagnosis. *IEEE Trans Ind Inform*. <https://doi.org/10.1109/TII.2024.3366993>
 44. Patil S, Phalle V (2018) Fault detection of anti-friction bearing using ensemble machine learning methods. *Int J Eng Trans B* 31(11):1972–1981. <https://doi.org/10.5829/ije.2018.31.11b.22>
 45. Nistane V, Harsha S (2018) Performance evaluation of bearing degradation based on stationary wavelet decomposition and extra trees regression. *World J Eng* 15(5):646–658. <https://doi.org/10.1108/WJE-12-2017-0403>
 46. Xu G, Liu M, Jiang Z, Söffker D, Shen W (2019) Bearing fault diagnosis method based on deep convolutional neural network and random forest ensemble learning. *Sensors (Switzerland)*. <https://doi.org/10.3390/s19051088>
 47. Liu A, Yang Z, Li H, Wang C, Liu X (2022) Intelligent diagnosis of rolling element bearing based on refined composite multi-scale reverse dispersion entropy and random forest. *Sensors*. <https://doi.org/10.3390/s22052046>
 48. Mitchell R, Frank E (2017) Accelerating the XGBoost algorithm using GPU computing. *PeerJ Comput Sci* 7:2017. <https://doi.org/10.7717/peerj-cs.127>
 49. Trizoglou P, Liu X, Lin Z (2021) Fault detection by an ensemble framework of Extreme Gradient Boosting (XGBoost) in the operation of offshore wind turbines. *Renew Energy* 179:945–962. <https://doi.org/10.1016/j.renene.2021.07.085>
 50. Zhang R, Li B, Jiao B (2019) Application of XGboost algorithm in bearing fault diagnosis. *IOP Conf Ser Mater Sci Eng*. <https://doi.org/10.1088/1757-899X/490/7/072062>
 51. Qi M, Zhou R, Zhang Q, Yang Y (2021) Feature classification method of frequency cepstrum coefficient based on weighted extreme gradient boosting. *IEEE Access* 9:72691–72701. <https://doi.org/10.1109/ACCESS.2021.3079286>
 52. Xia L, Zheng P, Li J, Tang W, Zhang X (2022) Privacy-preserving gradient boosting tree: vertical federated learning for collaborative bearing fault diagnosis. *IET Collabor Intell Manuf*. <https://doi.org/10.1049/cim2.12057>
 53. Cai G, Yang C, Pan Y, Lv J (2019) EMD and GNN-adaboost fault diagnosis for urban rail train rolling bearings. *Discrete Continuous Dyn Syst Ser S* 12(4–5):1471–1487. <https://doi.org/10.3934/dcdss.2019101>
 54. Xia T, Zhuo P, Xiao L, Du S, Wang D, Xi L (2021) Multi-stage fault diagnosis framework for rolling bearing based on OHF Elman AdaBoost-Bagging algorithm. *Neurocomputing* 433:237–251. <https://doi.org/10.1016/j.neucom.2020.10.003>
 55. Yao P, Liu Z, Wang Z, Bu S (2012) Fault signal classification using adaptive boosting algorithm. *Elektron Elektrotech* 18(8):97–100. <https://doi.org/10.5755/j01.eee.18.8.2635>
 56. Yuan Z, Zhou T, Liu J, Zhang C, Liu Y (2021) Fault diagnosis approach for rotating machinery based on feature importance ranking and selection. *Shock Vib*. <https://doi.org/10.1155/2021/8899188>
 57. Zhang C, Kong L, Xu Q, Zhou K, Pan H (2021) Fault diagnosis of key components in the rotating machinery based on Fourier transform multi-filter decomposition and optimized LightGBM. *Meas Sci Technol* 32(1):015004. <https://doi.org/10.1088/1361-6501/aba93b>
 58. Nemat Saberi A, Belahcen A, Sobra J, Vaimann T (2022) LightGBM-based fault diagnosis of rotating machinery under changing working conditions using modified recursive feature elimination. *IEEE Access* 10:81910–81925. <https://doi.org/10.1109/ACCESS.2022.3195939>
 59. Liu S, Ji Z, Wang Y, Zhang Z, Xu Z, Kan C (2021) Multi-feature fusion for fault diagnosis of rotating machinery based on convolutional neural network. *Comput Commun* 173:160–169. <https://doi.org/10.1016/j.comcom.2021.04.016>
 60. Xu Y, Cai W, Wang L, Xie T (2021) Intelligent diagnosis of rolling bearing fault based on improved convolutional neural network and lightGBM. *Shock Vib*. <https://doi.org/10.1155/2021/1205473>
 61. Jia X, Xiao B, Zhao Z, Ma L, Wang N (2021) Bearing fault diagnosis method based on CNN-LightGBM. *IOP Conf Ser Mater Sci Eng*. <https://doi.org/10.1088/1757-899X/1043/2/022066>
 62. Kiranyaz S, Devecioglu OC, Alhams A, Sassi S, Ince T, Abdeljaber O, Avci O, Gabbouj M (2024) Zero-shot motor health monitoring by blind domain transition. *Mech Syst Signal Process* 210:111147. <https://doi.org/10.1016/j.ymsp.2024.111147> (ISSN **0888-3270**)
 63. Xu Y, Wang E, Yang Y, Chang Y (2022) A unified collaborative representation learning for neural-network based recommender systems. *IEEE Trans Knowl Data Eng* 34(11):5126–5139. <https://doi.org/10.1109/TKDE.2021.3054782>
 64. Tao Y, Shi J, Guo W, Zheng J (2023) Convolutional neural network based defect recognition model for phased array ultrasonic testing images of electrofusion joints. *ASME J Press Vessel Technol* 145(2):024502. <https://doi.org/10.1115/1.4056836>
 65. Zheng W, Lu S, Yang Y, Yin Z, Yin L (2024) Lightweight transformer image feature extraction network. *PeerJ Comput Sci* 10:e1755. <https://doi.org/10.7717/peerj-cs.1755>
 66. Shi M-L, Lv L, Xu L (2023) A multi-fidelity surrogate model based on extreme support vector regression: fusing different

- fidelity data for engineering design. *Eng Comput* 40(2):473–493. <https://doi.org/10.1108/EC-10-2021-0583>
67. Li S, Chen H, Chen Y, Xiong Y, Song Z (2023) Hybrid method with parallel-factor theory, a support vector machine, and particle filter optimization for intelligent machinery failure identification. *Machines* 11(8):837. <https://doi.org/10.3390/machines11080837>
 68. Hu X, Tang T, Tan L, Zhang H (2023) Fault detection for point machines: a review, challenges, and perspectives. *Actuators* 12(10):391. <https://doi.org/10.3390/act12100391>
 69. Jing X, Wu Z, Zhang L, Li Z, Mu D (2024) Electrical fault diagnosis from text data: a supervised sentence embedding combined with imbalanced classification. *IEEE Trans Industr Electron* 71(3):3064–3073. <https://doi.org/10.1109/TIE.2023.3269463>
 70. Sharma V, Parey A (2016) A review of gear fault diagnosis using various condition indicators. *Proc Eng*. <https://doi.org/10.1016/j.proeng.2016.05.131>
 71. Sassi S, Badri B, Thomas M (2006) ‘TALAF’ and ‘THIKAT’ as innovative time domain indicators for tracking ball bearings (2006). <https://profs.etsmtl.ca/mthomas/Publications/Publications/A19-sassi.pdf>
 72. Wang J, Qiao L, Ye Y, Chen YQ (2017) Fractional envelope analysis for rolling element bearing weak fault feature extraction. *IEEE/CAA J Autom Sin* 4(2):353–360. <https://doi.org/10.1109/JAS.2016.7510166>
 73. Chen Z, Deng S, Chen X, Li C, Sanchez RV, Qin H (2017) Deep neural networks-based rolling bearing fault diagnosis. *Microelectron Reliab* 75:327–333. <https://doi.org/10.1016/j.microrel.2017.03.006>
 74. Rai VK, Mohanty AR (2007) Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert–Huang transform. *Mech Syst Signal Process* 21(6):2607–2615. <https://doi.org/10.1016/j.ymsp.2006.12.004>
 75. William PE, Hoffman MW (2011) Identification of bearing faults using time domain zero-crossings. *Mech Syst Signal Process* 25(8):3078–3088. <https://doi.org/10.1016/j.ymsp.2011.06.001>
 76. IEEE Region 10. Colloquium (3rd: 2008: Indian Institute of Technology Kharagpur), Institute of Electrical and Electronics Engineers. Kharagpur Section., IEEE Sri Lanka Section., and Damodar Valley Corporation., IEEE Region 10 Colloquium and Third International Conference on Industrial and Information Systems: ICIS-2008, December 8–10, 2008: theme: "Real-time communicative intelligence for tomorrow's industry": e-proceedings. IEEE, 2008
 77. Zarei J (2012) Induction motors bearing fault detection using pattern recognition techniques. *Expert Syst Appl* 39(1):68–73. <https://doi.org/10.1016/j.eswa.2011.06.042>
 78. Helmi H, Forouzantabar A (2019) Rolling bearing fault detection of electric motor using time domain and frequency domain features extraction and ANFIS. *IET Electr Power Appl* 13(5):662–669. <https://doi.org/10.1049/iet-epa.2018.5274>
 79. Nayana BR, Geethanjali P (2017) Analysis of statistical time-domain features effectiveness in identification of bearing faults from vibration signal. *IEEE Sens J* 17(17):5618–5625. <https://doi.org/10.1109/JSEN.2017.2727638>
 80. Azen R, Budescu DV (2003) The dominance analysis approach for comparing predictors in multiple regression. *Psychol Methods* 8(2):129–148. <https://doi.org/10.1037/1082-989X.8.2.129>
 81. Charbuty B, Abdulazeez A (2021) Classification based on decision tree algorithm for machine learning. *J Appl Sci Technol Trends* 2(01):20–28. <https://doi.org/10.38094/jastt20165>
 82. Breiman L (2001) Random forests. *Mach Learn* 45:5–32. <https://doi.org/10.1023/A:1010933404324>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.