IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Novel Machine Learning Techniques for Classification of Rolling Bearings

Quynh Nguyen Xuan Phan[1], Tuan Minh Le[3], Hieu Minh Tran[3], Ly Van Tran[1],and Son Vu Truong Dao[1,2]

[1] School of Industrial Engineering and Management, International University, Vietnam National University Ho Chi Minh City, HCMC, Vietnam
[2] School of Science, Engineering and Technology, RMIT University Vietnam, Ho Chi Minh City, Vietnam
[3] School of Electrical Engineering, International University, Vietnam National University Ho Chi Minh City, HCMC, Vietnam

Corresponding author: Son Vu Truong Dao (dvtson@hcmiu.edu.vn, son.daovutruong@rmit.edu.vn). Orcid: 0000-0001-9281-4869

ABSTRACT Rolling bearing faults frequently cause rotating equipment failure, leading to costly downtime and maintenance expenses. As a result, researchers have focused on developing effective methods for diagnosing these faults. In this paper, we explore the potential of Machine Learning (ML) techniques for classifying the health status of bearings. Our approach involves decomposing the signal, extracting statistical features, and using a feature selection employing Binary Grey Wolf Optimization. We then employ four different classifiers such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT), and Random Forest (RF) to diagnose faults based on the reduced set of features. To evaluate the performance of our methods, we utilize several performance indicators. Our results demonstrate that four Machine Learning methods can achieve a high-accuracy fault classification result of 99.85%, better than state-of-the-art methods, highlighting their potential for use in predictive maintenance applications.

INDEX TERMS Fault diagnosis, Feature Selection, Grey Wolf Optimization, Machine Learning, Rolling Bearing

I. INTRODUCTION

Rolling bearing or rolling element bearing is used in various rotating engines. It is known for its ability to support rotating elements, transferring loads between distinct parts of the machines. As a common but primary machinery component, rolling bearing receives great attention; thus, predicting its condition has become one of the most interesting research areas. Diagnosing rolling bearing faults can bring tremendous benefits to the industry since such fault recognition helps people locate potential breakdown machines before the condition worsens, directly saving money and time, ensuring a smooth working environment. These benefits emphasize the need to develop a system that can predict bearing conditions efficiently. Numerous technological systems, such as automotive systems [1], electromechanical systems [2], and industrial systems [3] have investigated the use of ML approaches in problem diagnostics. Therefore, more systems that applied Machine Learning methods were raised to help classify rolling bearing faults [4], [5].

Traditional processes of fault recognition often require experts' experiences and knowledge [6], [7]. Such techniques are also time-consuming and unreliable, making people fail to label rolling bearing faults in time. For these reasons, to reduce the dependence of the diagnosis process on manual factors, more Machine Learning techniques are raised and developed. These techniques are parts of Artificial Intelligence that allow systems to learn and solve identical problems automatically. Using Machine Learning, rolling bearing fault diagnosis can now be done promptly, proving that automatic failure recognition is no longer out of reach. However, although Machine Learning is undoubtedly more effective than traditional methods, these techniques encountered practical problems with high dimensional data. To predict the conditions of rolling bearing, analyzing high dimensional data is inevitable since faults are different in categories and sizes, and the vibration signals can vary depending on different working conditions. This provokes another challenge for bearing failure diagnosis.

This study focuses on extracting and labeling vibration signals using Machine Learning algorithms. The purpose of the dissertation is to classify the bearing fault types and their severities within extracted features. The Case Western Reserve University bearing fault dataset (CWRU) was used for verification and validation. In this research, we decompose signals using two different methods which are the Empirical Mode Decomposition (EMD) and Ensemble Empirical Mode Decomposition (EEMD) methods. Then, we

calculate the decomposed signals to achieve different statistical features. At this point, the features are selected using the Binary Grey Wolf Optimization method and then classified using four Machine Learning methods. Finally, some performance indicators are conducted to verify all methods. However, the method also faces several challenges. There is a lot of need to deal with huge and high-dimensional datasets that are accompanied by the cost of evaluating them, which results in expensive computational time. Furthermore, although there are competitive findings, there is still scope for doing better. To overcome these practical challenges and improve the model performance, future research should investigate other metaheuristics. The suggested method has great potential; however, it can only be effectively used in industries if these obstacles are addressed.

The other parts of this paper are organized in the following manner. The second and the third sections mention recent remarkable studies and the materials and methods, respectively. The fourth one describes experiments and results. Then, in the final section, the discussion and conclusions are presented.

II. RELATED WORKS

Many bearing fault diagnosis studies heretofore achieved remarkable results by analyzing vibration signals. In the context of this paper, the following noticeable journals using different Machine Learning techniques are presented.

The paper [8] presents a new technique for diagnosing rolling element bearings (REBs). Raw signal data is converted to 2D images by continuous wavelet transform (CWT) to facilitate efficient feature extraction. The authors employ SE-ResNet152 which gives better results than other models in terms of accuracy (96.42 %), precision (95.84 %), recall (96.96 %), and F1 score (96.31 %) on the Case Western Reserve University (CWRU) dataset for automatic identification of fault patterns.

In [9], the author describes an innovative approach for diagnosing faults in rolling-element bearings (REBs). Specifically, the authors propose a triboelectric nanogenerator(TENG)-embedded bearing, called T-bearing, that directly detects defects in rolling balls through triboelectric signals produced from the interaction between various components of the bearing. These signals are processed using Seasonal and Trend decomposition via Loess(STL) and AutoML with Bayesian optimization(BO); hence it is possible to reach a high recognition accuracy i.e., 99.48%, for differentiating diverse bearing ball statuses. This method markedly outperforms conventional means which only attain 78.34% classification without decomposing signals into seasonal-trend level fashion.

The authors in [10] researched bearing fault classification using accelerated Deep Learning. Indeed, this method used accelerated Convolutional Neural Networks (CNN) which compressed and speeded up normal CNN. The mentioned method was able to classify accurately 97.2% of the three bearing fault types.

Within this year, Li et al. (2019) [11] provided an Attention Mechanism system to classify bearing faults. By extracting and visualizing informative data segments, with the use of the Attention Mechanism, this method achieved a testing accuracy rate of 97.72% given only one labeled sample of each bearing condition, proving the ability to learn the pattern of the mentioned technique.

Also in 2019, Xu et al. (2019) [12] combined the original Convolutional Neural Network (CNN) technique with the original Random Forest (RF) technique to provide a system for bearing fault classification. By using CNN for multi-level feature extraction and then training different RF classifiers, the authors achieved an accuracy rate of 99.73%.

In 2020, Yuan et al. (2020) [13] introduced a classification method that was a combination of the original CNN technique and the original Support Vector Machine (SVM) technique. The method utilized CNN for feature extraction and then SVM for classification, solving the problem of in-depth signal feature extraction and the challenge of collecting massive fault signal datasets in real - life. By this combined model, the authors were able to achieve 98.75% average classification accuracy and an average training time of 35.82s, which was shorter than standard CNN (370s).

With the use of SVM, instead of combining techniques, Li, Yang, et al., 2019 [14] proposed the Deep Stacking Least Squares Support Vector Machine (DSLS - SVM) method for bearing classification. By training thirty modules of DSLS-SVM independently, the faults were managed and recognized with a mean testing accuracy of 99.90%, which is significantly higher than the accuracy of other methods Least Square – Support Vector Machine (LS-SVM: 76.60%, Kernel Ridge Regression (KRR): 73.65%, Kernel version of Deep Stacking Network (KDSN): 93.15%, Kernel Deep Regression Network (KDRN): 92.60%.

Recent developments have focused on integrating digital twins, explainable AI, and the Industrial Internet of Things (IIoT) to allow for more accurate and autonomous predictive maintenance (Pdm) systems. Research such as that conducted by Ucar et al. (2024), which explores the use of AI for predicting failures and scheduling preventative maintenance demonstrates the importance of reliability and future directions in AI-based PdM [15]. Similarly, Kaparthi and Bumblauskas (2020) show how decision trees are used concerning machine learning techniques in different industrial applications revealing their scalability and adaptability [16].

Despite these advances, deep learning methods have significant challenges when it comes to fault classification in PdM due to small datasets that do not allow for training complex models. Current research thus often makes use of strong feature selection methods data augmentation approaches, transfer learning, etc., thereby improving model performance.

To bridge the gap, this paper employs empirical mode decomposition (EMD) and ensemble empirical mode decomposition (EEMD) for feature extraction, then Grey Wolf Optimization (GWO) for feature selection. To enhance fault classification accuracy, classic ML classifiers like SVM, KNN, DT, and RF are combined with this approach. In so doing, the paper aims to provide a more reliable and scalable solution to PdM that bridges the gap between classical ML methods and the need for more advanced fault classification techniques in industry.

III. THEORETICAL BACKGROUND

A. DECOMPOSITION METHODS

Decomposition is key in fault classification because it effectively works on non-linear and non-stationary signals, which are often complex and noisy. Raw signals can obscure relevant features essential for accurate classification. Methods like Empirical Mode Decomposition (EMD) [17], Ensemble Empirical Mode Decomposition (EEMD)[18], and Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) stand out in tackling this challenge [19]. EMD decomposes complex signals into simpler parts called intrinsic mode functions (IMFs), showing underlying trends and patterns. EEMD makes this process better by adding noise to cut down on mode mixing, making it more stable and clearer. CEEMDAN goes a step further by adaptively adding noise, leading to an even more accurate separation of signal parts. The choice of EMD, EEMD, and CEEMDAN is backed by their ability to adaptively analyze signals, picking out fault-related details from unneeded noise. These methods make sure the extraction of important features is strong and insightful, thus improving the ability of basic machine learning classifiers to tell apart faults in tasks.

1) EMPIRICAL MODE DECOMPOSITION (EMD)

This method breaks down multi-component signals into IMFs-Intrinsic Mode Functions which are modes that connect to the physical characteristics of the signals.

The general algorithm involved the 3 main steps [20]:

- Identify all local extrema of the signal function.
- Build the mean envelope using lower and upper envelopes.
- Achieve the residual by subtracting the mean from the signals.

Even though EMD is an efficient algorithm in signal processing; some drawbacks cannot be underestimated such as the mode aliasing endpoint effect among others. An improved version of EMD called EEMD has been proposed to solve the problem associated with mode aliasing [21].

2) ENSEMBLE EMPIRICAL MODE DECOMPOSITION (EEMD)

Modified from the original technique, EEMD is considered an effective approach to deal with the mode mixing issue [20]. This method involved 3 steps :

- Form signals x' from the equation:

$$x^i(t) = x(t) + \omega^i(t) \quad (1)$$

With $\omega^i(t)$ is defined as the white noise realization.

- Use EMD and decompose x^i to k modes of IMF.
- Calculate the final decomposition by averaging all corresponding modes.

Note that each trial has its independent processes in EEMD which leads to getting each one a residual component. Moreover, if trials are not enough to reduce the added Gauss white noise completely, the following issues may occur 1) incomplete decomposition and 2) reconstruction error.

3) COMPLETE ENSEMBLE EMPIRICAL MODE DECOMPOSITION WITH ADAPTIVE NOISE (CEEMDAN)

To reduce reconstruction error and make the reconstructed signal noiseless, complementary CEEMDAN has been suggested. It replaces the EEMD's random white noise with adaptive white noise without a basis function [22] and calculates the unique residual signal to get individual IMFs that result in less computation for complete signal reconstruction. Specifically, the following are the steps followed in implementing the CEEMDAN algorithm:

(1) Let $E_j(\cdot)$ stand for the jth modal component decomposed by the EMD method. Consider $y(t)$ as the basic signal to be decomposed, $\omega(t)$ as Gaussian white noise, ε as the white noise level added for the kth time, imf_j^* as the jth modal component obtained by CEEMDAN decomposition, and imf_{ij} as the ith modal component obtained by EMD in the jth stage. Then the signal made by CEEMDAN adds noise to the initial signal based on a set signal-to-noise level $y(t) + \varepsilon_0 \omega_i(t)$ $(i = 1,2,\ldots,I)$, Perform EMD decomposition for I times, and then average the outcomes to find the first modal component:

$$imf_1^*(t) = \frac{1}{I}\sum_{i=1}^{I} imf_{i1}(t) \quad (2)$$

(2) Assuming $r_k(t)$ represents the kth residual component, we then compute the initial residual component of the first phase, $j = 1$, in this manner:

$$r_1(t) = y(t) - imf_1^*(t) \quad (3)$$

The signal $r_1(t) + \varepsilon_1 E(\omega_i(t))$, where i ranges from 1 to I, undergoes decomposition through EMD to extract its initial modal component. Following this, the secondary mode is characterized as:

$$imf_2^*(t) = \frac{1}{I}\sum_{i=1}^{I} E_1(r_1(t) + \varepsilon_1 E(\omega_i(t))) \quad (4)$$

(4) Define the jth-order signal as:

$$\mathrm{r_j(t)} = \mathrm{r_{j-1}(t)} - \mathrm{imf_j^*(t)} \quad (5)$$

(5) Decompose the signal $\mathrm{r_j(t)} + \varepsilon_j E_j(\omega_i(t))$ (where i goes from 1 to I) using EMD and get the jth modal piece. Next, the $(j+1)th$ mode of CEEMDAN is:

$$\mathrm{imf_{j+1}^*(t)} = \frac{1}{I}\sum_{i=1}^{I} E_1(r_j(t) + \varepsilon_j E_j(\omega_i(t))) \quad (6)$$

(6) Increase j by 1 and do step (4) again until the resulting residual component cannot be decomposed. At this point, there are J total modes. Here, the residual component is:

$$R(t) = r(t) - \sum_{j=1}^{j} \mathrm{imf_j^*(t)} \quad (7)$$

The modal and residual component of the first signal, found through CEEMDAN break down, are shown as such:

$$y(t) = \sum_{j=1}^{j} \mathrm{imf_j^*(t)} + \mathrm{R(t)} \quad (8)$$

B. FEATURE EXTRACTION

Aiming to reveal the characteristics of input vibration signals, feature extraction is determined as a step that reduces the number of data dimensions and is beneficial in lessening needed resources [23]. There are myriad feature extraction methods. However, within the context of this research, only the following time domain and frequency domain statistical features were performed.

- Maximum:
$$xmax=(xi) \quad (9)$$
- Minimum:
$$xmin=(xi) \quad (10)$$
- Mean:
$$x_{mean} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (11)$$
x_i: each value from the population
- Standard Deviation:
$$x_{std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - x_{mean})^2} \quad (12)$$
- Root Mean Square:
$$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} {x_i}^2} \quad (13)$$
- Skewness:
$$x_{sk} = \frac{1}{N}\sum_{i=1}^{N}\frac{(x_i - x_{imean})^3}{x_{std}^3} \quad (14)$$
- Kurtosis:
$$x_{kur} = \frac{1}{N}\sum_{i=1}^{N}\frac{(x_i - x_{imean})^4}{x_{std}^4} \quad (15)$$
- Shape factor:
$$x_{shape} = \frac{\mathrm{x}_{rms}}{\bar{x}} \quad (16)$$
- Mean Frequency:
$$f_{meanfr} = \frac{1}{N}\sum_{j=1}^{N} f_i \quad (17)$$
- Power bandwidth:
$$x_{powerbw} = powerbw(x_i) \quad (18)$$
- Crest Factor:
$$x_{crest} = \frac{|x_i|}{x_{rms}} \quad (19)$$
- Form:
$$x_{form} = \frac{x_{rms}}{x_{mean}} \quad (20)$$
- Peak to peak:
$$x_{p2p} = |x_i| - |x_i| \quad (21)$$
- Variance:
$$x_{var} = \frac{1}{N}\sum_{i=1}^{N}(x_i - x_{mean})^2 \quad (22)$$
- Median:
$$\begin{cases} x_{\frac{n}{2}} \; for \; odd \\ \dfrac{x_{\frac{n+1}{2}} + x_{\frac{n+1}{2}}}{2} \; for \; even \end{cases} \quad (23)$$
- Mean Square:
$$x_{ms} = \frac{1}{N}\sum_{i=1}^{N} x_i^2 \quad (24)$$
- Margin:

$$x_{margin} = \frac{|x_i|}{(\frac{1}{N}\sum_{i=1}^{N}\sqrt{|x_i|})^2} \quad (25)$$

- Impulse:

$$x_{impulse} = \frac{|x_i|}{x_{mean}} \quad (26)$$

- Median Frequency:

$$f_{medfr} = medfreg(x_i) \quad (27)$$

- Band Power:

$$x_{bandpw} = bandpower(x_i) \quad (28)$$

C. MACHINE LEARNING METHODS

1) SUPPORT VECTOR MACHINE (SVM)

Given small datasets, this technique can bring high accuracy rates in classification and regression problems. Its basic idea is to separate the dataset into two half spaces by defining an optimal hyperplane. SVM can bring a high accuracy rate in classification problems, given a small dataset. Furthermore, it is also suitable for high-dimensional data. This is because SVM has a strong ability in generalization and high spare representation. Its advantages also include the ability to overcome overfitting situations. On the other hand, SVM also has its disadvantages. One of the disadvantages is that this method depends significantly on its chosen parameters and kernel choices. In particular, SVM is not a suitable tool when it comes to redundant data. Furthermore, SVM cannot generate discriminative features, given raw input

The algorithm of SVM is defined in Figure 1. To classify data points of dissimilar categories, as a rule, this method tries to find one optimal hyperplane so that the largest distance from the hyperplane to the parallel lines that pass through support vectors [13] can be obtained.

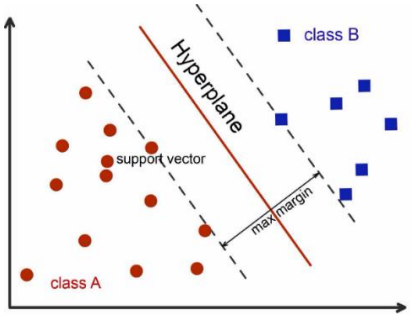

FIGURE 1. **The architecture of the Support Vector Machine**

2) K - NEAREST NEIGHBORS (KNN)

K- Nearest Neighbors is a beneficial supervised method for problems such as classification or regression. In terms of classification, the label of one new data is viewed and predicted based on the K-nearest neighbor points of this data in the training set. Then, we can achieve the classification results through major voting between points or assigning weights scores to the neighbors [7]. KNN is less complex than the other Machine Learning methods. K- Nearest Neighbors uses near data points to categorize unlabeled points into groups; therefore, performing k-NN's prediction is simple and quicker than other methods. The architecture of KNN is indicated in Figure 2.

On the other hand, k-NN also has its disadvantages. As its prediction is based on near data points, k-NN is quite sensitive to outliers and noise points, especially when given a small K number. This may lead to low-accuracy classifications. When K is a considerable number, the k-NN method faces another disadvantage. K-NN is a technique whose prediction is based on testing not a training process. The distances between data points computation may be expensive, especially with large datasets or high dimensional data. More considerable K results more complex process which directly affects the prediction of this method.

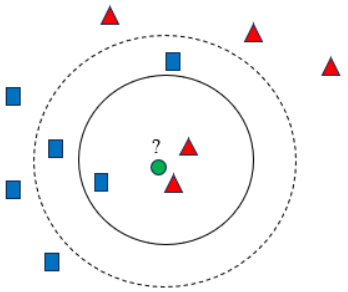

FIGURE 2. **The architecture of K -Nearest Neighbor**

3) DECISION TREE (DT)

In general, this method classifies groups by constructing a tree-structured model. The model contains main nodes called decision nodes which are potential options and leaf nodes which are the achieved result if one decision were made [24]. On the one hand, DT is a simple Machine Learning method. This method is well-known and widely used thanks to its simple characteristic and easy-to-understand algorithm. DT can analyze missing values datasets and cope with both numerical data and categorical data. It is also suitable for large datasets. On the other hand, DT is sensitive and depends on the dataset. The Decision Tree structure changes differently even with slight changes in the dataset. Furthermore, this technique may lead to overfitting problems.

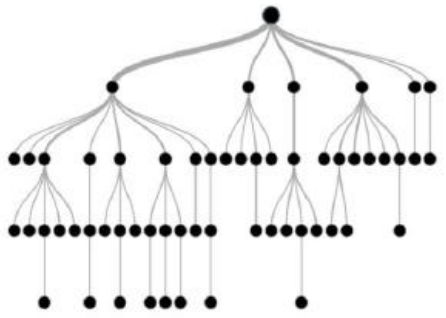

FIGURE 3. **The Architecture of Decision Tree**

4) RANDOM FOREST (RF)

Random Forest applies a bootstrap aggregating to tree learners [25]. This technique aims to get accurate predictions by merging different decision trees.

RF is known for its ability to achieve a high accuracy rate in diagnostic problems. One of the strengths of RF is that it can overcome overfitting errors. This method also can automatically fulfill the dataset's missing values. one of the disadvantages of RF is that this technique needs a large dataset for constructing highly accurate models. Due to this fact, performing the RF method often results in quite complex computation when predicting. Figure 4 describes the architecture of the Random Forest model.

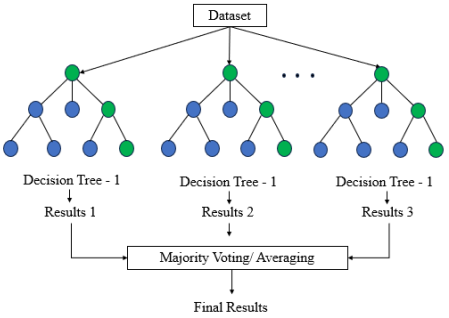

FIGURE 4. **The architecture of Random Forest**

D. FEATURE SELECTION – GREY WOLF OPTIMIZATION (GWO)

Feature Selection is a process that selects subsets of features and variables out of an original plenty-of-feature dataset [12]. By performing this process, only important and useful data is maintained, reducing the computational cost and time and at the same time, on occasions, improving the accuracy of the models. This is mainly because feature selection reduces the high-dimensionality characteristics of the dataset by eliminating the redundant data, making it easier and more effective for other later tools to analyze the data [26], [27], [28], [29].

The three main methods in performing Feature Selection are the Intrinsic approach, Filter approach, and Wrapper approach. The Intrinsic approach involves techniques that select features and is viewed as one of their steps in performing Machine Learning. The Filter approach uses statistical computing filters to investigate the relationships between data and find the best input sets. The Wrapper approach results in selecting features by choosing subsets of features that give out the best performance.

1) GREY WOLF OPTIMIZATION - GWO

First proposed by the author Mirjalili et al. [30], this new technique is known as a metaheuristic optimization method that simulates the hunting prey processes of a pack of five to twelve grey wolves. Regarding GWO [31], to imitate the nature of grey wolves, the wolves in this method are separated into four levels which are the main leader Alpha, the leader supporters Beta and Delta, and the member Omega (Figure 5).

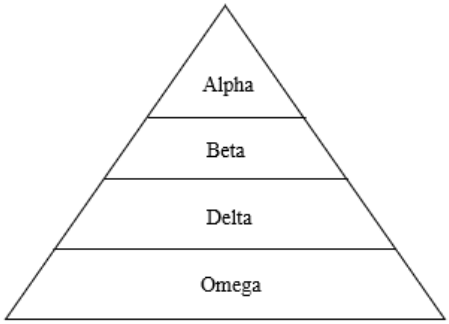

FIGURE 5. **Position of Grey Wolves' pack**

According to the rule, the highest position in the pack is the wolf named Alpha. This one here is appointed as the pack's main leader and plays the role of the decision-maker. The second and third highest appointments of the wolves' pack are respectively the Beta and the Delta wolves which play the role of supporting Alpha in managing the pack. The Omega ones are responsible for obeying their leaders and managers to surround their prey (Figure 6).

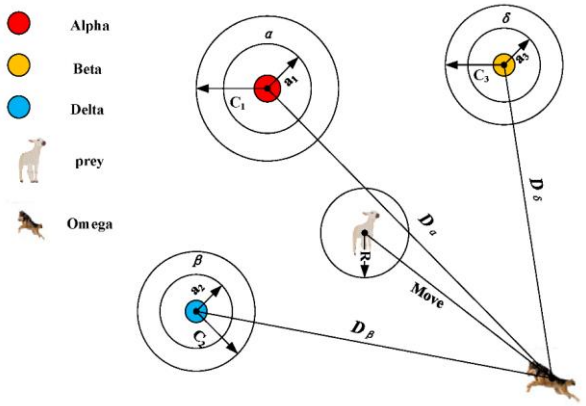

FIGURE 6. **Wolves' ways of hunting.**

The algorithm of the GWO method can be presented as follows [31]

The encircling positions of the wolves in the pack can be calculated through:

$$\vec{X}_{(t+1)} = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \tag{29}$$

With $\vec{X}_{(t+1)}$ defined dolf member's position within the t+1 iteration, $\vec{X}_p(t)$ defined as the prey's location at iteration t, $\vec{A}$ as the coefficient, $\vec{D}$ as the distance from wolf to prey

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}_t\right| \tag{30}$$

With $\vec{C}$ as the coefficient

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (31)$$

$$\vec{C} = 2\vec{r}_2 \quad (32)$$

With $\vec{r}_1$ as the first random number of [0,1], $\vec{r}_2$ as the second random number of [0,1], $\vec{a}$ as the coefficient that reduces linearly between [2,0]

$$\vec{a} = 2 - 2\left(\frac{t}{T}\right) \quad (33)$$

With t defined as the current iteration, and T as the identified maximum number of iterations

The equation for updating the current location of one Omega wolf member that follows their three-level leader wolves is:

$$\vec{X}_{(t+1)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \quad (34)$$

Knowing that:

$$\vec{X}_1 = \left|\vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha\right| \quad (35)$$

$$\vec{X}_2 = \left|\vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta\right| \quad (36)$$

$$\vec{X}_3 = \left|\vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta\right| \quad (37)$$

With $\vec{X}_1$ as the Alpha main leader's position, $\vec{X}_2$ as the Beta's position, and $\vec{X}_3$ as the Delta's position.

2) BINARY GREY WOLF OPTIMIZATION (BGWO)

As the name of this technique is Binary Grey Wolves Optimization, this technique uses a binary vector to result in the wolves' positions [31], [32]. The binary vector is as follows:

$$X^d_{(t+1)} = \begin{cases} 1, \text{in case sigmoid}\left(\frac{X^d_1 + X^d_2 + X^d_3}{3}\right) \geq r \\ 0, \text{otherwise} \end{cases} \quad (37)$$

With r as a random number of [0,1]

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-10(x-0.5)}} \quad (38)$$

When x= 1, the feature is chosen.

IV. METHODOLOGY

The method shown in Figure 7 starts by loading the data, and then dividing it into two parts for different uses. Next, the data is preprocessed and clean for study. After that, it's analyzed with two techniques: Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD), and Complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN). Features are then pulled out from both the time and frequency areas. Lastly, these features are fed into machine learning models such as SVM, KNN, DT, and RF for classification.

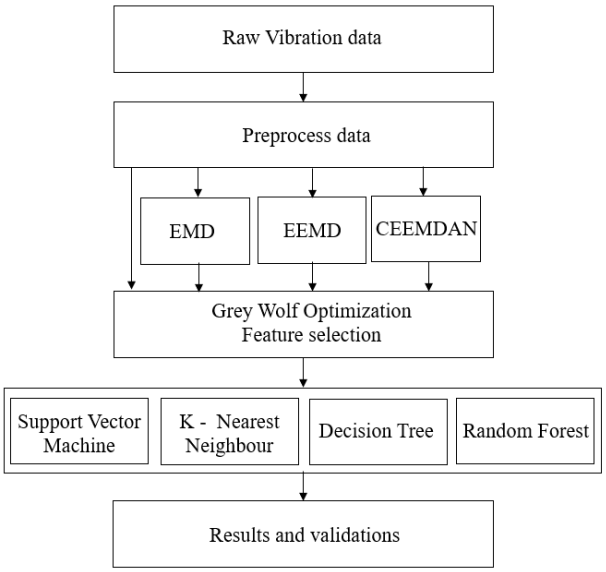

FIGURE 7. Conceptual design structure.

A. DATASET DESCRIPTION

The Case Western Reserve University dataset (CWRU) collects vibration signal data of different bearing health conditions (includes normal state and fault conditions ranging from 7 mils to 40 mils) using a Reliance Electric motor of 2 hp, under the support of a torque encoder, a dyno, and some control electronics [33]. The dataset was used on the CWRU experimental bench,which is shown in Figure 8.

Usually, four major conditions are considered: normal (healthy bearing), inner race fault, outer race fault, and ball fault. Data is collected under different load conditions such as 0 HP, 1 HP, 2 HP, and 3 HP for each of these conditions; as well as at various speeds comprising 1797 RPM, 1772 RPM, 1750 RPM, and 1730 RPM. Furthermore, the dataset should also contain information regarding fault sizes which are typically equal to 0.007 inches, 0.014 inches, and 0.021 inches for inner race, outer race, and all faults. This dataset contains 1000 samples of normal condition for one thousand for each inner race fault ,outer race fault, and ball fault while every type of fault has been further divided into samples at different sizes of these faults. Each sample usually contains a one-second-long time series signal recorded at a high sampling rate such as 12 kHz resulting in 12,000 data points per sample.

Within the context of this paper, 12 kHz sampling frequency data with a motor load of 0hp and motor speed of 1797rpm is selected for performing all experiments. A total of

10 conditions are classified in this study. Respectively, the conditions include normal condition (Normal_0), fault of 7 mils diameters in the inner race condition (IR007_0), fault of 7 mils diameters in the ball condition (B007_0), fault of 7 mils diameters in the outer race condition (OR007_0), fault of 14 mils diameters in the inner race condition (IR014_0), fault of 14 mils diameters in the ball condition (B014_0), fault of 14 mils diameters in the outer race condition (OR014_0), fault of 21 mils diameters in the inner race condition (IR021_0), fault of 21 mils diameters in the ball condition (B021_0), and fault of 21 mils diameters in the outer race condition (OR021_0).

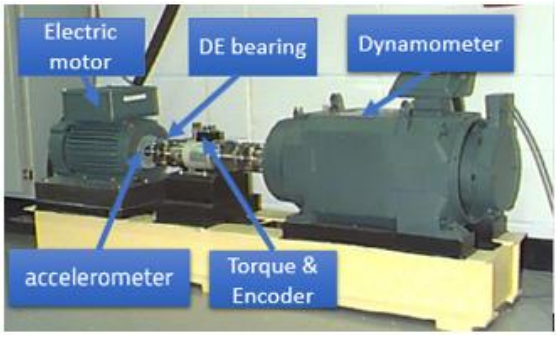

FIGURE 8. **CWRU's experiment setup [33]**

B. DATASET PREPROCESSING

For experimentation, the CWRU datasets of 10 conditions are downloaded, loaded into the MATLAB program, and then checked for missing values. Each of the 10 condition datasets is then split into two sets- one is for training purposes, and one is for testing purposes. The splitting ratio between these is 70/30. From this step, the dataset for testing is set aside, and processed within the same framework as the dataset for training purposes, but separately.

Cleaning data plays an indispensable role in preparing data for the training model, and removing outliers is considered as one of the most efficient cleaning data steps. To make the prepared data representative, outliers, which are elements that stayed far away from the mean (over three times the standard deviation value), are defined and removed. An example of removing outliers of the B014_0_DE_train dataset is presented in the following figures.

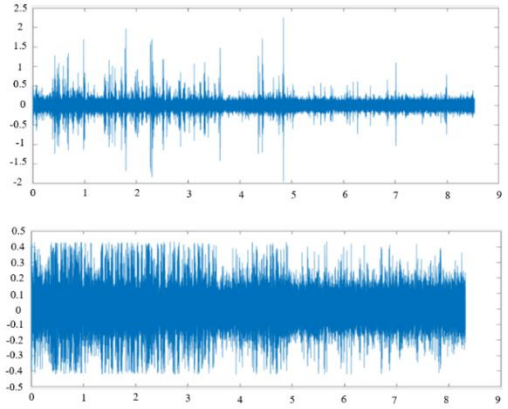

FIGURE 9. **The plot of B014_0_DE_train before and after removing the outlier**

C. SIGNAL DECOMPOSITION

The datasets are then taken as the input for the original EMD, and its modified form – EEMD and CEEMDAN to achieve several IMFs. To choose the suitable IMFs, the correlation between the originally prepared dataset and each IMF is conducted and compared. The IMFs with the highest correlation numbers are selected and stored for further processing.

After that, the datasets are segmented into subsets with the same length so that there are enough statistical features calculated to represent the whole large dataset. Using the segment length of 1024, for training sets, 81 subsets of each condition are extracted. While for testing sets, it is possible to obtain 34 segmentations of data.

D. FEATURE EXTRACTION

Each segment of 1024 data points is calculated to achieve 20 features which are the time-domain features and the frequency-domain features. These features are listed in the section above. After feature extraction, the training feature file has a size of 81 x 20 double, and the testing feature file has a size of 34 x 20 double. The feature files are then concatenated, stored, and imported to only 1 CSV file, with the training's file size at this time is 810 x 21 (twenty features and fault names), and the testing set's file size is 340 x 21 (twenty features and fault names).

After twenty feature types are extracted for each originally prepared dataset, EMD dataset, EEMD dataset, and CEEMDAN dataset the twenty features of the originally prepared datasets are concatenated horizontally with the twenty features of the EMD signal-type datasets to create files of forty features. A similar step is being conducted for EEMD and CEEMDAN signal-type datasets. The combined feature set file size at this time for training is 810 x 41 (twenty features achieved from the raw signal type, twenty features achieved from the decomposed type, and fault names), and for testing is 340 x 41 (twenty features achieved from the raw signal type, twenty features achieved from the decomposed type and fault names).

E. FEATURE SELECTION

The training features files are inputted into the BGWO function for selecting subsets of features with the maximum iterations of 500 and 50 wolves. In this case, k-NN is used as the classification for implementing BGWO. The feature selection process reduces the original feature sets from 20 to 9 types per file and reduces the combined feature sets from 40 to 16 types per file.

Some statistical features selected from the feature selection method include average, root mean square, form, peak to peak, variance, mean square, margin, impulse, and median frequency. Each of these characteristics has its benefits and shortcomings. The Mean provides a central value of a dataset that is important in summarizing the data set's central tendency but can be misleading when data are skewed or contain outliers. The Root Mean Square is useful in highlighting larger values and hence it may be used in

situations where peaks are very important while it can also exaggerate extreme outcomes. The Form shows how spread-out data points are with respect to each other; however, its usefulness depends on the distribution type. Peak-to-peak measures the range of variation, which is straightforward and useful for amplitude variation, yet it ignores the frequency of value occurrence. Variance indicates data variability around the mean, providing a sense of dispersion, but like the mean, it is sensitive to outliers. Mean Square provides a similar benefit to the Root Mean Square but can be redundant if used alongside it. Margin and Impulse help understand extreme values and their occurrences, but they can be skewed by rare extreme outliers. Lastly, Median Frequency is robust against outliers in assessing central tendency, though it might not represent the data's distribution shape effectively. These features collectively provide a comprehensive statistical summary but must be chosen considering the data's nature and analysis objectives.

F. CLASSIFICATION

The features are inputted into four classifiers for training. Then, the models' performances are validated using the features of testing sets. As Machine Learning is non-deterministic and can generate different results in different run times, to be unbiased, ten trials of each method are conducted and then averaged to achieve the presented results below.

The parameter settings of machine learning models are summarized in the Table I:

TABLE I
PARAMETER SETTINGS

ID	Machine Learning Models	Parameters
1	Support Vector Machine (SVM)	kernel = radial random_state = 1 tolerance = 0.001 gamma = c(0.05,0.1,0.5,1,5))
2	K Nearest Neighbor (KNN)	n_neighbors = 3 metric = Eudidean
3	Decision Tree (DT)	min_samples_split = 2 min_samples_leaf = 1 random_state = none max_leaf_nodes = none
4	Random Forest (RF)	n_estimators = 100 random_state = 0

G. PERFORMANCE EVALUATIONS

In the context of this research, three metrics including Accuracy, Specificity, and Positive Predictivity are used for the performance evaluation of the proposed method:

- The Accuracy ratio illustrates the accuracy of the method. This metric's equation is as follows:

$$\text{Accuracy } \frac{TP+TN}{TP+TN+FP+FN} \tag{39}$$

- The Sensitivity ratio indicates the rate of truly positive detection over the truly positive population. The equation for Sensitivity is:

$$\text{Sentivity} = \frac{TP}{TP+FN} \tag{40}$$

- The Specificity ratio indicates the rate of truly negative detection over the truly negative population. The equation for Specificity is:

$$\text{Specificity} = \frac{TN}{FP+TN} \tag{41}$$

With:
True Positive (TP): number of positive- detected cases that did have the conditions.
True Negative (TN): number of negative-detected cases that did not have the conditions.
False Positive (FP): number of detected- positive cases that did not have the conditions.
False Negative (FN) : number of detected- negative cases that did have the conditions.

Apart from the above statistics, the confusion matrices are also conducted as another method for measuring the accuracy of the proposed method. These matrices provide a general view of classification results and have the structure, presented in Table II:

TABLE II
CONFUSION MATRIX STRUCTURE

Target / Predicted	Negative	Positive
Negative	TP	FP
Positive	FN	TP

V. EXPERIMENTAL RESULTS

A. ANALYSIS RESULTS USING VARIOUS MACHINE LEARNING MODELS BEFORE FEATURE SELECTION

Table III shows the summary results of various machine learning models. After all features from the training set were calculated, they were inputted into classifiers for training. Then, as the training processes finished, the models' performances were validated using the features of testing sets. The SVM classifier uses forty features which are achieved by extracting from the raw and CEEMDAN signals obtaining a classification rate of 99.41% accuracy. This result outperforms other methods in accuracy metrics. Furthermore, this method also results in a high average sensitivity of 99.37% and specificity of 99.83%. Apart from SVM, other classifiers also achieve high accuracies (above 88%). With the use of the extracted features achieved from the raw and CEEMDAN signal types, the accuracy metrics results of k-NN, DT, and RF are 95.29%, 93.23%, and 93.82% respectively. The confusion matrices of four classifiers using 40 features achieved from the raw and EEMD signals are presented in Figure 10 - 13 below.

The vital meaning of the model's performance is provided by the confusion matrix for the SVM classifier using 40 extracted features from raw and CEEMDAN signals. From this matrix, we can find out how many true

positives, false positives, true negatives, and false negatives the SVM can predict each class accurately. On the other hand, if there are a lot of misclassifications (either as false positives or false negatives), it suggests that the SVM may have trouble with some classes or should be further tuned to suit the dataset's characteristics. These metrics include sensitivity and specificity as well as accuracy and can be calculated using these results which give a complete picture of how well the classifier works in different respects.

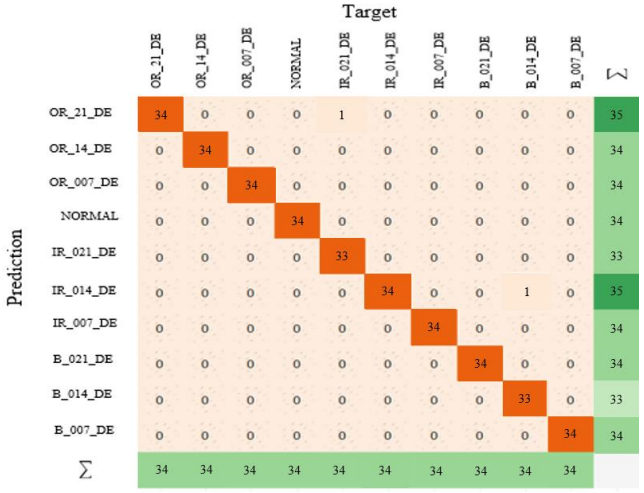

FIGURE 10. Confusion matrices of SVM using 40 extracted features from raw and CEEMDAN signals

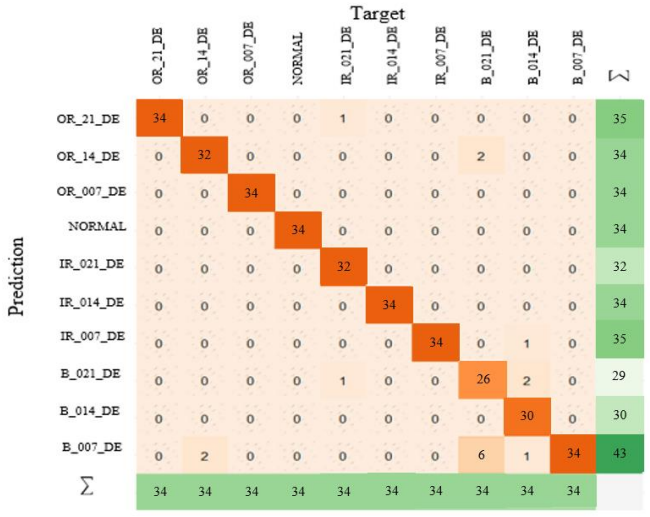

FIGURE 11. Confusion matrices of KNN using 40 extracted features from raw and CEEMDAN signals

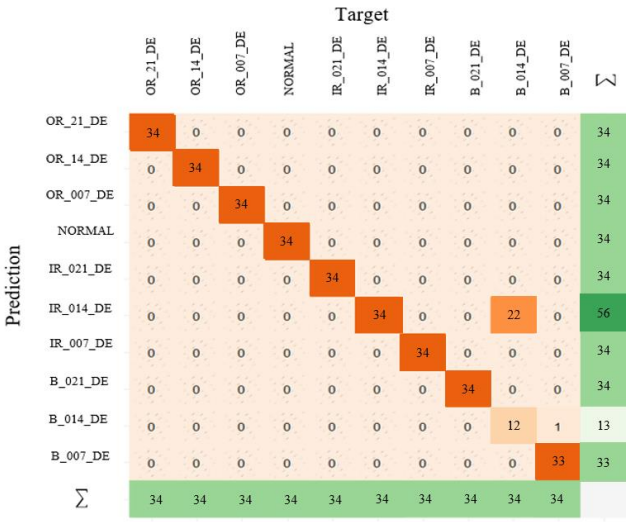

FIGURE 12. Confusion matrices of DT using 40 extracted features from raw and CEEMDAN signals

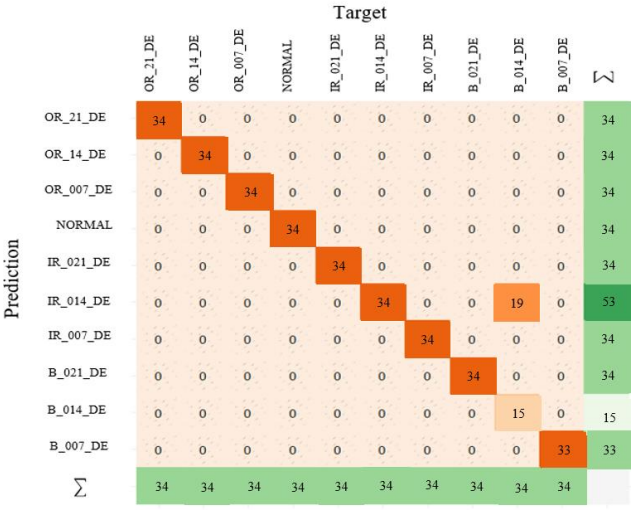

FIGURE 13. Confusion matrices of RF using 40 extracted features from raw and CEEMDAN signals

However, these models still contain disadvantages. Although the SVM classifier using the features extracted from the raw and CEEMDAN signal achieves the highest general accuracy rate, this method takes 105 seconds to compute, which is far longer than the computational time of other methods (k-NN's longest computational time is approximately 0.78 seconds, DT's longest computational time is approximately 0.15 second, RF's longest computational time is approximately 1.38 second). Therefore, it is undeniable that performing SVM is more time-consuming than the other three methods.

TABLE III
RESULTS SUMMARY BEFORE APPLYING FEATURE SELECTION

Classifiers	Signal types	Number of features	Testing accuracy (%)	Sensitivity (%)	Specificity (%)	Classifying time (s)
SVM	Raw	20	98.32	98.29	99.80	52.29
	EMD	20	93.85	93.83	99.28	67.50
	EEMD	20	95.97	95.97	99.55	53.25
	CEEMDAN	20	96.93	96.91	99.42	63.9
	RAW+EMD	20	97.47	97.49	99.71	77.53
	RAW+EEMD	40	98.41	98.38	99.80	87.57
	Raw+ CEEMDAN	**40**	**99.41**	**99.37**	**99.83**	**105.08**
K-NN (k=3)	Raw	20	94.29	94.28	99.33	0.41
	EMD	20	88.09	88.05	98.70	0.29
	EEMD	20	90.94	90.96	99.00	0.26
	CEEMDAN	20	91.85	91.85	99.49	0.31
	Raw+ EMD	40	94.94	94.94	99.40	0.53
	Raw+ EEMD	40	94.44	94.43	99.39	0.65
	Raw+ CEEMDAN	40	95.29	95.38	99.88	0.78
DT	Raw	20	92.35	92.40	99.20	0.08
	EMD	20	89.41	89.40	98.80	0.07
	EEMD	20	89.41	89.40	98.80	0.07
	CEEMDAN	20	90.30	90.30	99.79	0.08
	Raw+ EMD	40	92.35	92.40	99.20	0.11
	Raw+ EEMD	40	92.35	92.40	99.20	0.13
	Raw+ CEEMDAN	40	93.23	93.32	99.69	0.15
RF	Raw	20	92.94	92.90	99.20	0.72
	EMD	20	91.26	91.29	99.04	0.83
	EEMD	20	91.21	91.23	99.03	0.83
	CEEMDAN	20	92.12	92.11	99.52	0.99
	Raw+ EMD	40	92.71	92.68	99.20	1.17
	Raw+ EEMD	40	92.82	92.79	99.20	1.15
	Raw+ CEEMDAN	40	93.82	93.72	99.68	1.38

B. EXPERIMENTAL RESULTS USING FEATURE SELECTION METHODS – GWO

After performing feature selection, the average results of each classifier slightly change and are presented as follows. From the presented results of the twelfth thousand sampling data above (Table IV), after the feature selection process, the computational times of most methods plunge compared with the before feature selection model. Especially, most classifiers perform better. Within ten run trials, the SVM classifier using 18 selected features extracted from the raw and CEEEMDAN signals achieves a classification rate of 99.90% accuracy. This value is much higher compared to the result obtained before feature selection (99.41%). Regarding sensitivity and specificity, the method achieves 99.90% sensitivity and 99.96% specificity. The confusion matrices of four classifiers using 16 features extracted from the raw and CEEMDAN signals are presented in Figure 14-17 below.

The finding of the analysis reveals that Raw+EEMD signal types for machine learning classifiers achieved better testing accuracy, sensitivity, and specificity than other models. EEMD is a method to decompose non-linear and non-stationary signals that can help in exposing patterns. SVMs are known for their efficiency in high-dimensional spaces and a chosen subset of features might be optimal to balance the complexity and the generalization capability of the model, which will reduce overfitting and enhance accuracy. Furthermore, SVM can construct a hyperplane with maximum margin between classes so this could be useful when selected features are Raw +EEMD leading to improved performance.

Apart from achieving a higher accuracy rate, after selecting features, the computational time of models was reduced significantly. Although the gap between the computational time of SVM and other methods was still large, this reduction in time has proved the potentiality and importance of feature selection in reducing computing time and data dimensions. After feature selection, all classifiers still performed well. The four methods gave high-accuracy classification results of above 89%.

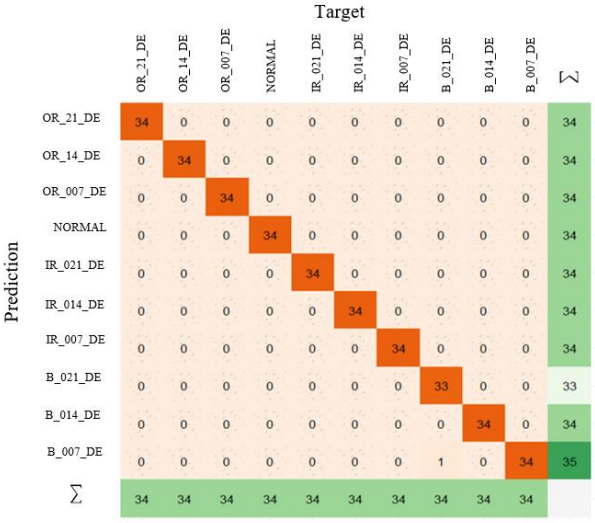

FIGURE 14. Confusion matrices of SVM using 18 selected features extracted from raw and CEEMDAN signal

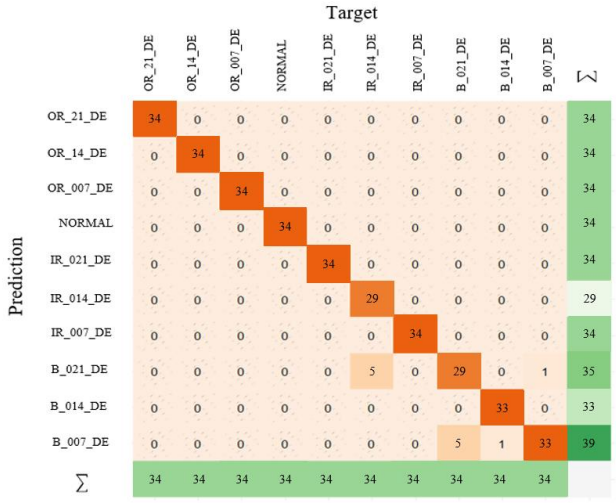

FIGURE 15. Confusion matrices of K-NN using 18 selected features extracted from raw and CEEMDAN signal

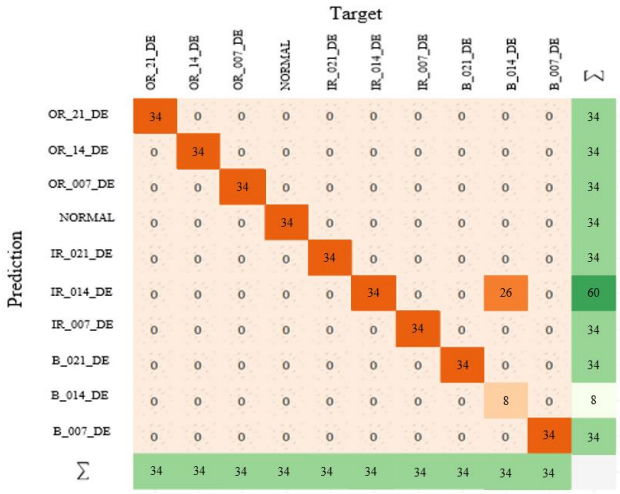

FIGURE 16. Figure 1 Confusion matrices of DT using 18 selected features extracted from raw and CEEMDAN signal

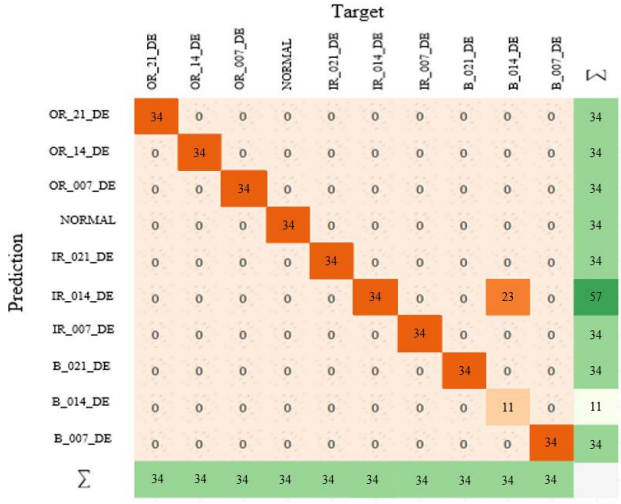

FIGURE 17. Confusion matrices of RF using 18 selected features extracted from raw and CEEMDAN signal

Another necessary instrument to evaluate the effectiveness of the classifier is the SVM confusion matrix with 18 selected features from raw and CEEMDAN signals. This matrix, which has fewer features than the former 40-feature model helps show how feature selection impacts SVM's ability to separate classes. Conversely, a compact feature set can in some cases enhance model generalization by reducing overfitting and removing noise from data. The confusion matrix will indicate whether the SVM maintains or even increases its accuracy as complexity is reduced.

TABLE IV
RESULTS SUMMARY AFTER APPLYING FEATURE SELECTION

Classifiers	Signal types	Number of features	Features	Testing accuracy (%)	Sensitivity (%)	Specificity (%)	Classifying time (s)
SVM	Raw	7	2,3,5,12,14,18,19	99.47	99.46	99.92	27.33
	EMD	7	1,4,13,14,17,18,20	94.32	94.31	97.62	20.39
	EEMD	9	2,3,5,12,13,14,17,18,19	94.85	94.85	99.45	47.73
	CEEMDAN	10	2,3,4,5,12,13,14,17,18,19	95.80	95.81	99.84	57.28
	Raw + EMD	14	2,3,10,12,13,14,18,19,23,26,32,33,37,39	98.59	98.56	99.82	59.79
	Raw + EEMD	16	2,3,4,10,12,13,14,18,20,22,26,30,31,32,37,38	99.85	99.85	99.96	36.08
	Raw + CEEMDAN	**18**	2,3,4,10,12,13,14,18,20,22,26,30,31,32,33,37,38	**99.90**	**99.90**	**99.96**	**43.29**
K-NN (k=3)	Raw	7	2,3,5,12,14,18,19	95.44	95.45	99.50	0.28
	EMD	7	1,4,13,14,17,18,20	96.11	96.14	99.58	0.21
	EEMD	9	2,3,5,12,13,14,17,18,19	89.82	89.82	98.89	0.23
	CEEMDAN	10	1,2,4,5,13,14,17,18,19,20	90.72	90.70	99.88	0.28
	Raw + EMD	14	2,3,10,12,13,14,18,19,23,26,32,33,37,39	90.38	90.39	98.94	0.22
	Raw + EEMD	16	2,3,4,10,12,13,14,18,20,22,26,30,31,32,37,38	95.71	95.72	99.50	0.27
	Raw + CEEMDAN	18	2,3,10,12,13,14,18,19,20,23,26,32,33,37,38,39	96.47	96.67	99.89	0.32
DT	Raw	7	2,3,5,12,14,18,19	91.47	91.50	99.10	0.04
	EMD	7	1,4,13,14,17,18,20	89.12	89.10	98.80	0.04
	EEMD	9	2,3,5,12,13,14,17,18,19	89.71	89.70	98.90	0.05
	CEEMDAN	10	1,3,4,5,13,14,17,18,19,20	90.61	90.61	99.89	0.06
	Raw + EMD	14	2,3,10,12,13,14,18,19,23,26,32,33,37,39	91.47	91.50	99.10	0.05
	Raw + EEMD	16	2,3,4,10,12,13,14,18,20,22,26,30,31,32,37,38	91.47	91.50	99.10	0.07
	Raw + CEEMDAN	18	2,3,4,10,12,13,14,18,20,22,26,30,31,32,33,37,38,39	92.35	92.41	99.59	0.08
RF	Raw	7	2,3,5,12,14,18,19	92.26	92.27	99.15	0.51
	EMD	7	1,4,13,14,17,18,20	95.76	95.78	99.50	0.43
	EEMD	9	2,3,5,12,13,14,17,18,19	95.26	95.27	99.47	0.51
	CEEMDAN	10	1,2,3,5,12,13,14,18,19,20	96.21	96.21	99.65	0.63
	Raw+ EMD	14	2,3,10,12,13,14,18,19,23,26,32,33,37,39	97.94	97.95	99.76	0.59
	Raw+ EEMD	16	2,3,4,10,12,13,14,18,20,22,26,30,31,32,37,38	92.35	92.40	99.20	0.62
	Raw+ CEEMDAN	18	2,3,10,12,13,14,18,19,23,26,31,32,33,37,38,39	93.23	93.32	99.69	0.74

C. PERFORMANCE EVALUATION

1) BEFORE AND AFTER THE FEATURE SELECTION STEP

The 12,000 sampling data classification results before selecting features are compared with the results after selecting features (using 50 wolves and 500 iterations). The comparison in Table V is presented as follows:

TABLE V
RESULTS COMPARISON

Classifiers	Signal	Changing rate of the number of features	Changing the rate of testing accuracy	Changing rate of computational time
SVM	Raw	-65%	1.17%	-47.73%
	EMD	-65%	0.5%	-69.79%
	EEMD	-55%	-1.17%	-10.37%
	CEEMDAN	-50%	-1.17%	-10.36%
	Raw+ EMD	-65%	1.15%	-22.88%
	Raw+ EEMD	-60%	1.46%	-58.80%
	Raw+ CEEMDAN	-55%	0.49%	-58.80%
K-NN	Raw	-65%	1.22%	-31.71%
	EMD	-65%	9.1%	-27.59%
	EEMD	-55%	-1.23%	-11.54%
	CEEMDAN	-50%	-1.23%	-9.68%
	Raw+ EMD	-65%	-4.8%	-58.49%
	Raw+ EEMD	-60%	1.34%	-58.46%
	Raw+ CEEMDAN	-55%	1.24%	-58.97%
DT	Raw	-65%	-0.95%	-50%
	EMD	-65%	-0.32%	-42.86%
	EEMD	-55%	0.34%	-28.57%
	CEEMDAN	-50%	0.34%	-25%
	Raw+ EMD	-65%	-0.95%	-54.55%
	Raw+ EEMD	-60%	-0.95%	-46.15%
	Raw+ CEEMDAN	-55%	-0.94%	-46.67%
RF	Raw	-65%	-0.73%	-29.17%
	EMD	-65%	4.93%	-48.19%
	EEMD	-55%	4.44%	-38.55%
	CEEMDAN	-50%	4.44%	-36.36%
	Raw+ EMD	-65%	5.64%	-49.57%
	Raw+ EEMD	-60%	-0.51%	-46.09%
	Raw+ CEEMDAN	-55%	-0.63%	-46.38%

For the proposed method, this method was influenced positively after being applied BGWO feature selection. There was an increase of 0.49% in accuracy and a decrease of 58.80% (more than half) of the computational time of the classification.

It was clear that the other achieved results were also positive. Regarding accuracy rates, by using sets of selected features, half of the methods (10 over 20) maintained or increased their accuracies. In particular, the highest level of increase in accuracy was 9.1% (k-NN classifier using EMD signal's features), and the highest decrease in accuracy was 4.8% (k-NN classifier using combined raw and EMD's features). Apart from the mentioned k-NN method which has a 4.8% decrease in accuracy, other methods showed a slight reduction in accuracy rate (smaller than 1%).

Regarding computational time, the computational times of the majority of methods plunged at the highest rate decreasing 69.79% compared with the before feature selection model.

From the above table, it was clear that the feature selection process was potential. It brought a myriad of benefits to models when it was applied to classification problems. Using feature selection could result in reducing computational expenses, and computational time, and increasing accuracy rates (in cases).

2) THE PROPOSED METHOD AND RECENT RESEARCHERS

Table VI compares the performance of different research methods in terms of accuracy rates and computation times, as reported in various studies. This comparison highlights the effectiveness of the methods across different research works. From the table below, regarding the accuracy rate of bearing classifications using the 12,000 samplings CWRU data, the proposed method (SVM using extracted features from raw and CEEMDAN signal - 99.90%) performed slightly better than the research method in [10], [12] and [13]. Besides, the proposed method still needs further improvement since its accuracy rate was the same as the research [14].

Regarding computational time, the three previous works [10], [12], [14] did not disclose these statistics; therefore, it was impossible to compare. For research [13], the proposed method (43.29 seconds) performed slightly lower than the CNN and SVM combined method.

TABLE VI
RESULTS COMPARISON

Research	Dataset	Methods	Accuracy rate (%)	Computational time (s)
Afrasiabi et al., 2019 [10]	CWRU	accelerated CNN	97.2%	N/A
Xu et al., 2019 [12]	CWRU	CNN and RF	99.73%	N/A
Li, Yang, et al., 2019 [1]	CWRU	DSLS-SVM	99.90%	N/A
Yuan et al., 2020 [13]	CWRU and MFPT	CNN and SVM	98.75%	35.82

From the achieved results, it was clear that the proposed method worked effectively. Within the SVM method, classifying bearing failures using the features extracted from raw (preprocess) data and EEMD decomposed data gave a high accuracy rate. Under the feature selection process, most classifiers performed better than before. Therefore, it was logical to suggest performing BGWO feature selection before inputting the features set to classifiers

D. ANALYSIS

1) GENERAL RESULTS ANALYSIS

From the presented results of the twelfth thousand sampling data above, after the feature selection process, the proposed method – SVM diagnosing using the raw and EEMD signal's combined features – was still the most

effective method in terms of accuracy rate. By using 16 feature types, this method achieved an accuracy rate of 99.90%, which was higher than the result that it had achieved by using the original 40 feature types (99.41%). Regarding the sensitivity and specificity, the method achieved 99.90% in sensitivity and 99.96% in specificity.

Apart from achieving a higher accuracy rate, after selecting features, the computational time of the proposed method (and of SVM methods in general) reduced significantly. Although the gap between the computational time of SVM and other methods was still large, this reduction in time has proved the potentiality and importance of feature selection in reducing computing time and data dimensions. After feature selection, all classifiers still performed well. The four methods gave high accuracy classification results of above 89% (The reduction in accuracy metrics was not noticeable).

Moreover, with the SVM classifier, using the combined features from raw and CEEMDAN signals, the analysis was conducted by changing the population of wolves. The classification results were achieved using features selected from a 500 constant number of iterations and 5 different values of wolves' population - 10 wolves, 12 wolves, 30 wolves, 35 wolves, and 50 wolves.

The bar chart below displays how accurate three wolf populations are in terms of Testing Accuracy, Sensitivity, and Specificity. In general, the Testing Accuracy and Specificity increase with population size. That means a larger population size of wolves may be more dependable and consistent for these measures. On the other hand, Sensitivity does not seem to change much with any different group sizes implying that it is not strongly determined by population size. The chart therefore clearly distinguishes between different populations and indicates possible trends among these factors that are essential in various investigations.

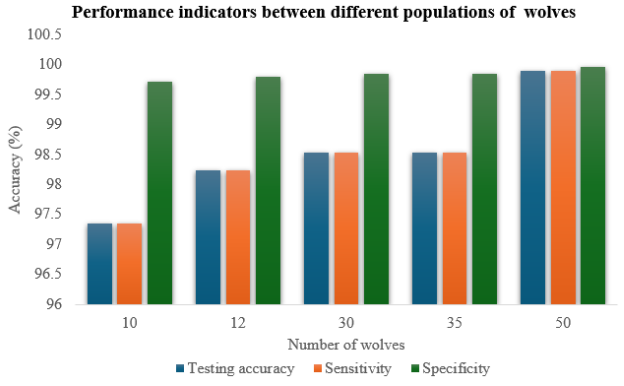

FIGURE 18. Changes in performance indicators between different populations of wolves

2) SENSITIVITY ANALYSIS

To remove incomplete noisy clumsy and inconsistency in the dataset, a pre-processing step is very necessary. There is a wide range of sample selection methods for training data sets during cross-validation among others. The k-fold cross-validation method breaks actual samples into k equal-sized subsamples [34], [35]. For testing the classification model each subsample contains validation data, repeated k times. Over repeated random sub-sampling, this method has an advantage as at least once, training and validation are used for each of the validations. Here k is the unfixed parameter that will be chosen by the user. K-fold cross-validation into k parts using one part as test data and the rest (k-1) as train data split documents.

In this study, we adopted 10-fold cross-validation to boost up training effectiveness of the models. The network's input data is divided into 10 subgroups through the application of the 10-fold algorithm. The other subgroup acts as test data whereas nine subgroups serve as training data for every round of models' training. Because most of the information is utilized to teach these models ten times instead of when they are trained over less number iterations therefore reduced bias is achieved. Figure 19 shows a common k-fold design. The model in this research is trained for 10 iterations after the input data is divided into k =10.

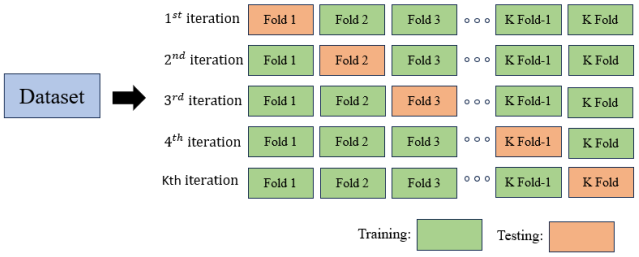

FIGURE 19. K-fold cross-validation process

The bar chart given below shows the performance of four different machine learning models SVM, KNN, DT, and RF in terms of accuracy, specificity and sensitivity based on a mean and standard deviation of 10-fold cross-validated approach indicating consistency and reliability of the models. The SVM models show the highest performance across all metrics with almost 100% accuracy and high specificity as well as sensitivity hence a sign that positive and negative classifications are unaffected. In contrast, KNN is not effective at all while DT falls much behind as far as these three measures are concerned. These error bars explain how each model is stable or reliable by illustrating respective graphical representations due to variances across different cross-validation folds.

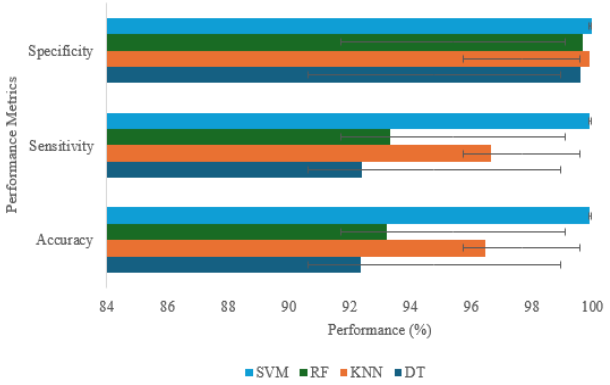

FIGURE 20. Performance metrics with K-fold (K=10) cross-validation obtained from SVM, RF, KNN, and DT

VI. CONCLUSION

Machine Learning has been used widely in research for failure detection. Through these studies, positive results have been achieved. However, the research area encountered practical problems as its computational time is not competitive and the computational expense is high. The main reasons behind these challenges are that the dataset needed for performing predictions is often large and has the characteristics of high dimensionality. As a small attempt to solve these problems, this paper uses Machine Learning to predict the conditions of the bearing. To achieve the result, we apply EMD and EEMD decomposition on the signal, then combine features extracted from both preprocessed signals and decomposed signals. After that, informative features are selected by BGWO, and the selected features are inputted into classifiers. With a time running of 36.08 seconds, by analyzing 16 selected features that are extracted from the raw and EEMD signals, the SVM classifier achieves the results of 99.85% accuracy. These statistics show that through this method, the dimensions of data are reduced effectively, making the computational expense and time more competitive. However, it still needs further improvement. The promising future direction for this research is to expand it to other metaheuristics [36], [37], [38], [39], [40], [41], [42].

To further enhance the performance of machine learning in failure detection, integrating advanced methods like Deep Stacking Least Squares (DSLs)[14], Extreme Gradient Boosting (XGBoost) [43], and kernel-based Support Vector Machines (SVMs) [44] is promising. These techniques can improve computational efficiency and accuracy, making machine learning more practical and effective for real-world applications in failure detection. Moreover, we aim to deepen our study in predictive maintenance by considering additional information such as Remaining Useful Life (RUL) related data. Leveraging RUL data will enable us to finely assess the predictive strength of our selected features across varied working conditions, thus boosting the accuracy of failure predictions. In the next steps, we will deeply test and fine-tune our AI models to better deal with the complex data of RUL. By doing so, our goal is to develop more robust predictive maintenance methods that are not just accurate but also effective in real-world applications.

VII. ACKNOWLEDGEMENT

This work has been conducted through the RMIT Vietnam Research Grants-2024, administered by the Office for Research & Innovation.

REFERENCES

[1] M. Syafrudin, G. Alfian, N. L. Fitriyani, and J. Rhee, "Performance Analysis of IoT-Based Sensor, Big Data Processing, and Machine Learning Model for Real-Time Monitoring System in Automotive Manufacturing," *Sensors*, vol. 18, no. 9, Art. no. 9, Sep. 2018, doi: 10.3390/s18092946.

[2] F. Arellano-Espitia, M. Delgado-Prieto, V. Martinez-Viol, J. J. Saucedo-Dorantes, and R. A. Osornio-Rios, "Deep-Learning-Based Methodology for Fault Diagnosis in Electromechanical Systems," *Sensors*, vol. 20, no. 14, Art. no. 14, Jan. 2020, doi: 10.3390/s20143949.

[3] R. Marino, C. Wisultschew, A. Otero, J. M. Lanza-Gutierrez, J. Portilla, and E. de la Torre, "A Machine-Learning-Based Distributed System for Fault Diagnosis With Scalable Detection Quality in Industrial IoT," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4339–4352, Mar. 2021, doi: 10.1109/JIOT.2020.3026211.

[4] X. Liu, L. Xia, J. Shi, L. Zhang, L. Bai, and S. Wang, "A Fault Diagnosis Method of Rolling Bearing Based on Improved Recurrence Plot and Convolutional Neural Network," *IEEE Sens. J.*, vol. 23, no. 10, pp. 10767–10775, May 2023, doi: 10.1109/JSEN.2023.3265409.

[5] H. Tao, J. Qiu, Y. Chen, V. Stojanovic, and L. Cheng, "Unsupervised cross-domain rolling bearing fault diagnosis based on time-frequency information fusion," *J. Frankl. Inst.*, vol. 360, no. 2, pp. 1454–1477, Jan. 2023, doi: 10.1016/j.jfranklin.2022.11.004.

[6] A. Alkaya and İ. Eker, "Variance sensitive adaptive threshold-based PCA method for fault detection with experimental application," *ISA Trans.*, vol. 50, no. 2, pp. 287–302, Apr. 2011, doi: 10.1016/j.isatra.2010.12.004.

[7] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "KNN Model-Based Approach in Classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, R. Meersman, Z. Tari, and D. C. Schmidt, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2003, pp. 986–996. doi: 10.1007/978-3-540-39964-3_62.

[8] G. Wu, X. Ji, G. Yang, Y. Jia, and C. Cao, "Signal-to-Image: Rolling Bearing Fault Diagnosis Using ResNet Family Deep-Learning Models," *Processes*, vol. 11, no. 5, Art. no. 5, May 2023, doi: 10.3390/pr11051527.

[9] F. Dong *et al.*, "Triboelectric nanogenerator-embedded intelligent bearing with rolling ball defect diagnosis via signal decomposition and automated machine learning," *Nano Energy*, vol. 119, p. 109072, Jan. 2024, doi: 10.1016/j.nanoen.2023.109072.

[10] S. Afrasiabi, M. Afrasiabi, B. Parang, and M. Mohammadi, "Real-Time Bearing Fault Diagnosis of Induction Motors with Accelerated Deep Learning Approach," in *2019 10th International Power Electronics, Drive Systems and Technologies Conference (PEDSTC)*, Feb. 2019, pp. 155–159. doi: 10.1109/PEDSTC.2019.8697244.

[11] X. Li, W. Zhang, and Q. Ding, "Understanding and improving deep learning-based rolling bearing fault diagnosis with attention mechanism," *Signal Process.*, vol. 161, pp. 136–154, Aug. 2019, doi: 10.1016/j.sigpro.2019.03.019.

[12] G. Xu, M. Liu, Z. Jiang, D. Söffker, and W. Shen, "Bearing Fault Diagnosis Method Based on Deep Convolutional Neural Network and Random Forest Ensemble Learning," *Sensors*, vol. 19, no. 5, Art. no. 5, Jan. 2019, doi: 10.3390/s19051088.

[13] L. Yuan, D. Lian, X. Kang, Y. Chen, and K. Zhai, "Rolling Bearing Fault Diagnosis Based on Convolutional Neural Network and Support Vector Machine," *IEEE Access*, vol. 8, pp. 137395–137406, 2020, doi: 10.1109/ACCESS.2020.3012053.

[14] X. Li, Y. Yang, H. Pan, J. Cheng, and J. Cheng, "A novel deep stacking least squares support vector machine for rolling bearing fault diagnosis," *Comput. Ind.*, vol. 110, pp. 36–47, Sep. 2019, doi: 10.1016/j.compind.2019.05.005.

[15] A. Ucar, M. Karakose, and N. Kırımça, "Artificial Intelligence for Predictive Maintenance Applications: Key Components, Trustworthiness, and Future Trends," *Appl. Sci.*, vol. 14, no. 2, Art. no. 2, Jan. 2024, doi: 10.3390/app14020898.

[16] S. Kaparthi and D. Bumblauskas, "Designing predictive maintenance systems using decision tree-based machine learning techniques," *Int. J. Qual. Reliab. Manag.*, vol. 37, no. 4, pp. 659–686, Jan. 2020, doi: 10.1108/IJQRM-04-2019-0131.

[17] B. Qi, Y. Li, W. Yao, and Z. Li, "Application of EMD Combined with Deep Learning and Knowledge Graph in Bearing Fault," *J. Signal Process. Syst.*, vol. 95, no. 8, pp. 935–954, Aug. 2023, doi: 10.1007/s11265-023-01845-z.

[18] F. Dao, Y. Zeng, and J. Qian, "A novel denoising method of the hydro-turbine runner for fault signal based on WT-EEMD," *Measurement*, vol. 219, p. 113306, Sep. 2023, doi: 10.1016/j.measurement.2023.113306.

[19] Y. Hu, Y. Ouyang, Z. Wang, H. Yu, and L. Liu, "Vibration signal denoising method based on CEEMDAN and its application in brake disc unbalance detection," *Mech. Syst. Signal Process.*, vol. 187, p. 109972, Mar. 2023, doi: 10.1016/j.ymssp.2022.109972.

[20] R. Abdelkader, A. Kaddour, A. Bendiabdellah, and Z. Derouiche, "Rolling Bearing Fault Diagnosis Based on an Improved Denoising Method Using the Complete Ensemble Empirical Mode Decomposition and the Optimized Thresholding Operation," *IEEE Sens. J.*, vol. 18, no. 17, pp. 7166–7172, Sep. 2018, doi: 10.1109/JSEN.2018.2853136.

[21] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 01, no. 01, pp. 1–41, Jan. 2009, doi: 10.1142/S1793536909000047.

[22] Y. Li, Y. Li, X. Chen, J. Yu, H. Yang, and L. Wang, "A New Underwater Acoustic Signal Denoising Technique Based on CEEMDAN, Mutual Information, Permutation Entropy, and Wavelet Threshold Denoising," *Entropy*, vol. 20, no. 8, Art. no. 8, Aug. 2018, doi: 10.3390/e20080563.

[23] Y. Yang, P. Fu, and Y. He, "Bearing Fault Automatic Classification Based on Deep Learning," *IEEE Access*, vol. 6, pp. 71540–71554, 2018, doi: 10.1109/ACCESS.2018.2880990.

[24] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.

[25] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[26] T. T. M. Huynh, T. M. Le, L. T. That, L. V. Tran, and S. V. T. Dao, "A Two-Stage Feature Selection Approach for Fruit Recognition Using Camera Images With Various Machine Learning Classifiers," *IEEE Access*, vol. 10, pp. 132260–132270, 2022, doi: 10.1109/ACCESS.2022.3227712.

[27] L. V. Tran, H. M. Tran, T. M. Le, T. T. M. Huynh, H. T. Tran, and S. V. T. Dao, "Application of Machine Learning in Epileptic Seizure Detection," *Diagnostics*, vol. 12, no. 11, Art. no. 11, Nov. 2022, doi: 10.3390/diagnostics12112879.

[28] S. V. T. Dao, Z. Yu, L. V. Tran, P. N. K. Phan, T. T. M. Huynh, and T. M. Le, "An Analysis of Vocal Features for Parkinson's Disease Classification Using Evolutionary Algorithms," *Diagnostics*, vol. 12, no. 8, Art. no. 8, Aug. 2022, doi: 10.3390/diagnostics12081980.

[29] T. Le, T. Pham, and S. Dao, "Using Machine Learning to Predict the Defaults of Credit Card Clients," in *Fintech with Artificial Intelligence, Big Data, and Blockchain*, P. M. S. Choi and S. H.

Huang, Eds., in Blockchain Technologies. , Singapore: Springer, 2021, pp. 133–152. doi: 10.1007/978-981-33-6137-9_4.

[30] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[31] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016, doi: 10.1016/j.neucom.2015.06.083.

[32] J. Too, A. R. Abdullah, N. Mohd Saad, N. Mohd Ali, and W. Tee, "A New Competitive Binary Grey Wolf Optimizer to Solve the Feature Selection Problem in EMG Signals Classification," *Computers*, vol. 7, no. 4, Art. no. 4, Dec. 2018, doi: 10.3390/computers7040058.

[33] "Bearing Data Center | Case School of Engineering | Case Western Reserve University," Case School of Engineering. Accessed: Dec. 22, 2023. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[34] K. Pal and Biraj. V. Patel, "Data Classification with k-fold Cross Validation and Holdout Accuracy Estimation Methods with 5 Different Machine Learning Techniques," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, Mar. 2020, pp. 83–87. doi: 10.1109/ICCMC48092.2020.ICCMC-00016.

[35] S. M. Malakouti, M. B. Menhaj, and A. A. Suratgar, "The usage of 10-fold cross-validation and grid search to enhance ML methods performance in solar farm power generation prediction," *Clean. Eng. Technol.*, vol. 15, p. 100664, Aug. 2023, doi: 10.1016/j.clet.2023.100664.

[36] T. Pham and S. Dao, "6 - Plant leaf disease classification based on feature selection and deep neural network," in *Handbook of Deep Learning in Biomedical Engineering*, V. E. Balas, B. K. Mishra, and R. Kumar, Eds., Academic Press, 2021, pp. 155–189. doi: 10.1016/B978-0-12-823014-5.00010-7.

[37] T. N. Pham, L. V. Tran, and S. V. T. Dao, "Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection," *IEEE Access*, vol. 8, pp. 189960–189973, 2020, doi: 10.1109/ACCESS.2020.3031914.

[38] T. M. Le, L. V. Tran, and S. V. T. Dao, "A Feature Selection Approach for Fall Detection Using Various Machine Learning Classifiers," *IEEE Access*, vol. 9, pp. 115895–115908, 2021, doi: 10.1109/ACCESS.2021.3105581.

[39] T. M. Le, T. N. Pham, and S. V. T. Dao, "A Novel Wrapper-Based Feature Selection for Heart Failure Prediction Using an Adaptive Particle Swarm Grey Wolf Optimization," in *Enhanced Telemedicine and e-Health: Advanced IoT Enabled Soft Computing Framework*, G. Marques, A. Kumar Bhoi, I. de la Torre Díez, and B. Garcia-Zapirain, Eds., in Studies in Fuzziness and Soft Computing. , Cham: Springer International Publishing, 2021, pp. 315–336. doi: 10.1007/978-3-030-70111-6_15.

[40] M. T. Le, M. Thanh Vo, L. Mai, and S. V. T. Dao, "Predicting heart failure using deep neural network," in *2020 International Conference on Advanced Technologies for Communications (ATC)*, Oct. 2020, pp. 221–225. doi: 10.1109/ATC50776.2020.9255445.

[41] M. T. Le, M. T. Vo, N. T. Pham, and S. V. T. Dao, "Predicting heart failure using a wrapper-based feature selection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 3, Art. no. 3, Mar. 2021, doi: 10.11591/ijeecs.v21.i3.pp1530-1539.

[42] T. M. Le, T. M. Vo, T. N. Pham, and S. V. T. Dao, "A Novel Wrapper–Based Feature Selection for Early Diabetes Prediction Enhanced With a Metaheuristic," *IEEE Access*, vol. 9, pp. 7869–7884, 2021, doi: 10.1109/ACCESS.2020.3047942.

[43] L. TonThat, V. T. Son Dao, H. T. Minh Tri, and M. T. Le, "A Feature Subset Selection Approach For Predicting Smoking Behaviours," in *2023 IEEE Statistical Signal Processing Workshop (SSP)*, Jul. 2023, pp. 145–149. doi: 10.1109/SSP53291.2023.10208015.

[44] N. N. Y, T. V. Ly, and D. V. T. Son, "Churn prediction in telecommunication industry using kernel Support Vector Machines," *PLOS ONE*, vol. 17, no. 5, p. e0267935, thg 5 2022, doi: 10.1371/journal.pone.0267935.

QUYNH N.X PHAN is currently a master's student in Statistics and Operational Research at the University of Edinburgh School of Mathematics. She received a Bachelor of Engineering in Logistics and Supply Chain Management from the International University Vietnam National University Ho Chi Minh City. She is interested in Machine Learning, Simulation, and Operational Research Optimization.

TUAN M. LE graduated from the International University, Vietnam National University, y, Vietnam with a bachelor's degree in electrical engineering, in 2018 and M.SC in Electronics Engineering, in 2022. He is currently a Laboratory technician. His research concentrates on Embedded

systems and Internet of Things (IoT) and machine learning (ML).

HIEU M. TRAN received a B.Eng. degree in automation and control engineering from International University, Vietnam National University Ho Chi Minh City, Vietnam, in 2021, where he is currently pursuing the M.Eng. degree in electrical engineering. He is also a Teaching Assistant at the International University, Vietnam National University Ho Chi Minh City. His research interests include artificial intelligence on the edge and deep learning.

LY VAN TRAN received the B.Eng. degree in mechanical engineering in 1996, and the M.Eng. degree in manufacturing systems engineering in 2000. From 2000 to 2019, he was in various industries with a wide range of roles from operations to senior management. His research interests include operation research and product development.

SON V.T DAO received the B. Eng in Aeronautical Engineering, M. Sc. In Manufacturing Systems and Technology, Ph.D. in Sensors Technology, in 2004, 2005, and 2010 respectively. From 2006 to 2008, he was with Hylax Ltd (Singapore) designing high power laser systems. From 2010 to 2012, he was a Postdoctoral fellow at Kindai University (Japan). From 2012 to 2015, he worked as a Senior Scientist at Ritsumeikan University (Japan). His research field covers the development of advanced imaging sensors and systems and applied artificial intelligence.