A FAULT DIAGNOSIS APPROACH FOR ROTATING MACHINERY BASED ON A COMBINATION OF MACHINE LEARNING AND PARTICLE SWARM OPTIMIZATION ALGORITHM

PHUOC-BAO-DUY NGUYEN[1]

[1]Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT)
268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
[1]Vietnam National University Ho Chi Minh City, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Abstract:

The early identification and classification of common faults in rotating drive systems, such as bearings and gears, are crucial for early fault detection and maintenance, preventing severe damage to the system. This paper introduces a cost-effective method to address this issue, based on extracting time-domain features from vibration signals obtained from sensors, combined with the Particle Swarm Optimization and machine learning algorithms to reduce the number of features required for classification. The approach aims to minimize computational complexity while maintaining exceptionally high accuracy. The proposed method is validated on three datasets, demonstrating its effectiveness and reliability.

Keywords:

fault diagnosis, machine learning, particle swarm optimization, time-domain features

1. Introduction

Rotating machinery plays an important role in industrial fields [1][2][3]. During operation, issues such as gear chipping or damage to the inner or outer surfaces of the bearings can easily occur, leading to excessive system vibration and reduced accuracy. If these damages are not detected and addressed early, they can cause severe harm to the machinery, resulting in high repair costs and potential risks to users. Many solutions have been studied for the early detection of these damages, mostly through time series signal from vibration sensors [4][5][6]. Wang et al [5] used Convolution Neural Network (CNN) with Efficient Channel Attention (ECA), testing on Case Western Reserve Univerity (CWRU) bearing dataset and Southeast University (SEU) gearbox datasets, achieved excellent classification results under noise-free conditions, but the qualities decreased when the signals were augmented with Gaussian white noise. Shao et al [6] proposed a Deep CNN combined with a pre-trained model, investigated CWRU dataset and gearbox dataset obtained form the Drivetrain Dynamic Simulator, although good results were also achieved, but the model's performance was worse when it was trained from scratch instead of from the pre-trained model. Tran et al [7] used a pre-trained Deep Neural Network to extract high level features, then a traditional machine learning classifiers like Support Vector Machine (SVM), k-Nearest Neighbor (kNN) or Random Forest (RF) Classifier was used these features for further classification. This approach achieved impressive results with Paderborn University (PU) bearing dataset, CWRU dataset and Machinery Failure Prevention Technology (MFPT) vibration dataset. Raj et al [8] proposed a 2D-CNN model combined with principal component analysis (PCA), achieved above 98% accuracy on CWRU dataset. The aforementioned methods, although achieving great results but required high computational workload.

There are also many studies on conventional algorithms for this problem. Shen et al [9] used CWRU to simulation the improved grey wolf optimizer (IGWO) combined with SVM, and get the accuracy of 98.75%. Wang and Du [10] tested an fault diagnosis approach northern goshawk optimization (NGO) - SVM and modified hierarchical fluctuation dispersion entropy of tan-sigmoid mapping (MHFDE_TANSIG) on the UConn and SEU gear dataset, obtained an average accuracy of 97.25%.

This study presents a method that combines time series features extraction from sensor signals over time with one of two widely used machine learning algorithms, Support Vector Machine and Random Forest Classifier, integrates the Particle Swarm Optimization (PSO) algorithm to reduce the number of features required for classification, aiming to decrease computational cost. In order to evaluate the effectiveness of the proposed methods, three datasets were investigated: CWRU bearing datasets [11], University of Connecticut (UOC) gear fault datasets [12] and SEU gearbox datasets [13], achieved excellent classification results with accuracy above 99%. The proposed method also demonstrates robustness in the case where the signal was augmented by Gaussian white noise.

2. Proposed method

2.1 Time series features extraction

To facilitate classification and fault identification, the time-domain signals obtained from the sensors were first divided into samples consisting of N points. Then, the time-domain features of these samples were extracted. In addition to common features such as maximum, minimum, root mean square, mean, standard deviation, variance, energy, and data range, the following features also need to be used:

- Skewness measures the extent of the asymmetry of the distribution curve, compared with the normal distribution, and it can be calculated in many ways. The simplest formula is Pearson's second skewness one, also known as median skewness. Kurtosis characterizes how peaked or flatten of a normal-like distribution, larger kurtosis means sharper peak of the distribution curve. Crest factor is defined as the ratio of the peak value to the rms value, it illustrates how extreme of a waveform's peak. Crest factor 1 shows that there is no peak, such as square wave, while higher crest factor indicates higher peak. Form factor is the ratio of the rms value to the arithmetic mean. A pure sinusoidal waveform has a form factor of 1.11.

- Quartile and Inter-quartile range: the first quartile Q_1, the second quartile Q_2 (also known as the median) and the third quartile Q_3 are the points where 25%, 50% and 75% of data are below these points, respectively. The inter-quartile range (IQR) is the distance between Q_3 and Q_1.

- Autocorrelation: a lag-k autocorrelation r_k is a correlation between a time series and a time-shifted version by k time steps of itself:

$$r_k = \frac{\sum_{i=1}^{N-k}(x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{1}$$

- Mean difference (MD) of a time series indicates the average change between consecutive values and provides insights into the trend and smoothness of the series. Second order difference mean (SODM) of a time series measures the average change of the MD, and demonstrates the acceleration trends in the data. Mean absolute deviation (MAD) measures the average absolute deviation of each value from the mean of time series. MAD is a metric for assessing variability of time series, but it is less sensitive to extremely values compared to standard deviation.

- Coefficient of variation (CV) measures the standard deviation as a percentage of the mean, a higher CV shows a greater relative variability.

- Partial Autocorrelation Function (PACF) at lag k indicates the direct association between a time series and its past values at lag k, excluding the influence of intermediate lags. It can be determined by Yule-Walker equations or Durbin-Levinson algorithm.

- Augmented Dickey-Fuller (ADF) test is used to check if a time series is stationary, it has two key values: ADF statistic, which determines how strongly a time series is stationary, and p-value, which is a probability obtained by comparing the ADF statistic to Dickey-Fuller tables.

- Hurst exponent measures the long term memory of time series, its value ranges from 0 to 1, with the value of 0.5 indicates that the series is purely stochastic. There are many methods for estimating the Hurst exponent [14], in this paper, the classical rescaled range method is used.

- The Dominant frequency refers to the frequency component with the highest amplitude of a time series, it represents the most significant oscillation present in the data. Dominant frequency can be determined by taking Fast Fourier Transform of the time series, then identify the frequency with the highest amplitude. Spectral density at peak frequency, $P(f_{\text{peak}})$, demonstrates the power of a time series at its most dominant frequency in the frequency domain.

- Seasonality strength (SS) is typically a measure of how strong the periodic components are in a time series compared to its overall variability. It illustrates the importance of seasonal pattern relative to other source of variation such as trends or noise. Entropy of seasonality (H) shows the uncertainty in the seasonal pattern of a time series, also knows as Shannon entropy. In order to calculate H, the power spectrum $P(f_i)$ are computed first, then H can be obtained as:

$$H = -\sum p_i \log_2 p_i \tag{2}$$

  where $p_i = P(f_i)/\sum P(f_i)$ is the probability of each frequency based on its power.

- Number of outliers quantifies how many data points in a sample significantly deviate from the expected pattern. Any point x_i with a $|Z_i| > 3$ is considered an outlier, where Z is the z-score of that data point.

All of above features were used in this study to support the fault classification problem, serving as inputs of machine learning algorithm after applying the Standard Scaler (also known as z-score normalization):

$$X' = \frac{X - \mu_X}{\sigma_X} \tag{3}$$

where X is the original value of the feature, X' is the standardized value, μ_X and σ_X are the mean and the standard deviation of the feature, respectively.

2.2 Support Vector Machine

Support Vector Machine (SVM) is a robust supervised learning technique designed for both classification and regression problems [15]. It works by identifying an optimal hyperplane that best separates data points into different categories. This hyperplane is constructed to maximize the separation margin, which refers to the distance between the nearest data points—known as support vectors—of each class and the decision boundary. The distance between the hyperplane and the nearest data points is called the margin.

When the data is linearly separable, the hyperplane can be described as:

$$w \cdot x + b = 0 \tag{4}$$

where w represents the weight vector, x denotes the input feature vector, and b corresponds the bias term. In the case of nonlinear problem, SVM utilize a technique called the kernel trick to transform input data into a higher-dimensional space, enabling linear separability. Commonly used kernel functions include the polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel.

SVM is originally a binary classifier, however, for multi-class classification, SVM can be extended using One-versus-One or One-versus-All approaches. The former method involves training a separate SVM classifier for each pair of classes, for a dataset with n classes, $n(n-1)/2$ binary classifiers are required, then during prediction, each classifier votes for a class, and the class with the most votes is selected. The latter method, in the other hand, trains one SVM classifier per class, treating it as positive and all others as negative, then the classifier with the highest confident score determines the class.

2.3 Random Forest Classifier

The Random Forest (RF) Classifier is an ensemble learning algorithm based on decision trees, designed to improve classification accuracy while reducing overfitting. It was introduced by Leo Breiman [16] as an extension of the CART (Classification and Regression Trees) algorithm, which forms the foundation of individual trees in the Random Forest. The CART model splits data at each node using a criterion such as gini impurity or entropy, ensuring that the resulting subgroups are as pure as possible.

Gini impurity ore entropy are use to quantified the likelihood that a randomly selected instance would be misclassified if it were assigned a label based on the class distribution within a specific node:

$$\begin{aligned} \text{gini} &= 1 - \sum_{i=1}^{C} p_i^2 \\ \text{entropy} &= 1 - \sum_{i=1}^{C} p_i \log_2 p_i \end{aligned} \tag{5}$$

where p_i is the proportion of class i instances in the node, and C is the number of classes. Lower gini or entropy values indicate purer nodes.

The RF trains multiple trees on different subsets of data using bootstrap sampling (bagging) and aggregating their predictions via majority voting for classification. Additionally, at each node, it selects a random subset of features to determine the best split, reducing correlation between trees and improving generalization.

2.4 Particle Swarm Optimization

Inspired by swarm intelligence, Particle Swarm Optimization (PSO) optimizes solutions by simulating the cooperative movement of birds and fish in nature. It was introduced by James Kennedy and Russell Eberhart in 1995 [17] as a computational method for optimizing nonlinear functions. Each particle i in the swarm is represent by position x_i, velocity v_i, personal best position p_i and global best position g. The personal best and global best in PSO are determined based on the fitness function $F(x)$, which is defined differently depending on the specific problem.

The velocity and position of each particle are adjusted at every iteration based on the following equations:

$$w_i^{(t+1)} = wv_i^{(t)} + c_1r_1(p_i - x_i^{(t)}) + c_2r_2(g - x_i^{(t)}) \tag{6}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{7}$$

where w is the inertia weight; c_1, c_2 are the acceleration coefficients and r_1, r_2 are random numbers uniformly distributed in [0, 1]. The inertia weight w balances exploration and exploitation. The algorithm runs for a fixed number of iterations or until convergence.

For integrating PSO with SVM [18] or RF classifier, in order to select useful feature [19], each particle of PSO algorithm is a binary vector of length D, which is the number of features, a value of 1 means the feature is selected, and 0 means it is ignored.

Each particle's performance is evaluated using the classifier on the selected features, with the fitness function maximizes classification accuracy while minimizing the number of selected features:

$$F = \alpha \cdot \text{accuracy} - \beta \cdot \frac{\text{number of selected features}}{\text{number of total features}} \tag{8}$$

where α, β are weights controlling the trade-off between accuracy and feature reduction. Particles update their positions and velocities using (6), (7), and a sigmoid function converts velocity into a probability to decide whether a feature is selected.

2.5 Complete Algorithm

The complete algorithm for fault classification is shown in Fig. 1. First, the vibration signals are collected using sensors placed at several positions in the rotating machine system, then these time series are separated into samples with a length of N points, in this study, $N = 1024$. A total of 31 time-domain features are extracted from the above samples and stored as feature datasets for classification.

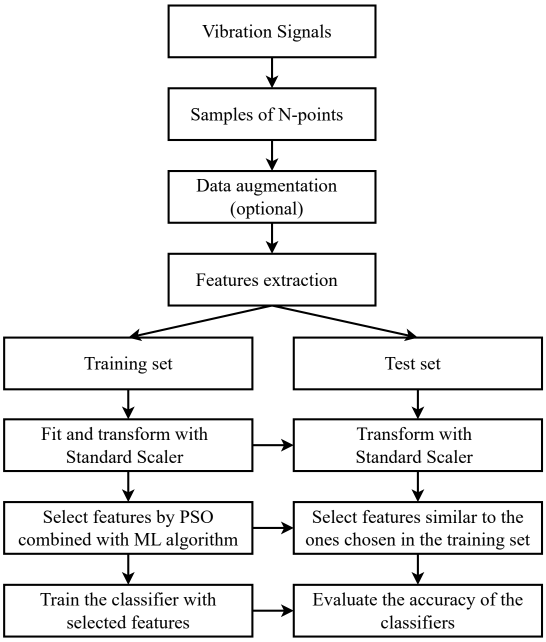
Vibration Signals
Samples of N-points
Data augmentation (optional)
Features extraction
Training set
Test set
Fit and transform with Standard Scaler
Transform with Standard Scaler
Select features by PSO combined with ML algorithm
Select features similar to the ones chosen in the training set
Train the classifier with selected features
Evaluate the accuracy of the classifiers

FIGURE 1. Flowchart for fault classification

Since the data used for training the model is often limited, and using only the available data cannot fully evaluate models in cases where signals from vibration sensors are affected by noise, this study also considers the case where the signal is augmented with Gaussian white noise $w \sim \mathcal{N}(0, \sigma_n^2)$ before extracting features, $\sigma_n = \eta \cdot \sigma$ where η is called noise level or noise factor, σ is the standard deviation of the original signal. Typically, the signal-to-noise ratio (SNR-dB), which can be computed from η and vice versa, is used more frequently than the noise level.

The feature datasets are split into training set and test set, with the ratio of 75% and 25%, respectively. The Standard Scaler, the algorithm for selecting features and the classifiers, which are SVM or RF, use the training set as input in order to train the model. The test set, in the other hand, is used to evaluate the accuracy of the model, with the selected features by PSO algorithm.

To assess the model's effectiveness, a confusion matrix should be utilized. This table compares actual labels with predicted ones, offering a comprehensive breakdown of correct and incorrect classifications. Key metrics which can be derived form the confusion matrix, such as accuracy, precision, recall or F1-score can be useful for imbalanced datasets, where accuracy alone can be misleading. However, the three case studies in this research use balanced datasets, so the the confusion matrix and accuracy metric are sufficient to evaluate the classification model.

3 Experimental verification

To verify the effectiveness of the proposed methods, three datasets were examined.

3.1 Dataset

First, the CWRU bearing dataset [11] is one of the most extensively used datasets for assessing algorithms designed for fault diagnosis and detection in rotating machinery. It was collected by the Bearing Data Center at Case Western Reserve University, USA, and has been extensively utilized in research related to machine learning, deep learning and fault detection in mechanical systems. The dataset was collected at 12 kHz or 48kHz using and electric motor test rig equipped with accelerometers to measure vibration signals. Bearings were tested under various load conditions, from 0 to 3HP and faults were created in different size in diameter: 0.007, 0.014 and 0.021 inches. In this study, only vibration signals at the drive end (DE) was used. There were a total of 400 samples for each state, which were randomly split into 300 samples for training and 100 sample for testing.

Second, UOC gear fault dataset was collected by University of Connecticut [12], with the vibration signals sampled at 20 kHz. In this dataset, nine distinct conditions were applied to the pinions on the output shaft, include a healthy state as well as eight fault states such as missing tooth, root crack, spalling and chipping tip. There were only 312 samples of each state in this dataset, which were randomly split into 234 samples for training and 78 sample for testing. As stated in [4], this dataset in one of the most challenging one to analyze, possibly due to the limited number of samples and requires careful preprocessing.

Finally, SEU dataset was provided by Southeast University [13], contained a bearing dataset and a gearbox dataset. This study investigates the gearbox dataset, with two kinds of working conditions at 20Hz - 0V and 30Hz - 2V. There are ten working conditions, with a total of 1023 samples for each state. Just like the previous two datasets, 75% of the samples, equivalent to 767 samples, were used for training, and the remaining 25%, equivalent to 256 samples, were used for testing.

3.2 Experimental results

First, the original signal is analyze without applying Gaussian white noise. The models are trained using the training set, and their performance is assessed based on the test set's accuracy, with results shown in Table 1. Clearly, the PSO algorithm significantly reduces the number of features required—by nearly half—while still maintaining an almost perfect accuracy.

TABLE 1. Accuracy of classification algorithms without noise

Dataset	Algorithm	Accuracy	Number of features
CWRU	PSO + SWM	99.00%	16
CWRU	PSO + RF	99.80%	17
UOC	PSO + SWM	99.00%	15
UOC	PSO + RF	99.29%	15
SEU	PSO + SWM	99.92%	13
SEU	PSO + RF	99.88%	17

Second, the original signal is augmented with Gaussian noise to evaluate the model's adaptability under real-world conditions. The accuracy is presented in Table 2 and Fig. 2, corresponding to different noise levels, a smaller SNR means higher noise levels. Even in cases of significant noise, the models maintain an acceptable level of accuracy across all scenarios. Notably, with the SEU dataset, accuracy remains unchanged even when the data is augmented with Gaussian noise, whereas with the CWRU dataset, accuracy only decreases slightly. The UOC dataset, on the other hand, appears to be the most challenging to analyze, possibly due to the limited number of training samples.

4 Conclusions

It can be concluded that the proposed classification methods achieve perfect accuracy with a moderate number of required features, especially when a large amount of training data is available. In cases with fewer data samples, the accuracy remains acceptable. The strength of the PSO algorithm is that it can be combined with various machine learning and deep learning algorithms to select suitable features for classification in different specific problems. This is especially useful when the original

TABLE 2. Accuracy of classification algorithms with noise

Dataset	Algorithm	Accuracy		
		SNR=0dB	SNR=2dB	SNR=4dB
CWRU	PSO+SVM	96.20%	96.50%	97.60%
CWRU	PSO+RF	97.90%	98.50%	98.60%
UOC	PSO+SVM	84.76%	85.75%	93.87%
UOC	PSO+RF	83.33%	84.05%	92.31%
SEU	PSO+SVM	99.84%	99.65%	99.84%
SEU	PSO+RF	99.80%	99.88%	99.92%

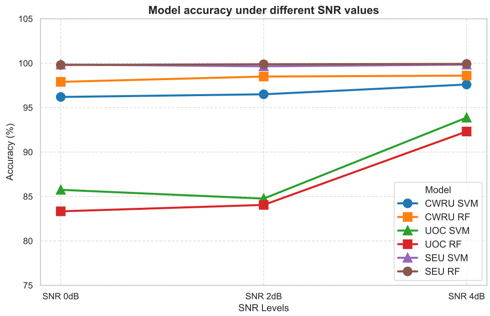

FIGURE 2. Model accuracy under different SNR values

algorithm itself does not have a feature importance attribute. This study primarily uses signals from a single sensor channel. In the future, research can focus on combining signals from multiple sensors, which will certainly further improve classification results.

References

[1] T. Kim and S. Lee, “A novel unsupervised clustering and domain adaptation framework for rotating machinery fault diagnosis,” IEEE Transactions on Industrial Informatics, vol. 19, no. 9, pp. 9404–9412, Sep. 2023.

[2] Z. Chen, Y. Liao, J. Li, R. Huang, L. Xu, G. Jin, and W. Li, “A multi-source weighted deep transfer network for open-set fault diagnosis of rotary machinery,” IEEE Transactions on Cybernetics, vol. 53, no. 3, pp. 1982–1993, Mar. 2023.

[3] J. Mei, M. Zhu, W. Liu, M. Fu, and Q. Tang, “Conditional variational encoder classifier for open set fault classification of rotating machinery vibration signals,” IEEE Transactions on Industrial Informatics, vol. 20, no. 3, pp. 3038–3049, Mar. 2024.

[4] Z. Zhao, T. Li, J. Wu, C. Sun, S. Wang, R. Yan, and X. Chen, “Deep learning algorithms for rotating machinery intelligent diagnosis: An open source benchmark study,” ISA Transactions, vol. 107, pp. 224–255, Dec. 2020.

[5] H. Wang, H. Zhu, and H. Li, “A rotating machinery fault diagnosis method based on multi-sensor fusion and eca-cnn,” IEEE Access, vol. 11, pp. 106 443–106 455, 2023.

[6] S. Shao, S. McAleer, R. Yan, and P. Baldi, “Highly accurate machine fault diagnosis using deep transfer learning,” IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2446–2455, Apr. 2019.

[7] K. Tran, L. Pham, V.-R. Nguyen, and H.-S.-H. Nguyen, “A robust deep learning system for motor bearing fault detection: leveraging multiple learning strategies and a novel double loss function,” Signal, Image and Video Processing, vol. 19, no. 4, Feb. 2025.

[8] K. K. Raj, S. Kumar, R. R. Kumar, and M. Andriollo, “Enhanced fault detection in bearings using machine learning and raw accelerometer data: A case study using the case western reserve university dataset,” Information, vol. 15, no. 5, p. 259, May 2024.

[9] W. Shen, M. Xiao, Z. Wang, and X. Song, “Rolling bearing fault diagnosis based on support vector machine optimized by improved grey wolf algorithm,” Sensors, vol. 23, no. 14, p. 6645, Jul. 2023.

[10] X. Wang and Y. Du, “Fault diagnosis of wind turbine gearbox based on modified hierarchical fluctuation dispersion entropy of tan-sigmoid mapping,” Entropy, vol. 26, no. 6, p. 507, Jun. 2024.

[11] Case Western Reserve University (CWRU) Bearing Data Center, “Case western reserve university (cwru) bearing data center,” 2019, accessed January 2025. [Online]. Available: https://csegroups.case.edu/bearingdatacenter/pages/download
data-file/

[12] P. Cao, Shengli Zhang, and J. Tang, “Gear fault data,” 2019, accessed January 2025.

[13] SEU gearbox datasets, “Seu gearbox datasets,” 2025, accessed January 2025. [Online]. Available: https://github.com/cathysiyu/Mechanical-datasets

[14] J. Mielniczuk and P. Wojdyłło, “Estimation of hurst exponent revisited,” Computational Statistics and Data Analysis, vol. 51, no. 9, pp. 4510–4525, May 2007.

[15] I. S. Andreas Christmann, Support Vector Machines. Springer New York, 2008.

[16] L. Breiman, Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[17] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proceedings of ICNN’95 - International Conference on Neural Networks, ser. ICNN-95, vol. 4. IEEE, pp. 1942–1948.

[18] T. T. K. Nga, T. V. Pham, D. M. Tam, I. Koo, V. Y. Mariano, and T. Do-Hong, “Combining binary particle swarm optimization with support vector machine for enhancing rice varieties classification accuracy,” IEEE Access, vol. 9, pp. 66 062–66 078, 2021.

[19] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimization for feature selection in classification: A multi-objective approach,” IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.