Journal of the Brazilian Society of Mechanical Sciences and Engineering (2023) 45:576
https://doi.org/10.1007/s40430-023-04471-9

TECHNICAL PAPER

Improved signal processing for bearing fault diagnosis in noisy environments using signal denoising, time–frequency transform, and deep learning

Hind Hamdaoui[1] · Looh Augustine Ngiejungbwen[1] · Jinan Gu[1] · Shixi Tang[2,3]

Received: 19 April 2023 / Accepted: 8 September 2023 / Published online: 9 October 2023

Abstract

Vibration signal processing is a crucial task in machine fault diagnosis. Several signal processing methods in the past relied on more conventional approaches to diagnose bearing defects. Recent advances in artificial intelligence have sparked the development of deep learning-based signal processing methods for bearing fault diagnosis. Many of the proposed methods do not apply to signals heavily polluted with noise. In this paper, a new method for bearing fault diagnosis is presented, which combines variational mode decomposition and wavelet thresholding to denoise vibration signal, time–frequency continuous wavelet transform to generate scalogram images, and deep learning structure to classify and diagnose bearing fault. The proposed method incorporates signal denoising, time–frequency transform, and deep learning to process and diagnose machine faults in the presence of background noise. Data collected from a bearing test rig are used to validate the accuracy of the proposed method, effectiveness, and robustness. The results show that the proposed method can extract "clean" vibration signals from noisy signals and accurately diagnose the fault.

Keywords Bearing fault diagnosis · Vibration analysis · Signal processing · Deep learning · Vibration analysis · Signal denoising

1 Introduction

Rolling element bearing (REB) is an important mechanical component widely used in various engineering components across many fields. Its health condition and proper functioning significantly impact the safety and stability of engineering equipment [1, 2] as well as the smooth functioning of many systems. The failure of REBs introduces dangerous and sometimes costly consequences [3, 4] to the process or system in which they are installed. The complex vibration transmission path of REBs makes it difficult to accurately diagnose REB faults [5]. As a result, advanced rolling bearing fault diagnosis methods are highly sought after to accurately diagnose and prevent catastrophic machinery failures [6].

Signal processing is a crucial task involved in machine fault diagnosis. Since the dawn of preventive maintenance and machine health monitoring, many vibration signal processing techniques have been developed to address the problem of bearing fault diagnosis. In general, signal processing techniques for machinery fault diagnosis can be classified into three broad categories: vibration analysis based on the direct manipulation of time series data, machine learning (ML) based, and deep learning (DL) based. Vibration analysis-based signal processing techniques can be further classified into time-domain, frequency-domain, and time–frequency domain analysis [7]. Traditional ML-based methods include artificial neural networks (ANN), principal component analysis (PCA), support vector machines (SVM), etc. Traditional DL-based algorithms for bearing fault diagnosis include a convolutional neural network (CNN), autoencoder, deep belief network (DBN), recurrent neural network

Technical Editor: Jarir Mahfoud.

✉ Jinan Gu
gjnan@ujs.edu.cn

[1] School of Mechanical Engineering, Jiangsu University, Zhenjiang, People's Republic of China

[2] School of Information Engineering, Yancheng Teachers University, Yancheng, People's Republic of China

[3] Jiangsu Engineering Laboratory of Cyberspace Security, Suzhou, People's Republic of China

(RNN), generative adversarial network (GAN), and deep learning-based transfer learning methods.

Time-domain analysis can be directly applied to vibration data since they are collected as time series data. Time-domain analysis is based on the use of statistical condition indicators such as mean, peak, peak-to-peak interval, and standard deviation, besides other higher order condition indicators such as probability density, kurtosis, crest factor, skewness and shape factor calculated from time-series signals to detect bearing faults [8, 9]. Time-domain vibration analysis is not limited to statistical condition indicators. Frequency-domain analysis is the most widely used technique for bearing fault diagnosis [10]. The advantages of frequency-domain analysis methods far surpass those of time-domain analysis because of their ability to easily isolate and identify important frequency components of a vibration signal as features. At the top of the list is fast Fourier transform (FFT), which can efficiently isolate narrowband spectra. Vibration data from REBs contain valuable high and low ranges of frequency spectra components. Since REBs produce short-duration pulses during generation, these pulses generate high energy in specific ranges of the frequency band of the power spectrum in the frequency domain. The FFT technique has been applied to generate a power spectrum commonly used to extract bearing fault features and identify bearing faults [11–13]. Other efficient frequency-domain tools for bearing fault detection commonly used include envelope spectrum analysis [14], frequency filter sideband analysis, higher order spectra [15], and cepstrum analysis [16]. Time–frequency domain analysis is considered superior to time-domain analysis and frequency-domain analysis. It extends the capability of frequency-domain analysis on nonstationary waveform signals, such as waveforms obtained under variable speed conditions. Time–frequency analysis methods commonly used for machine fault diagnosis include short-term Fourier transfer (STFT) [17, 18], Wigner-Ville distribution (WVD) [19, 20], reduced interference distribution (RID), wavelet analysis such as wavelet transform methods (continuous wavelet transform, CWT, and discrete wavelet transform, DWT) and wavelet packet transform (WPT). These methods can process nonstationary signals for machine fault diagnosis. However, they present some shortcomings which limit their application in bearing fault diagnoses, such as the fixed temporal resolution of STFT controlled by choice of SFT window and limited by the uncertainty principle, the cross-term interface of Wigner-Ville distribution, and energy leakage issue and wavelet function selection of wavelet transform [21, 22]. Empirical mode decomposition (EMD) and, later variant, empirical ensemble mode decomposition EEMD) are powerful time–frequency analysis tools used to analyze vibration signals for machine fault diagnosis [23–25]. Also, the EMD of the Hilbert–Huang transform (HHT) is another powerful technique that is used to decompose vibration signals without the base function and can be applied to both stationary and nonstationary signals [26, 27]. Some drawbacks of EMD, EEMD, and related methods are centered around computational complexity, long execution time, and complicated results with changing working conditions. The techniques discussed so far have been used to successfully diagnose machine faults but heavily rely on manual feature extraction and, in some cases, visual inspection, which requires much human expertise and is generally susceptible to errors.

More advanced fault diagnosis techniques have been developed in recent years that combine the efficient feature extraction capability using time–frequency domain transform techniques with DL's learning and classification capability to automate the machine fault diagnosis process. Using time–frequency domain techniques, 1-D vibration or acoustic emission signals can be converted to 2-D representation (image), characteristic of the corresponding fault type. The power of convolutional neural networks specializing mostly in image classification can then be exploited to efficiently and accurately classify the fault. Duong et al. [28] applied to envelop analysis to extract envelop signals from 1-D acoustic signals and then applied CWT with damage frequency band information to generate a defect signature wavelet image (DSWI). DSWI describes the acoustic emission signal in the time–frequency domain, reduces the nonstationary effect in the signal, and shows discriminate pattern visualization for different fault types associated with the defect signature. Classification results using deep CNN (DCNN) yielded 98.79% accuracy for the combined DSWI-DCNN approach. Verstraete et al. explored three time–frequency analysis techniques: STFT, WT, and HHT, to generate image representations of fault signals which are then fed into a CNN network for fault classification. This method resulted in the accurate prediction of the fault types. This method also explored noise scenarios but relied on the transform techniques to "see through" the noise. This could work in many cases. However, in others, low- or high-frequency noise could make it difficult for the CNN network to accurately distinguish the fault features from the noisy data. By far, the most common method for image generation in vibration signal analysis is CWT. The combined use of CWT and CNN for machine fault diagnosis has been widely explored in prior research [29–34]. Xiao et al. [33] developed the improved variational mode decomposition (IVMD) and CNN to process stationary vibration signals. The minimum average Pearson coefficient principle analyzes fault signals in time to determine the best signal decomposition mode. The signals are then transformed to the frequency domain using CWT for easy feature extraction. The extracted features are used to train CNN networks. IVMD-based CNN is more accurate than other methods, such as RNN, LSTM, and GRU. Despite the impressive results obtained using existing approaches, additional preprocessing techniques could be used to improve the accuracy of the results for real-world data. In real-world applications, vibration data is often mixed with noise which could make it difficult to accurately identify the faults. Advanced signal denoising techniques can reduce the noise components in vibration signals resulting in a reconstructed fault signal for fault identification. Denoising techniques such as empirical mode decomposition (EMD) [35], variational mode decomposition (VMD) [36], ensemble empirical mode decomposition (EEMD) [37], and WT have been used for time-series signal denoising. It is possible to leverage the power of these denoising techniques to preprocess real-world data before fault identification methods are applied. Even though existing DL-based fault diagnosis approaches have been accurately used to identify machine faults, these approaches still have some areas that require improvement. Most existing methods do not address a dedicated approach to dealing with very noisy vibration data often encountered in complex systems comprising many vibrations pruned subsystems. Given the fact that most existing methods use simulation data or data obtained from bearing test rigs (in the case of bearing fault diagnosis), these datasets are acquired in isolation from other equipment and components and do not represent a real-world setup. A close correlation between the noise frequency and that of the "clean" vibration signal poses a serious challenge for the DL models to accurately identify the fault features from the data.

This study proposes a novel method for bearing fault diagnosis that combines three crucial stages of signal processing: pre-processing, feature extraction, and fault classification. In the pre-processing stage, a VMD filtering method based on improved Bhattacharyya distance (VMD-IBD) with optimal parameters using the MIGA algorithm is prosed, which first uses VMD to decompose the input noise-added signal into intrinsic mode functions (IMFs). Then calculates the variance of each IMF and the original signal. Bhattacharyya distance is then used to calculate the similarity between the original signal and the IMFs to select the best IMFs for reconstructing of the filtered signal. The VMD-IBD filtering method removes low- and high-frequency noise components while l, leaving no effect on the intrinsic signal information. The next stage in pre-processing uses wavelet denoising through wavelet thresholding to remove any remaining same frequency noise from the signal. The effectiveness of the proposed denoising approach is demonstrated through comparative analysis of filtering performance metrics such as signal-to-noise ratio (SNR) and root mean square error (RMSE) against VMD and wavelet denoising. Feature extraction from the reconstructed vibration signal is achieved through 2D transform in the form of scalogram images and CNN deep neural network and classified using softmax classification at the output of the deep neural network. The performance of the proposed fault diagnosis method is compared with other DL-based approaches.

The rest of the paper is organized as follows. In Sect. 2, the proposed method for bearing fault diagnosis is discussed. Section 3 describes the experimental setup, conditions, and data acquisition methods. Section 4 presents the results and discussion of the proposed approach using bearing fault vibration signals from an experimental bearing fault test rig and other deep learning-based methods. Finally, Sect. 5 summarizes and concludes the paper with proposals for improvements to be explored in future work.

2 The proposed signal processing and fault diagnosis approach

This section describes the bearing fault diagnosis method. Figure 1 is an overview of the proposed method. The implementation of the approach is divided into several steps. The first involves data preprocessing and preparation. Gaussian white noise is added to the signal to simulate real-world data. The raw time series vibration signals, $x(t)$, are arbitrarily segmented into smaller segments of appropriate length denoted as $f(t)$ and denoised to remove noise VMD and WT. The CWT method converts the denoised signals into scalogram images and groups them according to the fault types, which will later serve as classes during classification. The image data are appropriately sized for input to the CNN. The time–frequency graphs of certain defects are similar, so a basic wavelet transform cannot reliably detect the fault frequencies. A more efficient diagnosis strategy must be developed. Technology based on deep learning has shown promise for use in data analysis. The CNN structure can discern the broad outlines of information and the small distinctions unseen to the human eye. Since CNN models are adept at multivariable processing, a time–frequency graph may serve as the input for training. As a result, the third step involves model building, training, and fault classification using CNN with the data generated in the second step. Due to CNN's excellent multivariate processing capability, it can be trained using scalogram images. The final step is testing and validation of the trained CNN model. The trained model is used to identify the bearing fault types based on the input data.

2.1 Sensor signal preprocessing

2.1.1 Additive white Gaussian noise (AWGN)

The vibration signals that validate signal processing methods are obtained from bearing test rigs. Bearing test rigs are designed for experimental investigation, often isolated from other larger systems. In real-world applications, the

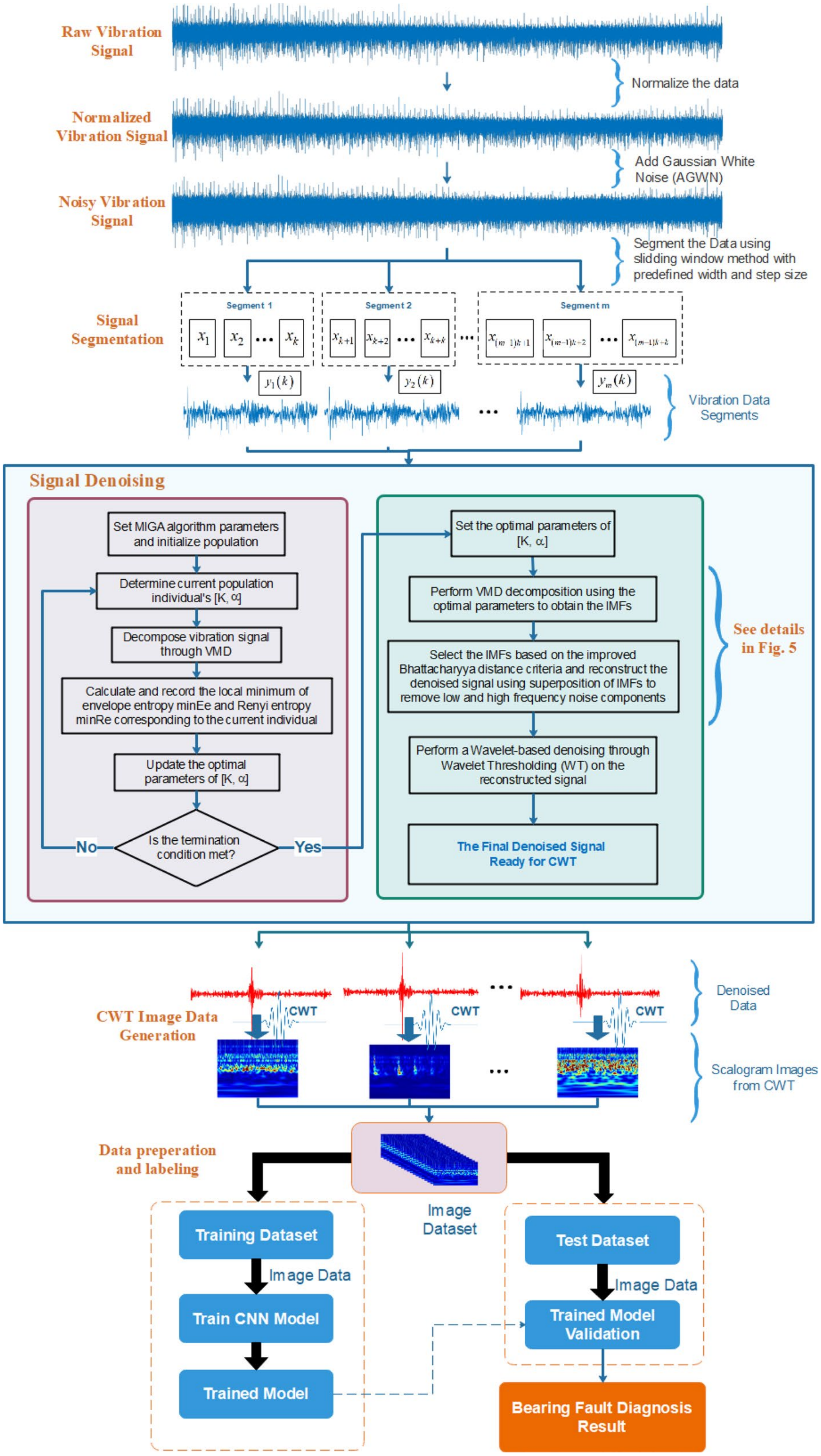

Raw Vibration Signal
Normalize the data
Normalized Vibration Signal
Add Gaussian White Noise (AGWN)
Noisy Vibration Signal
Segment the Data using slidding window method with predefined width and step size
Signal Segmentation
Segment 1
Segment 2
Segment m
x_1 x_2 ... x_k
x_{k+1} x_{k+2} ... x_{k+k}
$x_{(m-1)k+1}$ $x_{(m-1)k+2}$ ... $x_{(m-1)k+k}$
$y_1(k)$
$y_2(k)$
$y_m(k)$
Vibration Data Segments
Signal Denoising
Set MIGA algorithm parameters and initialize population
Determine current population individual's [K, α]
Decompose vibration signal through VMD
Calculate and record the local minimum of envelope entropy minEe and Renyi entropy minRe corresponding to the current individual
Update the optimal parameters of [K, α]
Is the termination condition met?
No
Yes
Set the optimal parameters of [K, α]
Perform VMD decomposition using the optimal parameters to obtain the IMFs
Select the IMFs based on the improved Bhattacharyya distance criteria and reconstruct the denoised signal using superposition of IMFs to remove low and high frequency noise components
Perform a Wavelet-based denoising through Wavelet Thresholding (WT) on the reconstructed signal
The Final Denoised Signal Ready for CWT
See details in Fig. 5
CWT Image Data Generation
CWT
CWT
CWT
Denoised Data
Scalogram Images from CWT
Data preperation and labeling
Image Dataset
Training Dataset
Image Data
Train CNN Model
Trained Model
Test Dataset
Image Data
Trained Model Validation
Bearing Fault Diagnosis Result

Fig. 1 An overview of the proposed methodology

bearing subsystem is often part of a larger system that vibrates alongside the system under investigation. These "external" vibrations introduce noise to the bearing subsystem, which can often lead to data processing errors if not carefully dealt with. To simulate a bearing test rig observed as part of a larger vibrating system, noise is added to the vibration data from a bearing test rig in the form of an AWGN to mimic the process of many random processes occurring in the whole system, a technique typically used to simulate such scenarios of data acquisition. The noise is additive because it is added to any noise that might be intrinsic to the bearing subsystem. WGN has the same probability density function as a normal or Gaussian distribution, and its values are identically distributed and statistically independent at any two points in time. The noisy data with the AWGN demonstrates the quality and accuracy of the denoising technique to retrieve the original vibration signal generating the image data used for fault identification.

An AWGN discrete vibration data X_i is the sum of input signal F_i and noise Z_i which is identically distributed and drawn from a nonzero normal distribution with variance $N = \mathrm{var}\left(F_i\right)$. The noise Z_i has a uniform power spectral density across the system's frequency band.

$$X_i = F_i + Z_i \tag{1}$$

$$Z_i \sim \mathcal{N}(0, N) \tag{2}$$

2.1.2 Time-series signal standardization

The measuring units for various variables are frequently varied in real-world situations. Before analysis, data are frequently standardized to remove the dimensional effect of variables and guarantee that each variable has the same expressive capacity. Equation (3) demonstrates how standardizing measured vibration signals rescales the signal distribution so that the mean observable value is closer to 0 and the standard deviation is closer to 1, reducing errors in signal acquisition while lowering computational complexity and calculation time. For a given sensor signal $x(k)$, the standardized signal $\tilde{y}(k)$ is defined by

$$\tilde{y}(k) = \frac{x(k) - \hat{x}_{\mu}(k)}{\sigma(x(k))} \tag{3}$$

where the mean $\hat{x}_{\mu}(k)$ and standard deviation $\sigma(x(k))$ of the original signal are defined as follows

$$\begin{aligned} \hat{x}_{\mu}(k) &= \frac{1}{N}\sum_{i=1}^{N} x_i(k) \\ \sigma(x(k)) &= \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(x_i(k) - \hat{x}_{\mu}(k)\right)^2} \end{aligned} \tag{4}$$

2.1.3 Time-series signal segmentation

To integrate the time–frequency wavelet transform into convolutional neural networks to process vibration signals and establish a bearing fault detection scheme, selecting an appropriate segment length to include at least one revolution of the shaft is crucial. The datasets collected from healthy and faulty bearings are segmented into smaller samples such that the vibration characteristics of each sample can be correctly and uniquely identified. Another advantage of segmenting the data is that the dataset can be zoomed in to capture hidden features buried deep within the wavelet transforms. An overlapping sliding window segmentation technique was applied to segment the data samples, as demonstrated in Fig. 2. The MATLAB implementation of the algorithm is presented in "Appendix A". Arbitrary segment lengths were used in this study. The segment lengths of the sliding window used in this study were chosen based on an adaptation of the Nyquist-Shannon sampling theorem [38]. In the context of vibration data analysis, this theorem suggests that the segment length should be long enough to capture at least two cycles of the highest frequency component related to the fault. Define the ball-pass frequency of outer-race (BPFO) as f_{BPFO} and the ball-pass frequency of inner-race (BPFI) as f_{BPFI}. According to the modified Nyquist-Shannon sampling theorem, the segment length L_{W} should satisfy the conditions: $L_{\mathrm{W}} \geq 2f_{\mathrm{BPFO}}$ and $L_{\mathrm{W}} \geq 2f_{\mathrm{BPFI}}$.

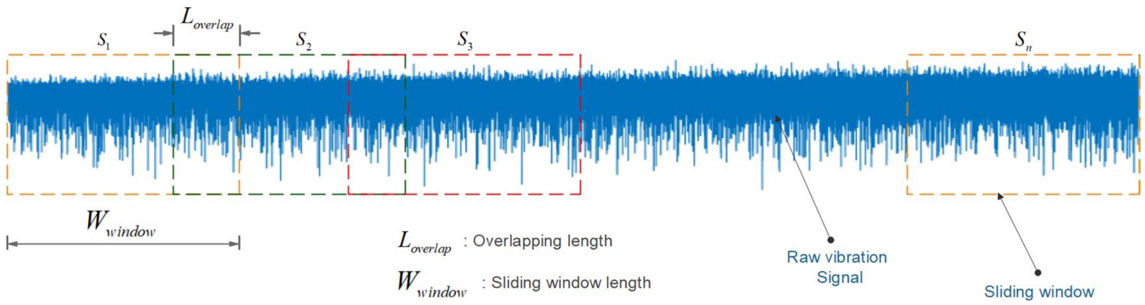


Fig. 2 Schematic illustration of the overlapping sliding window segmentation algorithm

Deep neural networks generally require a large dataset for training. The larger the dataset, the more accurately the model can be trained for its intended purpose. The segmentation technique adopted here serves the crucial purpose of data augmentation.

2.1.4 Time-series signal denoising using variable mode decomposition (VMD) and wavelet thresholding (WT)

Bearing vibration signal captured using encoders carries useful information about the signal and any faults present. During operation, the signal is not steady but includes high- and low-frequency noise components. Vibration data used in this study were obtained in a variable speed environment which makes traditional signal analysis methods such as Fourier transform and other spectral analysis methods insufficient to denoise and detect the faults. In this research, we used VMD, which filters out the low- and high-frequency noise signals from the original vibration signal. WT is then carried out to filter the same frequency noise data.

2.1.4.1 VMD decomposition A noisy time series vibration signal $x_n(t)$ can be decomposed into a finite number of intrinsic mode functions (IMFs) using VMD. The narrow-band IMFs obtained using VMD is defined by

$$x(t) = \sum_{k=1}^{K} u_k(t) = \sum_{k=1}^{K} A_k(t)\cos\left(\varphi_k(t)\right) \tag{5}$$

where u_k is an amplitude and frequency modulation signal defined as $u_k(t) = \sum_{k=1}^{K} A_k(t)\cos\left(\varphi_k(t)\right)$ with K-orders, $\varphi_k(t)$ is the phase of positive modes, and $A_k(t)$ is a slowly varying envelope signal for which each mode has an instantaneous slowly varying non-decreasing frequency, concentrated around a central value f_k. The process of VMD simultaneously calculates all the mode waveforms and their central frequencies and consists of determining a set $\langle u_k(t),\ f_k(t)\rangle$ that minimizes the constrained variation problem. The variational constraint problem can be defined as [36]

$$\begin{cases} \min\limits_{\{u_k\},\{f_k\}} \left(\sum_{k=1}^{K} \left\| \frac{\partial}{\partial t}\left[\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t)\right] e^{-j2\pi f_k(t)t} \right\|_2^2 \right), \\ \sum_{k=1}^{K} u_k(t) = x(t) \end{cases} \tag{6}$$

where $\{u_k\} := \{u_1, u_2, \ldots, u_K\}$ is the set of all modes, $\{f_k\} := \{f_1, f_2, \ldots, f_K\}$ is the set of all center frequencies, and $\delta(t)$ is a unit pulse function. The constrained problem in its current state is very difficult to solve. To calculate the waveform modes and central frequencies, the constrained problem is transformed into an unconstrained problem by finding an optimal solution to the augmented Lagrange function defined by

$$\begin{aligned} L\left(u_k(t), f_k(t), \lambda(t)\right) = {} & \alpha \sum_{k=1}^{K} \left\| \frac{\partial}{\partial t}\left[\left(\delta(t) + \frac{j}{\pi t}\right) * u_k(t)\right] e^{-j2\pi f_k(t)t} \right\|_2^2 \\ & + \left\| x(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), x(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \end{aligned} \tag{7}$$

where α is the penalty factor that ensures the signal reconstruction accuracy in the presence of Gaussian noise, $\lambda(t)$ is the Lagrange multiplier, the inner product $\langle f(t), g(t)\rangle = \int_{-\infty}^{\infty} f(t) * g(t)\mathrm{d}t$, L_2-norm $\|f(t)\|_2^2 = \langle f(t), f(t)\rangle$ and * denotes signal convolution.

The augmented Lagrange function has three parts. The first part of Eq. (7) is the regularization term which involves using the Hilbert transform to calculate the analytical signal associated with each mode, demodulate the analytical signal by multiplying with a complex exponent, and estimate the bandwidth of the analytical signal through L_2-norm. The next two terms enforce the constraint $x(t) - \sum_{k=1}^{K} u_k(t)$ by imposing a quadratic penalty and incorporating a Lagrange multiplier.

The above optimization problem defined by the augmented Lagrange function can be solved using the Alternate Direction Method of Multipliers (ADMM) algorithm proposed by Boyd et al. [39]. ADMM is a popular algorithm due to its remarkable effectiveness in minimizing objectives with linearly separable structures [40]. The complete optimization algorithm is shown in Table 1. A detailed development of the algorithm can be found in [36]. Only the final complete optimization algorithm is presented here. The saddle points of the Augmented Lagrange function are obtained by alternately updating u_k^{n+1}, f_k^{n+1}, and λ_k^{n+1}, which are the optimal solutions of Eq. (7).

The next section introduces the WT technique for enhanced noise reduction. WT is applied to the reconstructed signal from VMD.

2.1.4.2 Selection of optimum parameters for VMD using Multi-objective multi-island genetic algorithm (MIGA) VMD decomposition requires that the penalty factor α, decomposition number K, updating parameter of Lagrange multiplier τ, initialization of central frequencies f_k^1, relative and absolute tolerances ε_r and ε_a respectively, be set. The decomposition result depends on the values of α and K. In this research, $f_k^1 = 0$, $\tau = 0.01$, $\varepsilon_r = 1 \times 10^{-7}$, $\varepsilon_a = 1 \times 10^{-4}$ [36, 41, 42]. The bandwidth size of the IMF components depends on the penalty factor setting. The penalty factor α and the bandwidth are directly proportional. The value of K should be carefully chosen since a large K leads to the generation of unnecessary components. Optimization is therefore a necessary tool to set

Table 1 Algorithm for Complete Optimization of VMD

Algorithm: Complete Optimization of VMD

```
Start Algorithm
    INPUTS: $x_n(t)$, $\alpha$, $\varepsilon_r$, $\varepsilon_a$, $\tau$ (See section 2.2.3.2 for the values of these parameters)
    OUTPUT: $\{u_k\} := \{u_1, u_2, ..., u_K\}$, $\{f_k\} := \{f_1, f_2, ..., f_K\}$
    INITIALIZE PARAMETERS:
    . $\{\hat{u}_k^1\} := \{0\}$, $\hat{\lambda}^1 := 0$, $n = 0$, and $\{f_k^1\}$ is initialized using the locations of the signal peaks in the frequency domain, as
      a set of random numbers uniformly distributed in a carefully chosen interval, or as a uniformly sampled grid.
    While $\sum_k \|u_k^{n+1}(t) - u_k^n(t)\|_2^2 \big/ \|u_k^n(t)\|_2^2 < \varepsilon_r$ and $\sum_k \|u_k^{n+1}(t) - u_k^n(t)\|_2^2 < \varepsilon_a$ where $\varepsilon_r$ and $\varepsilon_a$ are the relative and
      absolute tolerances respectively, $u_k^{n+1}$ is the modal function of the $(n+1)th$ iteration, $f_k^{n+1}$ is the central frequency
      of the power spectrum of the $(n+1)th$ modal function, and $\lambda_k^{n+1}$ is the multiplication operator at the $(n+1)th$
      iteration,
    do:
      . $n := n+1$
        For $k = 1$ to $k = K$, do:
          . Update $\hat{u}_k$, the frequency domain waveform for each mode for all $f \geq 0$ using
              $\hat{u}_k^{n+1}(f) := \frac{\hat{x}(f) - \sum_{i<k} \hat{u}_k^{n+1}(f) - \sum_{i>k} \hat{u}_k^n(f) + \frac{\Lambda^n}{2}(f)}{1 + 2\alpha\left\{2\pi\left(f - f_k^n\right)\right\}^2}$   (8)
            $\hat{u}_k^{n+1}(f)$ is the Fourier Transform of the $kth$ mode of the $(n+1)th$ iteration.
          . Update the $kth$ central frequency $f_k^{n+1}$ of the power spectrum of the current modal function using
              $f_k^{n+1} := \frac{\int_0^\infty f\left|\hat{u}_k^{n+1}(f)\right|^2 df}{\int_0^\infty \left|\hat{u}_k^{n+1}(f)\right|^2 df} :\approx \frac{\sum f\left|\hat{u}_k^{n+1}(f)\right|^2}{\sum\left|\hat{u}_k^{n+1}(f)\right|^2}$   (9)
        End For
      . Dual Ascent: Update the Lagrange multiplier for the $(n+1)th$ iteration for all $f \geq 0$
              $\hat{\lambda}^{n+1}(f) = \hat{\lambda}^n(f) + \tau\left(\hat{x}(f) - \sum \hat{u}_k^{n+1}(f)\right)$   (10)
        where $\tau$ is the Lagrange multiplier update rate.
    End While
Return optimal modal functions $\{u_k\}$ and central frequencies $\{f_k\}$.
End Algorithm
```

Table 1 (continued)

$\hat{u}_k^{n+1}(f) = \int_{-\infty}^{\infty} u_k^{n+1}(t)e^{-j2\pi ft}\mathrm{d}t$, $\hat{f}_k^{n+1}(f) = \int_{-\infty}^{\infty} f_k^{n+1}(t)e^{-j2\pi ft}\mathrm{d}t$, and $\hat{\lambda}_k^{n+1}(f) = \int_{-\infty}^{\infty} \lambda_k^{n+1}(t)e^{-j2\pi ft}\mathrm{d}t$

the values of α and K. In this research, the choice range for α is $(25, 4500)$ and that for K $(2, 14)$. The MIGA algorithm is used to find the optimal values for α and K. MIGA algorithm I summarized in Table 2.

A key step in implementing the MIGA algorithm is selecting the appropriate fitness functions. The concept of envelope entropy E_e [43] and Renyi entropy R_e [44] are used to develop the fitness functions. The E_e value of each IMF component of a VMD decomposed signal can provide information about its sparsity. The larger the E_e value, the smaller the sparsity hence the noisier the IMF component. The larger the E_e value, the greater the sparsity of the IMF component, which indicates more periodic shocks in the signal. The fitness function can be written as

$$\min F_1 = \min E_e \tag{11}$$

where E_e is defined by

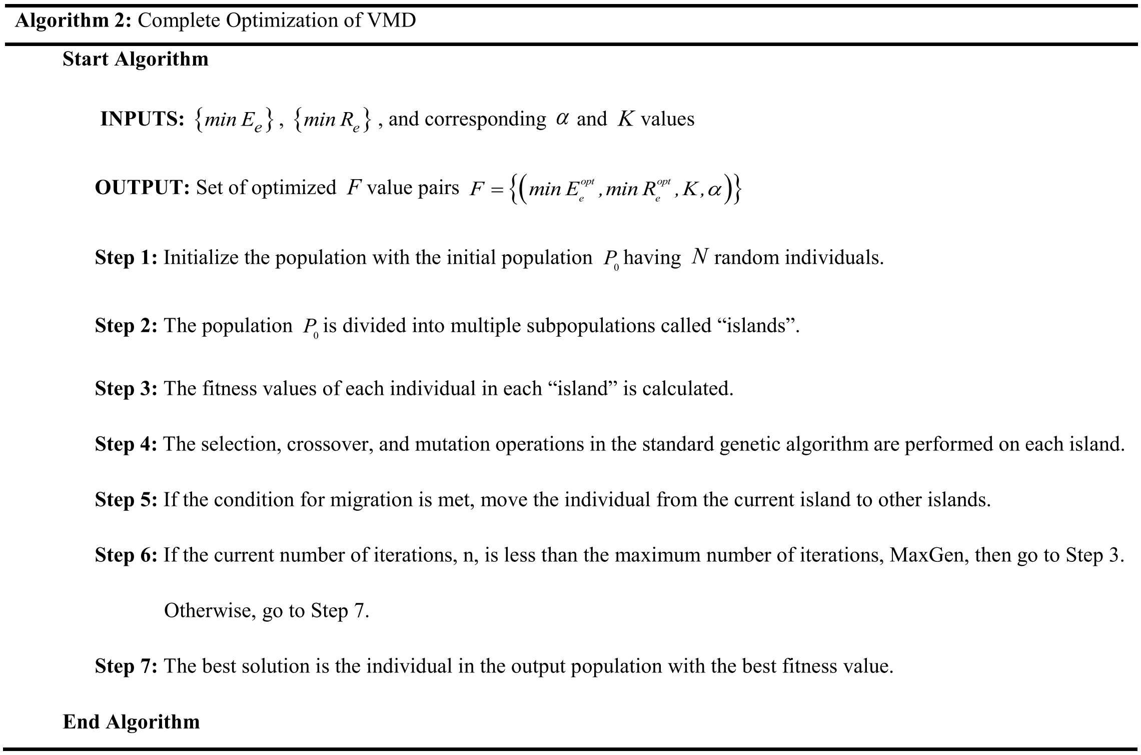
$$\begin{cases} E_e = -\sum_{j=0}^{N} e_j \log e_j;\; e_j = \dfrac{a(j)}{\sum_{j=1}^{N} a(j)}, \quad j = 1, 2, \ldots, N \end{cases} \tag{12}$$

$a(j)$ is the Hilbert demodulation of the original signal. A sample simulation signal for bearing fault can be used to show why E_e is a suitable parameter for the fitness function. The signal is expressed as

$$\begin{cases} y(t) = y_0 e^{(-2\pi f_n \zeta t)} \sin\left(-2\pi f_n \sqrt{1-\zeta^2 t}\right) \\ c(t) = \sum_{\tau} f(t-\tau) + n(t) \end{cases} \tag{13}$$

where $\tau = 0.02, 0.04, 0.06, \ldots$, $y(t)$ is a single pulse response, $n(t)$ is a Gaussian white noise whose noise intensity is determined by the standard deviation, and $c(t)$ is the sum of the time-shifted noisy impulse response. For the simulation, $y_0 = 3$, $f_n = 3000\,\text{Hz}$, $\zeta = 0.09$, $f_s = 200\,\text{kHz}$, the sampling number $N_s = 2 \times 10^6$, and the simulation time is $t = 10\,\text{s}$ to emulate that of the real signal from a bearing test rig.

Table 2 Multi-objective multi-island genetic algorithm (MIGA) for optimization of VMD



```
Algorithm 2: Complete Optimization of VMD
    Start Algorithm
        INPUTS: $\{min\,E_e\}$, $\{min\,R_e\}$, and corresponding $\alpha$ and $K$ values
        OUTPUT: Set of optimized $F$ value pairs $F = \{(min\,E_e^{opt}, min\,R_e^{opt}, K, \alpha)\}$
        Step 1: Initialize the population with the initial population $P_0$ having $N$ random individuals.
        Step 2: The population $P_0$ is divided into multiple subpopulations called "islands".
        Step 3: The fitness values of each individual in each "island" is calculated.
        Step 4: The selection, crossover, and mutation operations in the standard genetic algorithm are performed on each island.
        Step 5: If the condition for migration is met, move the individual from the current island to other islands.
        Step 6: If the current number of iterations, n, is less than the maximum number of iterations, MaxGen, then go to Step 3.
                Otherwise, go to Step 7.
        Step 7: The best solution is the individual in the output population with the best fitness value.
    End Algorithm
```

Figure 3 shows a simulated vibration waveform with increasing noise intensity and corresponding E_e values. It can be observed by visual inspection of the charts that an increase in noise intensity reduces the sparsity of the signal; hence the E_e value increases. Thus, the objective is to minimize the E_e value of the signal to expose the fault impact.

Renyi entropy R_e is very sensitive to signal changes and can be used to easily identify small changes in a signal [45]. A small noise in a fault signal increases the concentration of the man frequency as explained by better aggregation energy, hence a smaller R_e value. Thus, the objective is to minimize the R_e value, thereby decreasing the noise. The fitness function can be written as

$$\min F_2 = \min R_e \tag{14}$$

where R_e is defined by

$$R_e(X) = \frac{1}{1-\alpha} \ln \left[\sum_{k=1}^{n} \left(P\left(X = x_k\right) \right)^{\alpha} \right] \tag{15}$$

$\alpha \geq 0$ indicates the order of the R_e. The use of R_e in the fitness function is justified by observing the frequency domain waveforms of the simulated signal with increasing noise intensity, shown in Fig. 4. An increased noise level gradually submerges the main frequency in the background noise. Better noise aggregation is observed when the intensity is reduced hence a smaller R_e value, and vice versa.

For a given vibration signal, each IMF has its unique values of E_e and R_e due to the influence of the values of K and α. The final fitness function is taken as the average of the two fitness functions and used to find the most suitable values for the parameters K and α. Therefore,

$$\min F = \frac{1}{2}\left(\min E_e + \min R_e\right) \tag{16}$$

2.1.4.3 Selection criteria for Final IMF components from VMD using Improve Bhattacharyya Distance (IBD) The Bhattacharyya distance (BD) is a measure of the sensitivity or similarity of two probability density functions developed by Aril Kumar Bhattacharyya in the 1930s at the Indian Statistical Institute [46]. According to the original formulation, for any two probability distributions P and Q obtained through kernel density estimation defined on the same domain χ, the BD is defined by

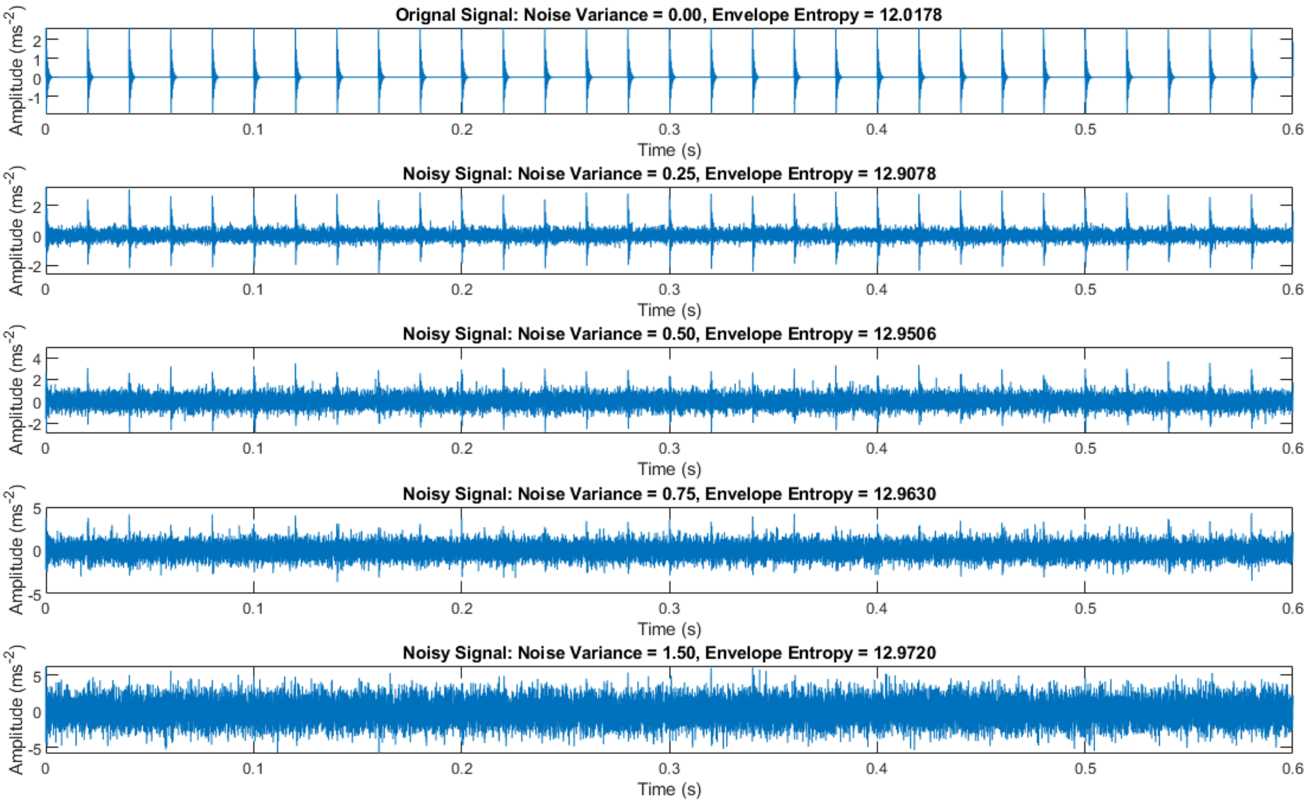


Fig. 3 Simulated time-domain waveform showing varying noise intensity as indicated by the Gaussian Noise Variance, and the variation in Envelope Entropy

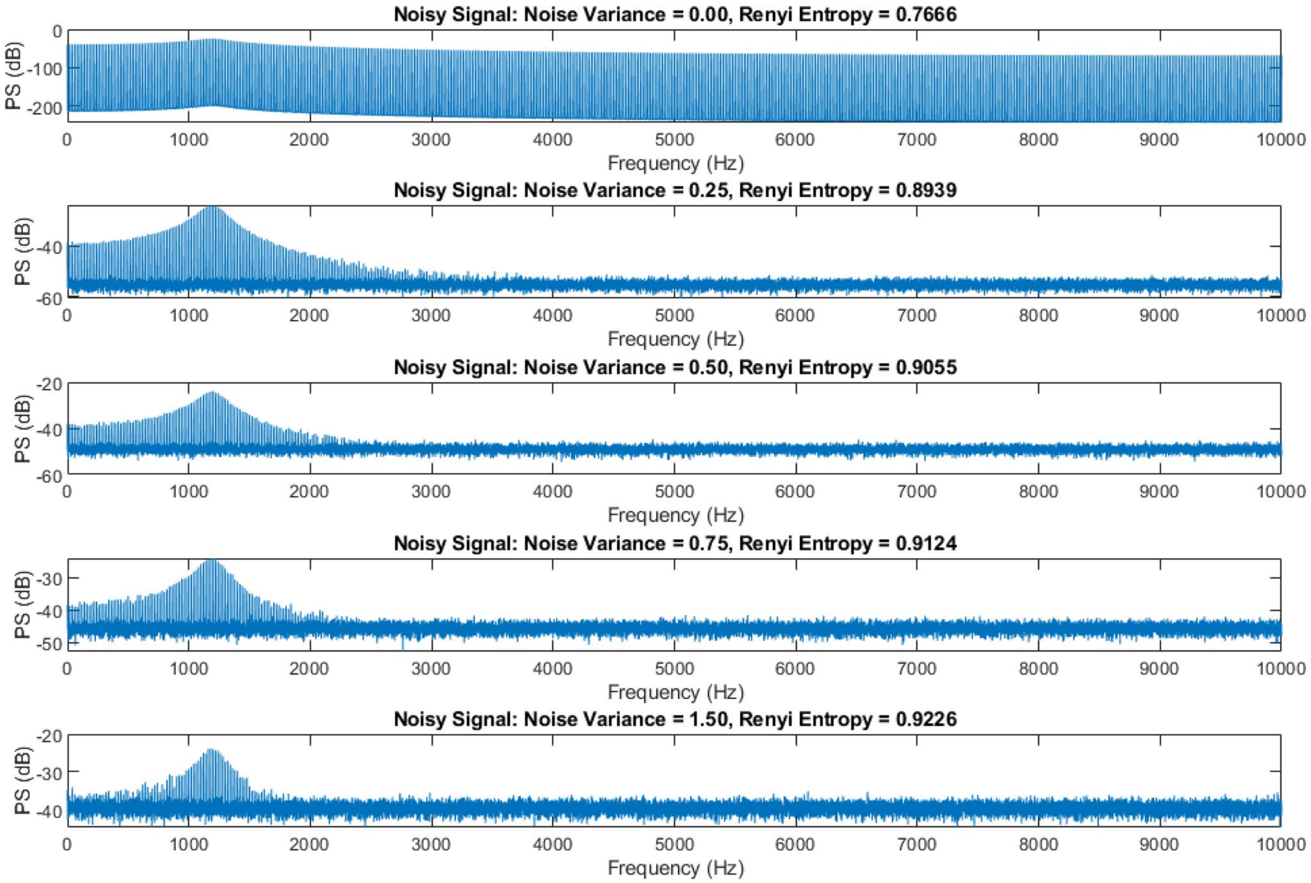


Fig. 4 Simulated frequency-domain waveform showing varying noise intensity as indicated by the Gaussian Noise Variance, and the variation in Envelope Entropy

$$D_{\mathrm{BD}}(P,Q) = -\ln[\mathrm{BC}(P,Q)] \tag{17}$$

where

$$\mathrm{BC}(P,Q) = \sum_{x\in\chi} \sqrt{P(x),Q(x)} \tag{18}$$

is the Bhattacharyya Coefficient (BC) for discrete probability distributions. IBD reformulates the BD based on variance rather than probability density and uses the new formulation to measure the distance between the two probability distributions. Since the new formulation implores variance, fewer number of computations are involved making the approach easier and computationally inexpensive.

In terms of variance, for any two discrete probability distributions X and Y, the variances are:

$$\begin{aligned} D(X) &= E\left(X^2\right) - [E(X)]^2 \\ D(Y) &= E\left(Y^2\right) - [E(Y)]^2 \end{aligned} \tag{19}$$

The BD for the two variances $D(X)$ and $D(Y)$ is defined by

$$D_{\mathrm{BD}}[D(X),D(Y)] = -\ln\left[\mathrm{BC}[D(X),D(Y)]\right] \tag{20}$$

where $BC[D(X),D(Y)]$ is the BC of the two distributions and

$$\mathrm{BC}[D(X),D(Y)] = \sum_{x\in X,y\in Y} \sqrt{D(X)D(Y)} \tag{21}$$

The IBD approach measures the similarity between the IMFs obtained by VMD decomposition and the original input noisy vibration signal to distinguish the effective from the non-effective IMF components. Through the experiments and tests on the vibration data used in this research, the VMD-IBD approach effectively removed low-frequency noise. The selected components are used to reconstruct the denoised through the principle of superposition. The steps involved in the VMD-IBD decomposition and reconstruction are summarized in the algorithm in Table 3 and illustrated in Fig. 5.

Table 3 Selection of VMD IMF components used for signal reconstruction using the Improve Bhattacharyya distance method

Algorithm 3: Steps of VMD-IBD procedure

Step 1: The original noisy input signal is decomposed by VMD using the optimum parameters of α and K obtained in the previous stage using the MIGA algorithm ([α_{opt}, K_{opt}]) to obtain K_{opt} IMFs, then the variance of each IMF and the input signal $f(t)$ is calculated, and the distance between the two variances is measured by using Bhattacharyya distance, to indicate the similarity between IMFs and input signal $f(t)$. Defined as follows:

$$L(i) = D_{BD}\left[D(IMF_i(t)), D(f(t))\right] \tag{22}$$

Step 2: Calculate the maximum slope θ_{max} of the distance between two adjacent IMF and the input signal

$$\begin{aligned} &\theta_{max} = max\left(\left|L(i+1) - L(i)\right|\right),\ i = 1,2,...,N-1 \\ &M = i_{max},\ i_{max} : Index\ at\ \theta_{max} \end{aligned} \tag{23}$$

Step 3: Reconstruct the effective IMF component to obtain a filtered signal (removes the high frequency noise components).

$$\hat{y}_{rec} = \sum_{m=K-M}^{K} IMF_m \tag{24}$$

2.1.5 Time-series signal denoising using wavelet thresholding

Raw vibration signals from a bearing test rig contain valuable information that is difficult to identify using time series data alone or other time-series processing methods. Advanced signal processing can identify crucial time–frequency domain information. Suppose $\hat{f}(t) = \left[\hat{f}(t_1), \hat{f}(t_2), \dots, \hat{f}(t_M)\right]$ is a "clean" vibration signal and $f(t) = \left[f(t_1), f(t_2), \dots, f(t_M)\right]$ is a "noisy" vibration signal, both of which are time-series signals, if σ is the noise level and z_i are independent and identically distributed standard Gaussian random variables which constitute a Gaussian white noise denoted as$z_i \sim N(0,1)$. The raw vibration signal can be modeled by

$$f(t_i) = \hat{f}(t_i) + \alpha z_i, \quad i = 1, 2, \dots, M \tag{25}$$

where $t_i = i/M$. Denoising the signal implies isolating $f(t)$ to minimize the mean-squared error defined by

$$M^{-1}E\left\|\hat{f} - f\right\|_2^2 = M^{-1}\sum_{i=1}^{M} E\left(\hat{f}(t_i) - \hat{f}(t_i)\right)^2 \tag{26}$$

subject to the fact that $\hat{f}$ is at least as smooth as f, where $\hat{f}$ is the reconstruction of f through wavelet analysis. Wavelet analysis can be defined using multiresolution analysis (MRA) approach first formulated by Mallat [47, pp. 674–693].

Orthogonal basis and local time–frequency analysis form the basis of diverse signal representations. Wavelet denoising involves using a thresholding algorithm in orthogonal decompositions such as multi-resolution analysis or wavelet packet transform. Wavelet thresholding requires that the proper threshold be selected to generate a better estimate of the denoised signal. Thresholding the wavelet coefficients of a signal keeps the local regularity of the signal and can be divided into three steps [48, 49]:

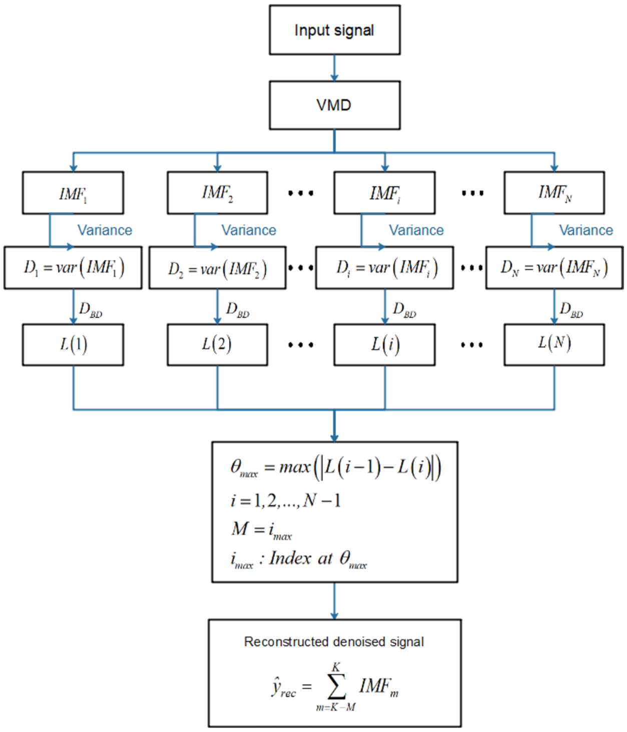


Fig. 5 Flowchart of VMD decomposition and Improve Bhattacharyya distance criteria for IMF selection and signal reconstruction

(1) *Decomposition* This step involves sequentially dividing the data into different components at different resolutions. Each decomposition results in two signals called approximation and details. The approximation signal is further decomposed into a new approximation and detail signal. The number of decompositions refers to the level, which depends on the analysis, the frequencies of interest, and the selected filters or wavelet type. To compute the wavelet decomposition of the signal at a level N on an orthogonal basis, choose a wavelet type and level used to generate a filter bank of conjugate mirror filters.

(2) *Thresholding* Decomposition is followed by selecting of a suitable threshold using a thresholding estimator on the orthogonal basis, applied on all coefficients, not including the lowest frequency energy components. Decomposition maintains the signal regularity.

(3) *Reconstruction* The final step is wavelet reconstruction of the denoised signal using the approximation coefficients of level M and the modified detail coefficients of levels from 1 to M.

MRA uses discrete wavelet transform (DWT) to decompose a signal using a series of conjugate mirror filter pairs. DWT analysis is very efficient and suited for transient, time-varying signals and performs best in detecting discontinuities and minute changes. For a basic formulation of MRA for wavelet analysis, consider a time series function $f(t) \in L^2(\mathbb{R})$ with respect to the wavelet function $\psi(t)$ and a scaling function $\varphi(t)$. A wavelet series expansion of $f(t)$ results in a series of coefficients called the DWT of $f(t)$. The DWT pair is defined as

$$W_\varphi\left(j_0, k\right) = \int_0^1 f(t)\,\overline{\varphi_{j_0,k}(t)}\,\mathrm{d}t = \frac{1}{\sqrt{M}} \sum_{t=0}^{M-1} f(t)\,\varphi_{j_0,k}(t) \tag{27}$$

$$W_\psi(j, k) = \int_0^1 f(t)\,\overline{\psi_{j,k}(t)}\,\mathrm{d}t = \frac{1}{\sqrt{M}} \sum_{t=0}^{M-1} f(t)\,\psi_{j,k}(t) \tag{28}$$

and

$$f(t) = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} W_\varphi\left(j_0, k\right) \varphi_{j_0,k}(t) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{k=0}^{M-1} W_\psi(j,k)\,\psi_{j_0,k}(t) \tag{29}$$

where $M = 2^j$. The transform comprises of M coefficients. With a minimum and maximum scale of 0 and $j-1$ respectively. The coefficients $W_\varphi\left(j_0, k\right)$ and $W_\psi(j,k)$ are the approximation and detail coefficients, respectively, computed through DWT. The required signal $f(t)$ is reconstructed through DWT synthesis or inverse DWT. The process in Eqs. (27)–(29) is valid only for orthonormal basis and tight frames [50], and Eq. (29) is called the multiresolution expansion of $f(t)$.

The MRA process consists of a sequence of successive approximation closed subspaces V_j defined by $V_j = \overline{\operatorname{span}_k \left\{\varphi_{j,k}(t)\right\}}$ (where V_j can be increased by increasing j), which satisfies the conditions $\ldots V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \ldots$, $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L^2(\mathbb{R})$ where $\bigcap_{j\in\mathbb{Z}} V_j$ denotes [51]. Each subspace V_j is the scale space. First, the signal f is projected ontoV_j. The space is then decomposed into a lower resolution space V_{j+1} which satisfy $V_{j+1} \perp$ and $V_{j+1} = V_j \oplus W_j$ [51], where $W_j = \operatorname{span}_k \left\{\psi_{j,k}(t)\right\}$. The scaling function is defined by

$$\varphi_{j_0,k}(t) = 2^{\frac{j_0}{2}} \varphi\left(2^{j_0} t - k\right), \quad \forall j_0, k \in \mathbb{Z}, \quad \varphi_{j_0,k}(t) \in L^2(\mathbb{R}) \tag{30}$$

k determines the position of $\varphi_{j_0,k}(t)$ along the time axis, j_0 determines its width, and $2^{j_0}/^2$ controls its amplitude. The wavelet function is defined by

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} \psi\left(2^j t - k\right) \tag{31}$$

$\psi(t)$ is also called the mother wavelet. Both $\varphi_{j_0,k}(t) \in B$ and $\psi_{j,k}(j,k) \in B$ where $B = \left[\left\{\psi_{j,k}(j,k)\right\}, \left\{\varphi_{j_0,k}(t)\right\}\right]$ is an orthogonal basis. The wavelet basis functions are obtained from a single mother wavelet by translation and scaling.

The basic concept of MRA is to find the approximate features and details of the signal through scalar products using scaling and wavelet functions. Vibration signals used in this study have sharp and gentle spikes observed where the faults occur. These fault spikes could be of high or low frequency, depending on the nature of the fault and the system parameters and functioning. The spikes observed, as well as other details of the signal, can be discriminated from the noise through MRA decomposition into different levels. The choice of the mother wavelet function and scaling function determines the final waveform shape of the denoised signal. Orthogonal basis wavelet is used to perform MRA, resulting in a unique reconstructed signal from the wavelet transform. MRA is optimal for extracting the useful signal and suppressing the noise. The method is characterized by high disturbance resistibility and accuracy. The decomposition is carried out on several levels. The maximum decomposition level depends on the size of the data, defined by the expression $L_{\max} = \log_2(N)$, N = length of the signal. Depending on the data, the maximum decomposition level $L_{\max}$ is rarely attained since "over decomposition" removes most of the useful components of the original signal. After performing several experiments, the decomposition level chosen for this study is 3 $\left(L_{\max} = \mathrm{ceil}\left(\log_2 15000\right) = 14\right)$, for a segment length of 15,000.

Figure 6 shows a flowchart of the thresholding procedure using MRA. The wavelet decomposition is done by filtering the input signal f by a pair of low- and high-pass mirror filters defined, respectively, by $h[k]$ and $g[k] = (-1)^{1-k}h[1-k]$. The filters $h[k]$ and $g[k]$ are used as filter banks at reconstruction. The decomposition results are approximates and detail coefficients evaluated using the filters $\overline{h}[k] = h[-k]$ and $\overline{g}[k] = g[-k]$. The relationship between the low- and

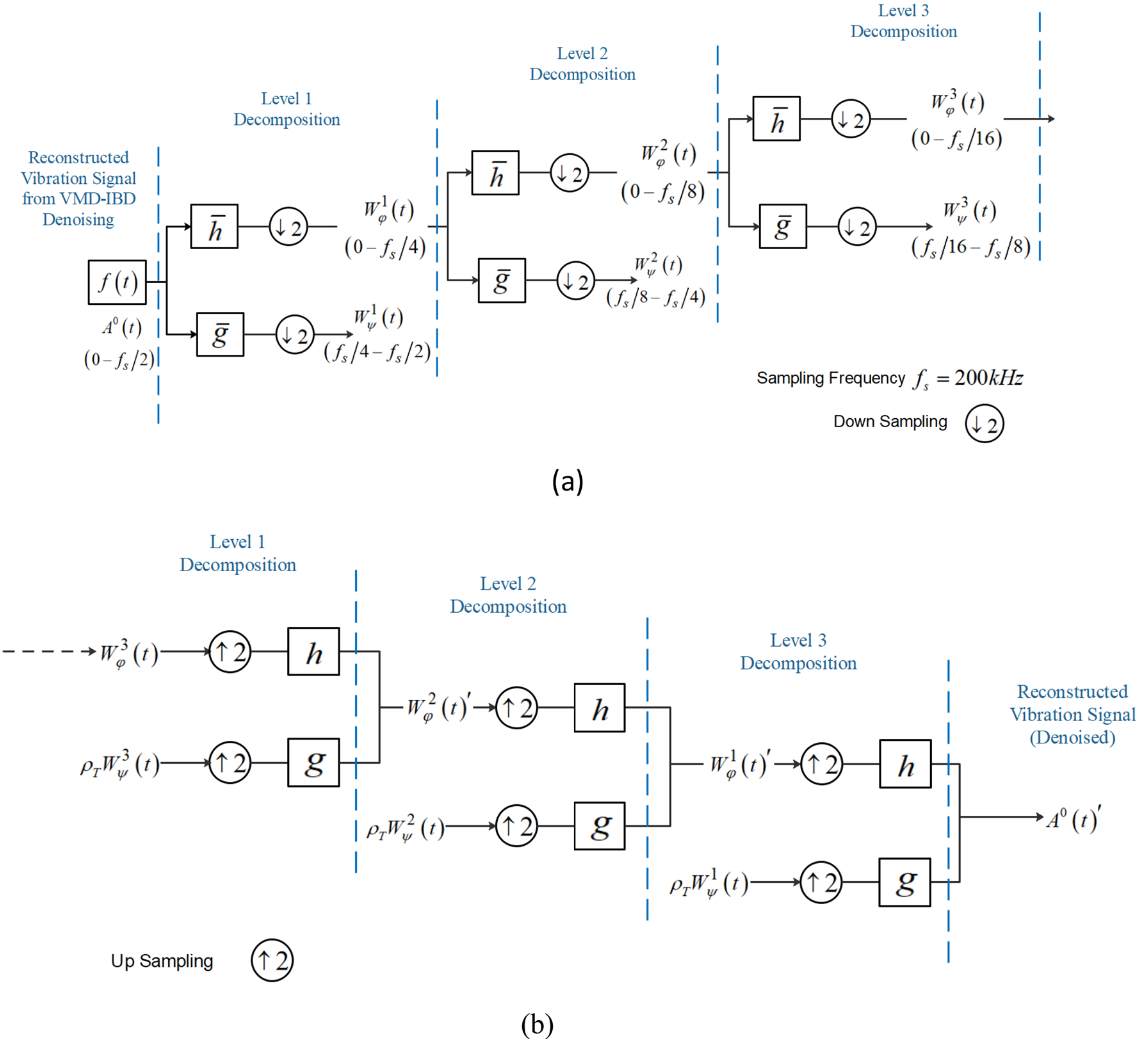


Fig. 6 Thresholding procedure with MRA, where the lowest-frequency approximate $W_\varphi^3(t)$ is kept (no processing using the thresholding function) **a** 3-level decomposition flowchart, **b** 3-level reconstruction flowchart

high-pass filter and the scaler and wavelet functions can be expressed as

$$\begin{aligned} W_{\varphi}^{j+1}[p] &= \sum_{k=-\infty}^{+\infty} h[k-2p]\, W_{\varphi}^{j}[n] = W_{\varphi}^{j} * \overline{h}[2p] \\ W_{\psi}^{j+1}[p] &= \sum_{k=-\infty}^{+\infty} g[k-2p]\, W_{\varphi}^{j}[n] = W_{\varphi}^{j} * \overline{g}[2p] \end{aligned} \tag{32}$$

At the reconstruction stage,

$$\begin{aligned} W_{\varphi}^{j}[p] &= \sum_{k=-\infty}^{+\infty} h[p-2n]\, W_{\varphi}^{j+1}[n] + \sum_{k=-\infty}^{+\infty} g[p-2n]\, W_{\varphi}^{j}[n] \\ &= W_{\varphi}^{j+1'} * h[p] + W_{\psi}^{j+1'} * g[p] \end{aligned} \tag{33}$$

The choice of wavelet function is not unique. However, it is chosen from several "already-built" wavelets based on the signal or image processing domain and is required to satisfy the multiresolution condition [50]. Many wavelet models have been developed over the years with different effects, such as Haar wavelet for rectangular-shaped signals and images, Daubechies wavelet for signal compression and solving fractal problems, Morlet wavelet for wavelet-based analysis, Meyer wavelet for image processing, image restoration, and biomedical signal compression, Symlet wavelet which is an asymmetrical Daubechies wavelet suitable for signal denoising, etc. Signal denoising within this study's context will use the eighth-order symlet wavelet (sym8). A flowchart of the WT denoising algorithm is shown in Fig. 7.

Donoho and Johnstone [52] suggested that the significant wavelet coefficients could be extracted by thresholding. The thresholding shrinks the wavelet coefficients towards zero if their absolute value is below a certain threshold level $\gamma \geq 0$. The simplest form of thresholding is hard thresholding which can be defined as

$$\eta_{\mathrm{H}}\left[\alpha_m(j,k),\gamma\right] = \begin{cases} \alpha_m(j,k), & \left|\alpha_m(j,k)\right| \geq \gamma \\ 0, & \left|\alpha_m(j,k)\right| < \gamma \end{cases} \tag{34}$$

where $\eta_{\mathrm{H}}\left[\alpha_m(j,k),\gamma\right]$ is the wavelet estimation coefficient after hard thresholding, $\alpha_m(j,k)$ is the wavelet decomposition coefficient before thresholding, and γ is the threshold level. This approach is a keep-or-kill approach that results in discontinuities in the amplitude of the shrunk coefficients. Soft thresholding approach, however prevents discontinuity and can be expressed as [53]

$$\eta_{\mathrm{S}}\left[\alpha_m(j,k),\gamma\right] = \begin{cases} \operatorname{sgn}\left(\alpha_m(j,k)\right)\left(\left|\alpha_m(j,k)\right| - \gamma\right), & \left|\alpha_m(j,k)\right| \geq \gamma \\ 0, & \left|\alpha_m(j,k)\right| < \gamma \end{cases} \tag{35}$$

where $\eta_{\mathrm{S}}\left[\alpha_m(j,k),\gamma\right]$ are the wavelet coefficients after the soft thresholding process and $\operatorname{sgn}(\cdot)$ is the signum function. This is the shrink-or-kill approach that prevents the discontinuities observed during hard thresholding. However, all other coefficients which represent the original signal are also shrunk, causing a decrease in the SNR of the denoised signal, which is highly undesirable.

Universal thresholding was chosen as the denoising method, which has been proven optimal for many applications, as stated in [54]. According to the universal thresholding method, the risk of the estimator with j-level threshold given by $\mathrm{Tr}_j = \sigma\sqrt{2\ln\left(n_j\right)}, \quad j = 1,\ldots,N$ and $N \geq 4$ is

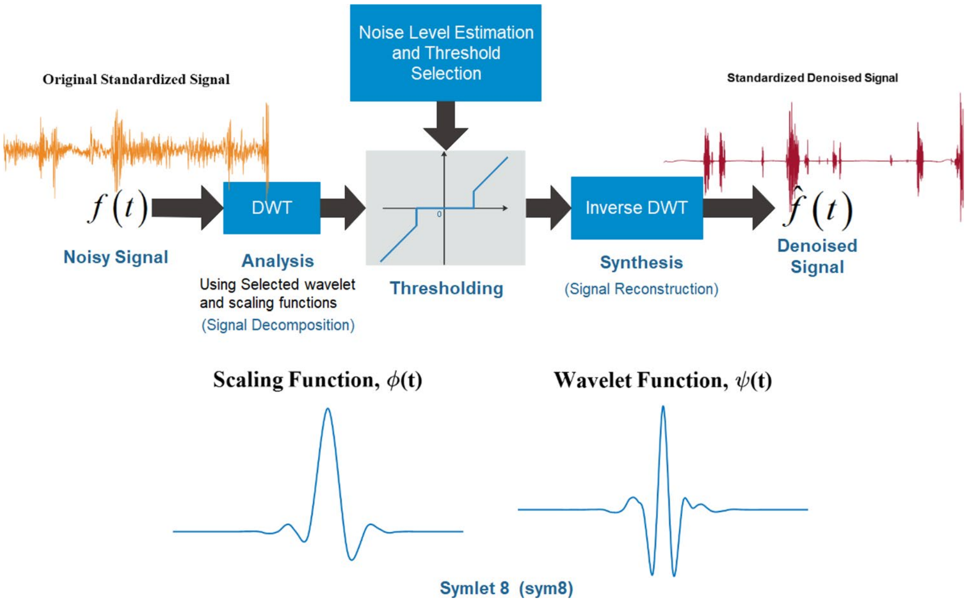

Noise Level Estimation and Threshold Selection
Original Standardized Signal
Standardized Denoised Signal
$f(t)$
DWT
Inverse DWT
$\hat{f}(t)$
0
Noisy Signal
Analysis
Using Selected wavelet and scaling functions
(Signal Decomposition)
Thresholding
Synthesis
(Signal Reconstruction)
Denoised Signal
Scaling Function, ϕ(t)
Wavelet Function, ψ(t)
Symlet 8 (sym8)

Fig. 7 Wavelet thresholding denoising algorithm flowchart

$$r_{\mathrm{th}}(f) \leq (2\ln N + 1)\left(\sigma^2 + \sum_{t=0}^{N-1} \min\left(\sigma^2, \ \left|f_B(t)\right|^2\right)\right) \tag{36}$$

where f_B is the projection of f on the basis B, σ is the standard deviation of the noise signal, r_{th} is the threshold risk, and n_j is the number of j-level wavelet coefficients. In this application, n_j is kept constant at $n_j = N$. Noise variance σ^2 estimation is an important factor during thresholding estimation. The noise variance is unknown in practical applications and must be estimated during wavelet denoising. According to Mallat [55], the variance of the noise W is estimated from the median M_x of absolute wavelet coefficients $\left|\left\langle W, \psi_{l,k}\right\rangle\right|$ by neglecting the influence of the piecewise smooth clean signal f:

$$\tilde{\sigma} = \frac{M_x}{0.6745} \tag{37}$$

The noise variance estimation is carried out using hard thresholding for the data used in this study. Due to the nature of the signal noise and through various experiments, a level-dependent variance estimation was chosen. That is, the influence of Gaussian noises is spread to all the levels; thus the variance should be estimated level by level [56].

In summary, for this research, the Universal Threshold wavelet denoising was carried out through MRA with a symlet 8 (sym8) wavelet. Level-dependent noise estimation was used with hard thresholding. The number of denoising levels was set to $L = 3$ for the best results, which retains the most information about the original signal (signal integrity) while removing as much noise as possible.

2.1.6 Performance metrics for evaluation of the denoising technique

For a given original vibration signal sequence fa and denoised vibration signal f_{den} both of length N, the root mean square error (RMSE) and signal-to-noise ratio (SNR) performance metrics are used to evaluate the quality of the denoising process. The RMSE is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left|f_{\mathrm{orig}}(n) - f_{\mathrm{den}}(n)\right|^2} \tag{38}$$

According to [57], for any two samples x and y, the correlation coefficient between the two samples is defined by

$$\rho^2(x, y) = \frac{\sigma_x^2}{\sigma_y^2} = \frac{\mathrm{SNR}}{1 + \mathrm{SNR}} \tag{39}$$

The correlation coefficient between f_{orig} and f_{den} is defined as

$$\rho\left(f_{\mathrm{orig}}, f_{\mathrm{den}}\right) = \frac{\mathrm{cov}\left(f_{\mathrm{orig}}, f_{\mathrm{den}}\right)}{\sigma\left(f_{\mathrm{orig}}\right) \cdot \sigma\left(f_{\mathrm{den}}\right)} \tag{40}$$

In terms of the samples of f_{orig} and f_{den}, the correlation coefficient can be defined as

$$\rho = \frac{\frac{1}{N}\sum_{n=1}^{N}\left(f_{\mathrm{orig}}(n) - \bar{f}_{\mathrm{orig}}\right)\left(f_{\mathrm{den}}(n) - \bar{f}_{\mathrm{den}}\right)}{\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(f_{\mathrm{orig}}(n) - \bar{f}_{\mathrm{orig}}\right)^2}\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(f_{\mathrm{den}}(n) - \bar{f}_{\mathrm{den}}\right)^2}} \tag{41}$$

The correlation coefficient here represents the percentage of the signal in the noise-added waveform. The SNR in decibels (dB) can be defined in terms of ρ as

$$\mathrm{SNR} = 10\log_{10}\left(\frac{\rho^2\left(f_{\mathrm{orig}}, f_{\mathrm{den}}\right)}{1 - \rho^2\left(f_{\mathrm{orig}}, f_{\mathrm{den}}\right)}\right) \tag{42}$$

2.2 Continuous wavelet transform (CWT)

Wavelet transforms have widely been used as great mathematical tools in various areas of signal processing to decompose and process complex signals, often in multiple dimensions [58, 59]. CWT is a mathematical method used to decompose and analyze variable signals that need variable time–frequency localization of features throughout the signal train [60]. Fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) are widely used in signal processing to analyze varying time–frequency signals but lack the crucial capability of providing a time history of the signal being processed as well as when the frequencies occur [61]. Given that fault detection scenarios often require the analysis of time series vibration signals, the changing frequencies of the signal with time make FFT and IFFT viable choices for analysis. Another reasonable argument for using wave transforms in analyzing such signals is that the features of the signals are often very complex and hidden deep in the frequency domain, which makes sense as not much can be deduced by simply observing the time series data.

The Fourier transform $F(f)$ of a signal is defined by

$$F(f) = \int_{-\infty}^{\infty} s(t)e^{-2\pi ift}\mathrm{d}t \tag{43}$$

for a time t and frequency f. Fourier transform is suitable for stationary signals and is ineffective in analyzing nonstationary signals [17, 62]. Consider a nonstationary time-varying signal $x(t)$. Wavelet transforms are base functions formed through the dilations and translation of a base prototype function $\psi(t)$. The CWT $W(a,b)$ of a signal $x(t)$ is defined by

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\psi\left(\frac{t-b}{a}\right)\mathrm{d}t, \quad \text{where } a, b \in \mathbb{R},\ a > 0 \tag{44}$$

The Morlet wavelet function can be used as the base prototype function for the construction of CWT, as defined by

$$\psi(t) = \left[\exp\left(-2i\pi f_0 t\right) - \exp\left(-2\pi^2 f_0^2 \sigma^2\right)\right] \exp\left(-\frac{t^2}{2\sigma^2}\right) \tag{45}$$

Taking the Fourier transform of Eq. (32) gives

$$\psi(f) = \sqrt{2\pi\sigma^2}\left\{ \mathrm{e}\left[-2\pi^2\sigma^2\left(f - f_0\right)^2\right] - \exp\left(-2\pi^2\sigma^2 f^2\right)\exp\left(-2\pi^2\sigma^2 f_0^2\right)\right\} \tag{46}$$

From Eqs. (32) and (33),

$$W(a,b) = \sqrt{a}\exp\left(2i\pi bf\right)\exp\left[-2\pi^2 a^2\sigma^2\left(f - \frac{f_0}{a}\right)^2\right] \tag{47}$$

where a is the scale parameter, σ is the time width of the undilated Morlet base prototype function and $s(t) = \exp\left(2i\pi ft\right)$. The frequency and scale are related by $f = f_0 / a$. CWT works by translating the base prototype function $\psi(f)$ with modulation frequency f centered at integral multiples of f_0 to form a set of frequency domain functions. For implementation, CWT expression can be written in discrete form using Riemann sum for an integral part, known as discrete wavelet transform DWT [63], as follows

$$W(k,l) = \frac{T_s}{\sigma\sqrt{2\pi 2^{k/M}}} \sum_{n=-\infty}^{\infty} s(n)\exp\left[-\frac{(n-1)}{2^{1+(k/M)} f_s^2 \sigma^2}\right] \exp\left[-2i\pi\frac{f_0}{f_s} 2^{-k/M}(n-l)\right] \tag{48}$$

where $1 \le k \le M$, $-\infty \le k \le \infty$, M denotes the product of the number of octaves and the voices in each octave, $t = nT_s$, T_s is the sampling time in seconds, and f_s is the sampling frequency in Hertz. When computed, CWT displays the transformed signal amplitude and phase distribution in time and scale.

Another useful wavelet transform technique that could be used is the short-time Fourier Transform (STFT) which is generally characterized by a fixed duration window. CWT uses dilation operations to carry out wavelet transform by dividing the time–frequency plane into resolution cells with variable lengths depending on the scale parameter a, which makes it suitable for variable frequency cases as opposed to STFT, where its resolution cells are of the fixed duration time window. Figure 8 shows sample wavelet transforms represented in two dimensions as scalogram images. The scalogram images resulting from CWT are resized to 224 × 224 × 3 to match the input of the CNN network.

2.3 Convolutional neural network (CNN)

Neural networks are a part of machine learning and are at the core of all deep learning algorithms. There are many classes of neural networks, but CNN is particularly interesting, which will be used in this research article [64, 65]. CNNs are a supervised learning method for image classification and computer vision tasks [66]. In comparison to other classification methods, CNNs provide a more scalable approach to image classification and object recognition applications by leveraging matrix multiplication to identify patterns in an image [67], with one drawback being the need for graphical processing units (GPUs) to handle the large complex computations in an optimized and reasonable time frame [68]. Another noticeable advantage of CNNs over other networks is their ability to adopt weight replications, receptive fields, and subsampling, low complexity, and anti-noise capability. The basic structure of a CNN has an input layer, convolution layers, pooling layers, fully-connected (FC) layers, and an output layer.

2.3.1 Convolution layer

The convolution layer distinguishes CNNs from other types of deep learning networks, and is where most of the computations and feature extraction occurs [69]. A typical input to the convolution layer is an RBG image. Part of the convolution layer is the feature detector, also called the filter or kernel, which “scans” across the receptive field of the image, searching for the presence of features in a process called convolution. The kernel is a 2D array of weights that may vary in size and represents part of the image. Applying the filter to a section of the image requires a dot product between the input pixels in that area and the filter matrix. The result is fed into an output array. The filter then shifts by a stride, and the process is repeated over the entire image. The output from the convolution layer is a feature or activation map. The weights of the feature map are adjusted by backpropagation and gradient descent during training. The hyperparameters in the convolution layer set initially before training commence include several filters, filter size, stride, and zero padding. The output from the output layer is activated by passing it through a Leaky Rectified Linear Unit (Leaky ReLU) to introduce nonlinearities.

Significant characteristics can be extracted by convolution, which is a special type of filtering method. There are multiple convolutional kernels in each convolutional layer under normal conditions. A convolution layer is created by convolutional filters and learnable kernels for nonlinear transformations. Convolutional filters and learnable kernels create a convolution layer for nonlinear transformations. The convolution process is expressed as

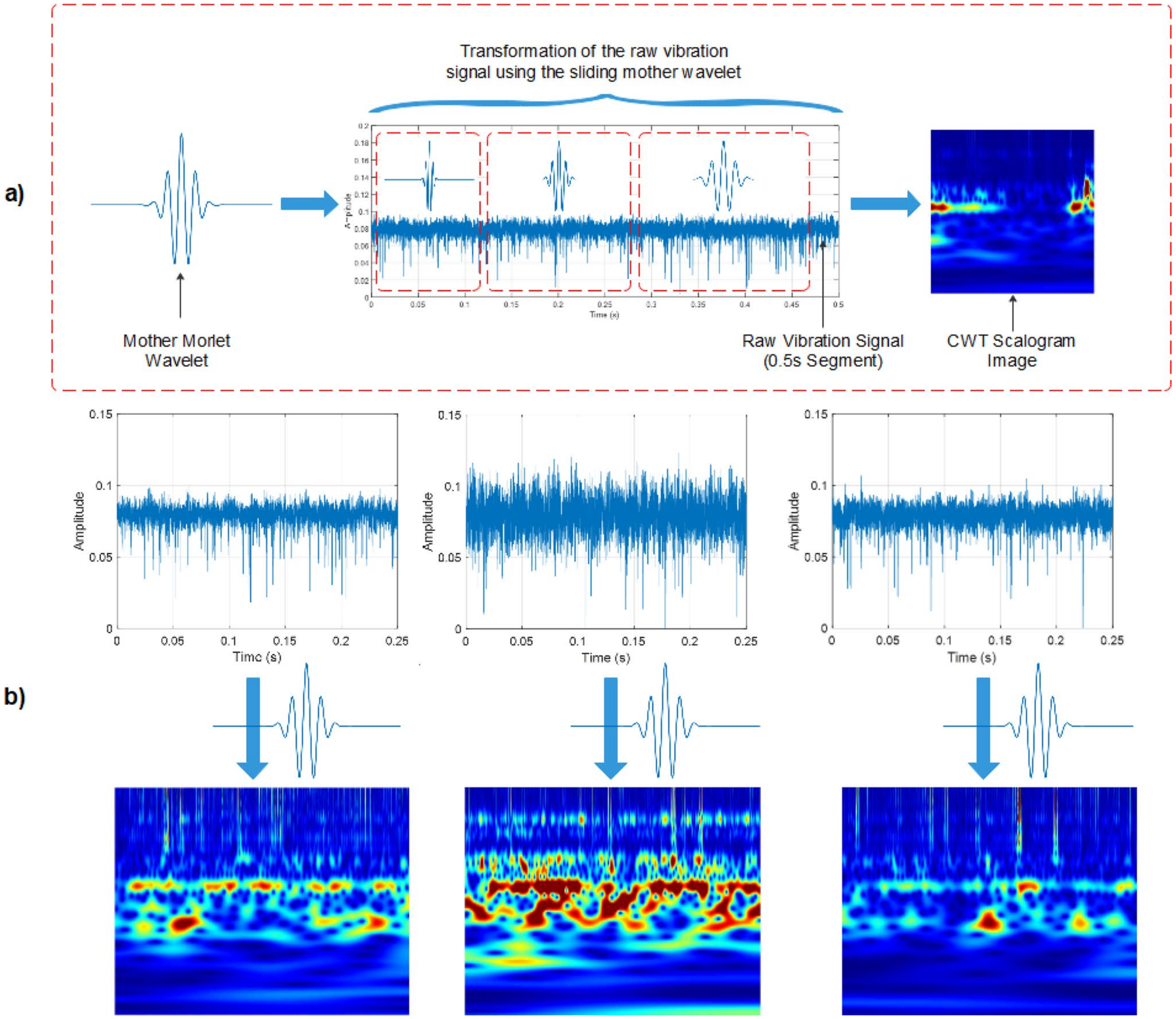


Fig. 8 **a** Flow diagram of proposed CWT wavelet transform. **b** Sample vibration signals with their corresponding CWT scalogram images, generated from time-series signal samples with 4500 datapoints into a 4500px by137px image resized to 780px by1080px for better visibility of the frequency peaks

$$y_k^n = \mathrm{conv}\left(y_i^{n-1}, \beta_{ik}^n\right) = \varphi\left(\sum_{i=1}^{M_k} y_i^{n-1} * \beta_{ik}^n + b_k^n\right) \tag{49}$$

where y_k^n is the kth feature graph of the ith convolution layer, $\varphi()$ is the activation function, M_k is the input graph set, β is the function Kernel and b_k^n is the corresponding bias.

2.3.2 Activation operation

After the convolution layer, the activation layer nonlinearly modifies the logit value of each convolution output and speeds up CNN convergence. When employing the back-propagation learning approach to update the parameters, the weights in the shallow layer become more trainable due to a leaky ReLU activation layer. The formula for the Leaky ReLU activation function is defined as follows:

$$a_k^n = \varphi\left(y_k^n\right) = \begin{cases} k_i y_{ik}^n, & y_{ik}^n \geq 0 \\ y_{ik}^n, & y_{ik}^n < 0 \end{cases} \tag{50}$$

where k_i is a fixed value in the interval (0, 1) inclusive and y_{ik}^n is the kth feature graph activation value of the convolution layer output y^n.

2.3.3 Polling layer

Pooling layers are down-sampling layers for dimensionality reduction. It reduces the number of input parameters. The pooling layer has a weightless filter that sweeps across the entire input, applying an aggression function to values within the receptive field. The results are used to populate the output matrix [69]. Max pooling sweeps a filter across the input and chooses pixels with a maximum value in the receptive field. The pooling layer reduces the input graph by extracting the main features while decreasing the dimensional complexity of the output and sensitivity to the environment with minimal loss of invalid information in the input feature graph [70]. For example, if the input feature size is 6 × 6, a pooling operation with the size of 2 × 2 and the step size of 2, down-samples the input features to an output of 3 × 3. The max pooling transformation process can be expressed as

$$y_k^n = \max_{(i-1)W \le t \le iW} \left\{ a_{tk}^n \right\} \tag{51}$$

where a_{tk}^n is the activation value of the tth neuron in the nth layer of the kth frame and W is the width of the pooling area. No parameters are learned in the pooling layer.

2.3.4 Batch normalization

The batch normalization (BN) layer reduces the deep neural network's internal covariance shift, accelerates training, increases efficiency, and improves generalization. In the first step of the BN layer process, the mean value μ_{B} of the mini-batch is subtracted from the input volume layer value and then divided by the standard deviation σ_{B}^2. However, this will result in the input value being constrained to a narrow range. As a result, once the standardization process is complete, it requires multiplication by a scaling amount denoted by γ and an offset value denoted by β. The input of the batch normalization layer is denoted by $y_k^n = \left[y_{1k}^n, y_{2k}^n, \ldots, y_{ik}^n, \ldots, y_{Nk}^n \right]$. The max-pooling transformation is described as follows:

$$\begin{aligned} \hat{y}_{ik}^n &= \frac{y_{ik}^n - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \\ z_{ik}^n &= \gamma_k^n * \hat{y}_{ik}^n + \beta_k^n \end{aligned} \tag{52}$$

where γ_k^n and β_k^n are the scaling factor and offset of the BN layer, respectively, z_{ik}^n is the output of the BN layer, and ε is a numerical stability constant.

2.3.5 Fully connected layer

Image classification and object recognition are performed in the fully connected (FC) layer based on the features extracted from previous layers. The FC is a finite number of neurons whose input and output are vectors [69]. It uses a softmax activation function to classify inputs accordingly. Let $\left\{a^i\right\}_{i=1}^n$ be the input sample set of the data to the FC layer. If $P\left(a^i = k|x^i\right)$ is the probability that the sample matches the correct label, then according to the softmax regression model, which constitutes the softmax activation function, the output is given by

$$R = \begin{bmatrix} P\left(a^i = 1|x^i;\ \phi_1^T x^i\right) \\ P\left(a^i = 1|x^i;\ \phi_2^T x^i\right) \\ \vdots \\ P\left(a^i = 1|x^i;\ \phi_K^T x^i\right) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} e^{\phi_j^T x^i}} \begin{bmatrix} e^{\phi_1^T x^i} \\ e^{\phi_2^T x^i} \\ \vdots \\ e^{\phi_K^T x^i} \end{bmatrix} \tag{53}$$

where x^i is the input eigenvector of the ith sample, K the number of classes or sample labels, the parameters of the softmax classification model is $\phi = \left[\phi_1,\ \phi_1,\ \ldots,\ \phi_K\right]^T$, and R is the output. A dropout layer can be added after the fully connected layer to limit overfitting [70]. The key goal of introducing the dropout layer is to remove nodes (input and hidden layer) in the neural network during training, thereby preventing overfitting. The dropout layer temporarily removes all forward and backward connections with the dropout nodes. This creates a new architecture from the existing parent network through a dropout probability. Overfitting happens when some neurons change in response to the mistakes of other neurons. This creates a complex co-adaptation between neurons, which in turn causes the overfitting problem. The dropout layer prevents co-adaptation between neurons, by randomly dropping out neurons thereby enhancing generalization.

For this study, the back-propagation (BP) algorithm and adaptive momentum estimation algorithm (Adam optimizer) [71] are used to train and optimize the training parameters of the CNN model. The training parameters are optimized by calculating the error between the real and predicted values. The weight and biases (trainable parameters) are rapidly updated and fine-tuned to minimize the estimated error. The training error is measured using Categorical Cross-Entropy (CCE) loss function $F(\theta)$ is defined as follows

$$\begin{cases} F(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{n} y_k^{(i)} \log\left(\hat{p}_k^{(i)}\right) \\ \hat{p}_k^{(i)} = \frac{e^{V_i}}{\sum_j e^{V_j}} \end{cases} \tag{54}$$

where n is the number of samples and m is the number of categories. Since the last layer is connected to the softmax layer for classification, CCE is used as the loss function such that the framework can optimize and calculate the gradients together.

The update rules for the Adam optimizer are defined as

$$m_{ij}^{(t)} = \beta_1 m_{ij}^{(t-1)} + \left(1-\beta_1\right)\frac{\partial C^{(t)}}{\partial w_{ij}}; \quad v_{ij}^{(t)} = \beta_2 v_{ij}^{(t-1)} + \left(1-\beta_2\right)\left(\frac{\partial C^{(t)}}{\partial w_{ij}}\right)^2 \tag{55}$$

where $m_{ij}^{(t)}$ is the decay rate of the mean gradients, $v_{ij}^{(t)}$ is the mean square of gradients, w_{ij}, β_1, and β_2 are weights, and C, the cost function that requires optimization. The normalized mean of gradients $\hat{m}_{ij}^{(t)}$ and mean of square gradients $\hat{v}_{ij}^{(t)}$ are defined as

$$\hat{m}_{ij}^{(t)} = \frac{m_{ij}^{(t)}}{\left(1-\beta_1^t\right)}; \quad \hat{v}_{ij}^{(t)} = \frac{v_{ij}^{(t)}}{\left(1-\beta_2^t\right)} \tag{56}$$

The final update rule for each weight w_{ij} is defined as

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \frac{\eta}{\sqrt{\hat{v}_{ij}^{(t)} + \varepsilon}}\hat{m}_{ij}^{(t)} \tag{57}$$

where η is the learning rate and ε is a numerical stability constant. For this study, $\eta = 0.00004$, $\beta_1 = 0.9$, $\beta_2 = 0.9$, and $\varepsilon = 1 \times 10^{-8}$.

3 Experimental setup, data acquisition and data preparation

3.1 Experiment setup and data acquisition

The effectiveness of the proposed fault diagnosis approach is tested and validated using experimental vibration signal data. A full experimental setup of the bearing fault test used to simulate the various fault types in this study is shown in Fig. 9.

The fault simulation test rig used to simulate a variety of bearing fault types and acquisition of vibration and rotational speed data is the Spectra Quest machinery fault simulation (model MFS-PK5M) at Ottawa University lab [72]. ER16K bearings with 9 balls, each with a pitch diameter of 38.52 mm and ball diameter of 7.94 mm, were used for the test, mounted at the ends of a central drive shaft connected to an electric motor whose speed is controlled by an AC drive unit. One end of the drive shaft had a healthy bearing, while the other end had a test bearing. An ICP accelerometer, model 623C01, was used to collect vibration data, while an EPC model 775 incremental encoder set at 1024 CPR measures the rotational speed. Vibration signals for healthy bearings and four error types were recorded during the tests for four different speed conditions at different rotating frequencies: increasing rotating speed, decreasing rotational speed, increasing then decreasing rotational speed, and decreasing then increasing rotational speed. The tests were carried out under no load conditions.

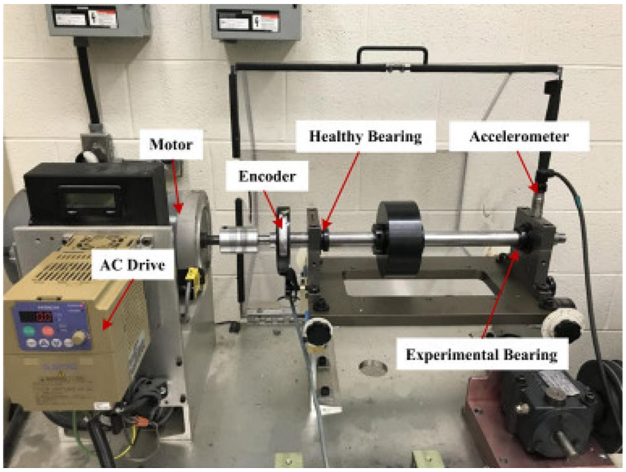


Fig. 9 Experimental Setup showing drive units, bearing units, and measuring instruments (Image source: [72])

3.2 Description of the dataset and computation environment

Experiments were conducted with data from a bearing test rig to validate the proposed signal processing method. The bearing test rig was operated under five operating conditions and varying rotational speeds. The vibration data for the tests were sampled at 200 kHz for 10 s per sample. Three trials are carried out for each operational speed setting resulting in 60 samples. Each sample has two channels: Channel 1 represents the accelerometer's vibration data, while Channel 2 represents the encoder's speed data. The bearing fault types in the University of Ottawa dataset [72] can be classified into: healthy (H), inner race fault (IF), outer race faults (OF), ball faults (BF), and combined fault (CF). The 60 samples are insufficient for any deep-learning model. Before converting into images, each sample was first divided into overlapping segments using the sliding window algorithm whose length could be chosen arbitrarily but sufficiently long enough to cover at least a complete revolution of the rotating shaft.

Three datasets were generated for this study: original dataset (data collected from the bearing test rig were not preprocessed before converting to images), noise added dataset (the data collected from the bearing test rig was supplemented with AWGN), and denoised dataset (data collected from the bearing test rig was supplemented with AWGN to simulate a scenario where the bearing subsystem is part of a larger system with vibration noise, and then denoised using the proposed denoising technique in this study). For each of the three datasets, six separate sub-datasets were generated using 2500, 5000, 7500, 10,000, 12,500 and 15,000 segment lengths that correspond to 59,940, 26,638, 17,100, 12,600, 9960 and 8220 scalogram images.

According to the parameters detailed in [72], the rotational frequency f_r of the bearing shaft falls between the range $9.8 \le f_r \le 29.0$ while the BPFO and BPFI falls between the range $35.0 \le f_{\mathrm{BPFO}} \le 103.5$ and $53.2 \le f_{\mathrm{BPFI}} \le 157.5$ respectively; thus the chosen segment lengths satisfy the conditions mentioned in Sect. 2.1.3. Each sub-dataset is grouped into five subsets corresponding to the respective fault types. Two separate sets of experiments were conducted in this study. The first set of experiments investigated the effects of denoising the vibration data instead of using unfiltered noisy data for signal processing. The second set of experiments investigates the effect of segment length and operating speed conditions on the model's overall performance. In all experiments, the same CNN architecture as previously described was used. For each experiment and image dataset, 70% of the randomized images were used to train, 20% to validate, and 10% to test the network. Each experiment was carried out 5 times, and the results averaged out to minimize the effect of randomness. This large dataset requires a large computation power made possible by GPU processors.

The computation was carried out with keras tensorflow 2.12.0 on a Dell Precision 7550 laptop with a Linux Operating system (Ubuntu 22.04.2 LTS distro): Intel Core i7 – 10875H CPU, 64 bits system from Intel corporation with 32 GB of RAM, equipped with an NVIDIA TU104GLM (Quadro RTX 4000 Mobile / Max Q) GPU graphics card and 1T SSD hard drive.

3.3 Description of the deep learning architecture

To evaluate the performance and accuracy of the proposed bearing fault diagnosis method, we apply it to vibration data from a bearing test rig under varying test conditions. Due to the complicated structure of deep learning models, it is difficult to choose appropriate parameters. However, based on prior research in the same field and experience, trial-and-error analysis was used to determine the structure and parameters of the CNN architecture in this research for fault classification and identification, as shown in Table 4, and the corresponding CNN structure is shown in Fig. 10.

The CNN architecture used for image classification is a modified version of the well-refined AlexNet [73]. The neural net comprises eight weighted layers (five convolution and three fully connected layers). In this study, the last fully connected layer has five neurons. Its output is fed to a 5-way softmax layer, producing a probability distribution over the 5 classes representing the five faults. A maxpooling layer with filter size 5×5 and stride of 2 pixels, for which the number of filters corresponds to the outputs of the respective convolution layers (96 filters and 256 filters, respectively, after C1 and C2), were added after the first two convolution layers. The subsequent 3 successive convolution layers do not have a maxpooling layer. The ReLU activation function was applied to the output of all convolution and fully connected layers.

The first convolution layer, C1 filters the 224×224×3 input image using 96 filters of size 11×11 with a stride of 4 pixels. The output has a size of 54×54×96. After passing through the first maxpooling layer P1, the output becomes

Table 4 Parameters of CNN network (Based on AlexNet architecture)

Layer name	Parameters	Output feature size	Training parameters
Input	Input Data (Scalogram Images)	227×227×3	Batch size = 32 Learning rate = 0.0001 Maximum epochs = 100 Optimizer is adam No. trainable parameters = 46,767,493
2D convolution layer (C1)	96 Filters, Size 11×11, Stride 4, activation “ReLU”	54×54×96	
Max pooling layer (P1)	96 Filter Size 3×3, Stride 2	26×26×96	
2D convolution layer (C2)	256 Filters, Size 5×5, Stride 2, padding 2, activation “ReLU”	26×26×256	
Max pooling layer (P2)	256 Filter Size 3×3, Stride 2	12×12×256	
2D convolution layer (C3)	384 Filters, Size 3×3, Padding 1, Stride 1, activation “ReLU”	12×12×384	
2D convolution layer (C4)	384 Filters, Size 3×3, Padding 1, Stride 1, activation “ReLU”	12×12×384	
2D convolution layer (C4)	256 Filters, Size 3×3, Stride 1, Stride 1, activation “ReLU”	12×12×256	
Max pooling layer (P3)	256 Filter Size 3×3, Stride 2	5×5×256	
Flatten layer (FL)	6400 Neurons	256×6400	
Fully connected layer (D1)	4096 Neurons, Dropout rate = 0.1	6400×4096	
Fully connected layer (D2)	4096 Neurons, Dropout rate = 0.1	4096×4096	
Fully connected layer (D3)	5 Neurons	4096×5	
Output layer	Classification data	5×1	

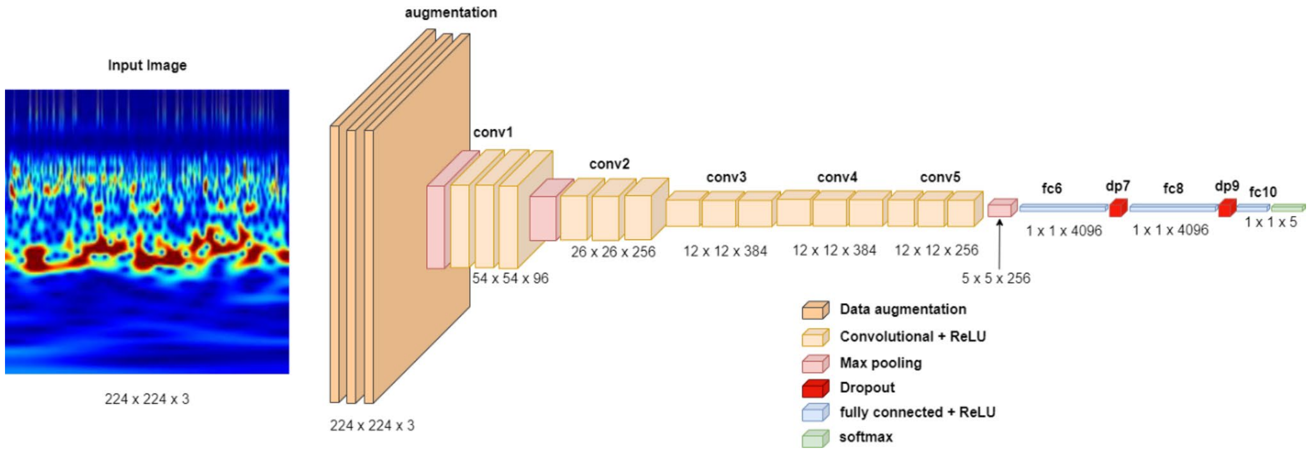


Fig. 10 CNN architecture

26 × 26 × 96, fed into the second convolution layer. The second convolution layer accepts the maxpooled output from C1 as input and filters it using 256 filters of size 5 × 5, stride of 2 pixels and padding of 2 pixels. The response from this layer is maxpooled using the P2 layer with an output size of 12 × 12 × 256. The output is then passed through three consecutive convolution layers without any maxpooling. The third convolution layer C3 has 384 filters of size 3 × 3, padding of 1 pixel and stride of 1 pixel with an output size of 12 × 12 × 384. The fourth convolution layer C4 has 384 filters of size 3 × 3, padding of 1 pixel and stride of 1 pixel. The output from this layer has a size of 12 × 12 × 384. The fifth convolution layer C5 has 256 filters of size 3 × 3, the same stride and padding as C3 and C4. The output from C5 has a size of 12 × 12 × 256 which is passed through a maxpooling layer P3 with 256 filters of size 3 × 3 and stride of 2 pixels with an output of 5 × 5 × 256.

The output from P3 is flattened through a flatten layer (FL) to a linear output of 6400 neurons before passing the results through three successive fully connected layers D1, D2 and D3. Each D1 and D2 have 4096 neurons followed by a dropout layer with a dropout rate of 0.1. The last FL D3 has 5 neurons corresponding to the five fault classes with a softmax classification function.

To solve the problem of overfitting, a sequence of data augmentation steps was used. First, the input image was randomly rotated with a rotation factor of 0.2. The second step involves randomly flipping the image along the horizontal and vertical axis. The third data augmentation method applied random contrast of 0.1 to all the images before finally rescaling the image. Every input image has input pixels in the range [0, 255]. The rescaling layer transforms the image pixels to the range [0, 1] by multiplying by a scale = 1./255. Also, a dropout layer with a dropout rate of 0.1 was applied after each of the first two FLs to reduce overfitting.

To train the network, an Adam optimizer with a CCE loss function was used with a minimum batch size of 32 at a learning rate of 0.0001. The maximum number of epochs for training was 100. The entire network has 46,767,493 trainable parameters.

4 Results and discussion

4.1 Performance analysis of the proposed vibration data denoising method

In this study, vibration signals from an experimental bearing test rig were used to evaluate the performance and accuracy of the proposed bearing fault diagnosis method. In order to observe the effects of noise when the bearing subsystem is treated as part of a larger system, the samples of the vibration data from a real bearing test rig were supplemented with AWGN. The proposed denoising method in the preprocessing stage was used to demonstrate how the added noise can be removed before further processing. The variance of the input noise-added signal was used as the variance of the AWGN. SNR and RMSE were used to evaluate the effectiveness of the proposed denoising technique and the results compared with VMD and wavelet denoising methods in isolation.

The first part of the denoising step implores the VMD denoising method, which is an effective way to decompose the vibration signal into IMFs. The MIGA algorithm was used to obtain optimal parameters for VMD decomposition. The optimal values for the penalty factor α and the decomposition number K were 12 and 2500, respectively. VMD decomposition of a random normalized sample vibration signal with AWGN of segment length of 20,000 using the optimal parameters is shown in Fig. 11. It can be observed that most of the signal noise is distributed in the high-frequency

IMF components of VMD as indicated by the central frequencies shown in Fig. 12.

Using the IBD approach summarized in Algorithm 3 in Table 3, the maximum slope is $\theta_{\max} = 0.15$ at index $i_{\max} = 10$. Applying the technique depicted in Table 3 to select the IMFs for reconstruction, the first two IMFs (IMF_1 and IMF_2) will be eliminated, leaving IMF_3 and IMF_{12} to reconstruct the VMD-IBD denoised signal. The reconstructed signal is further denoised using WT resulting in a better-quality signal. Figure 13 shows the results of denoising using various methods compared to the proposed method. It should be noted that most of the noise was removed during the WT stage, which is far superior to the VMD-IBD method. When the noise-added vibration signal was first processed with the VMD-IBD denoising method before WT denoising, the quality of denoising was better than directly processing it with WT. This shows that the proposed denoising method improved the quality of the denoised signal, as evident from the close similarity between the raw normalized vibration signal and that using the proposed method, as shown in Fig. 13.

To investigate the quality of denoising using the proposed scheme, 1000 random segments of length 15,000 for each of the five classes were used. After applying the proposed denoising method to these samples, the SNR and RMSE values were average. The results for the noise-added data, VMD denoised data, WT denoising data, and VMD-IBD+WT denoised data are shown in Fig. 14. Figure 14 shows the average RMSE of the denoising effects using various denoising methods. The noise-added data had the largest RMSE for each class, while the lowest RMSE was observed for samples processed with the proposed VMD-IBD+WT denoising method.

The smaller the RMSE, the better the performance of the method. For all 5 classes, WT performed better than the VMD approach, but both performed less than the proposed method. Figure 14 shows the average SNR values of the denoising effect, further confirming the observations with RMSE. The proposed denoising method shows the largest SNR values compared to the other methods showing the high quality of the denoised signal.

The quality of denoising is evident from the time–frequency transforms (CWT) shown in Fig. 15, whereby the noise recorded in the noise-added data in Fig. 15 and the remaining noise from the VMD denoising method is completely removed, as shown in Fig. 15. The denoised signal using the proposed method results in a signal that is very close to the original normalized raw vibration signal.

4.2 Performance analysis of the proposed bearing fault diagnosis method

Several experiments were conducted to evaluate the performance of the proposed bearing fault diagnosis approach.

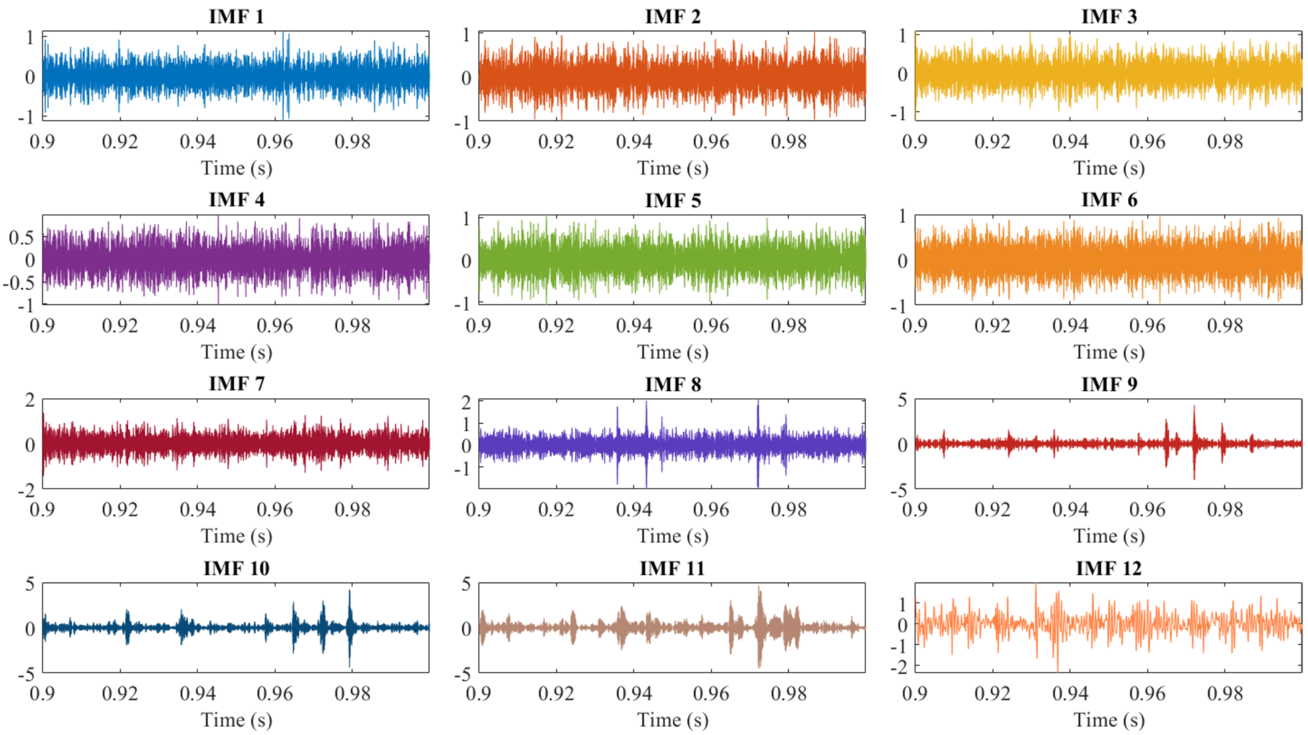


Fig. 11 Sample vibration signal decomposed using VMD with optimal parameters $K = 12$ and $\alpha = 2500$, showing 12 IMFs

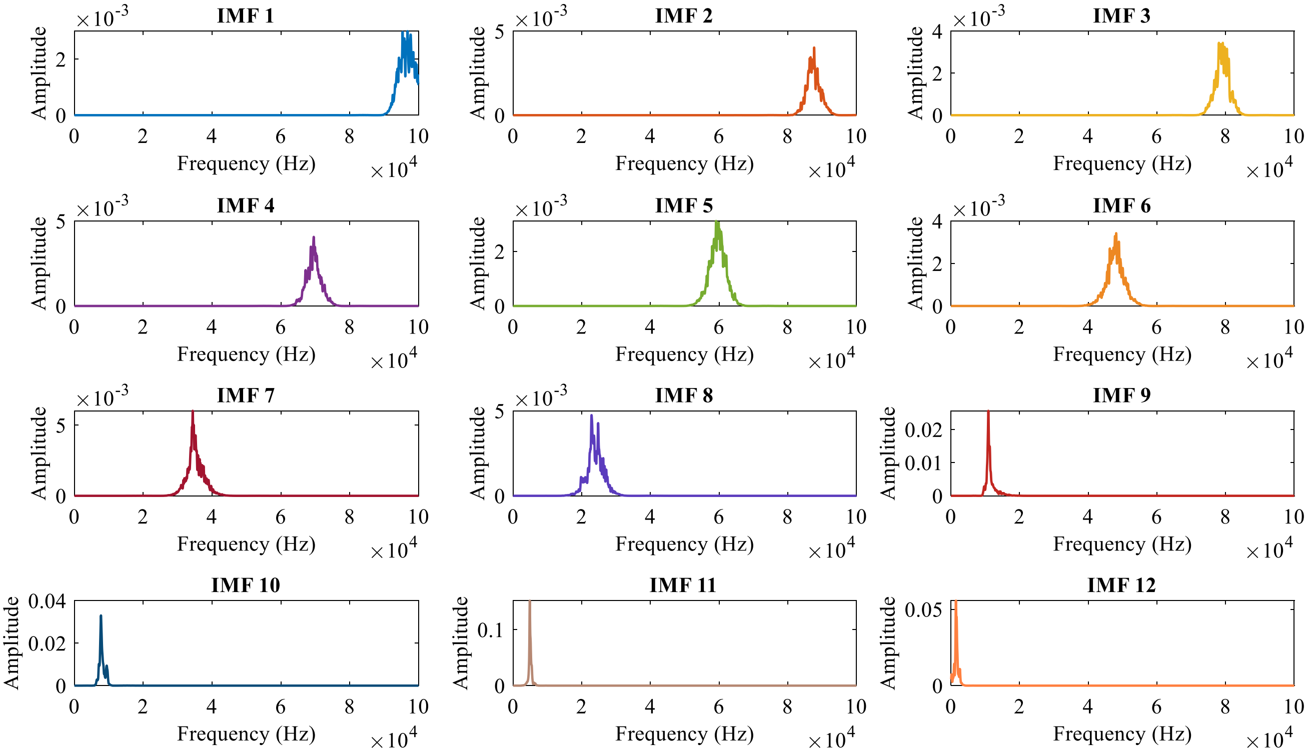


Fig. 12 Power spectrum of VMD decomposed IMFs showing the location of the central frequencies

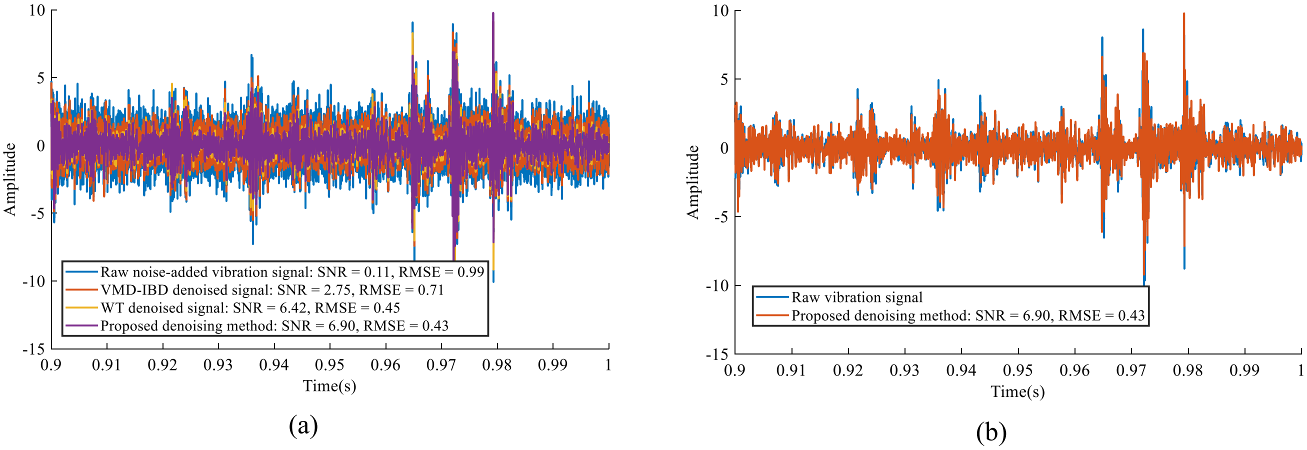
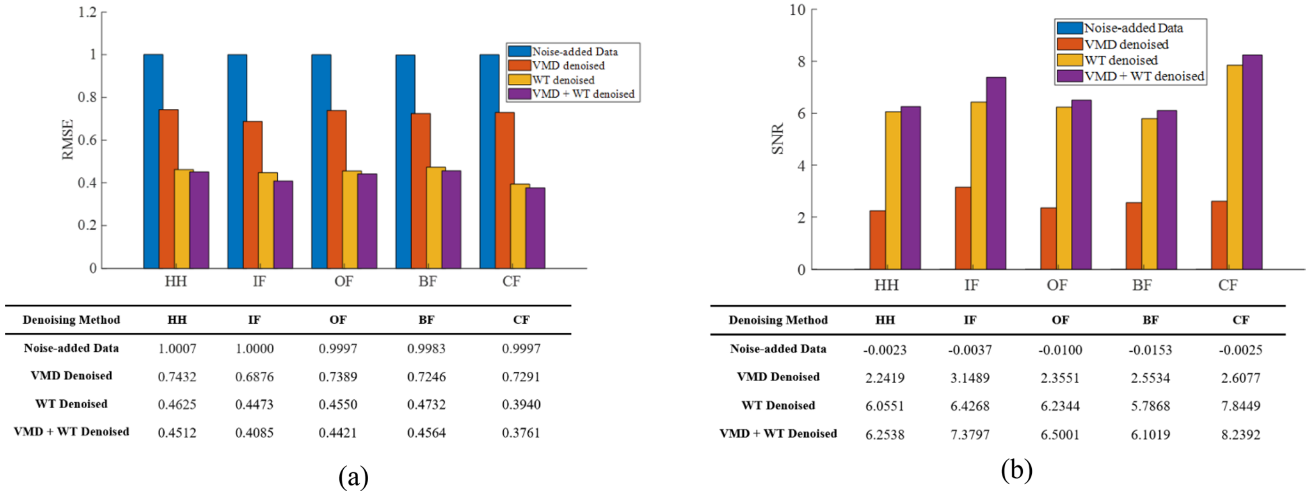


Fig. 13 **a** Comparison of various stages of signal pre-processing of a random inner race vibration signal. **b** Comparing the final denoised signal to the original normalized raw vibration signal sample with the highest SNR value and lowest RMSE value

Experiments were carried out on three separate groups of data generated from the same vibration dataset: original, noise-added, and denoised. Datasets with segments 2500, 5000, 7500, 10,000, 12,500 and 15,000 were generated for each category of datasets earlier mentioned. A total of five separate groups of experiments were carried out to evaluate the effectiveness of the method. Note that the fault classification results of each experiment were averaged over 5 runs to reduce the effects of randomness.

Experiments were conducted to evaluate how segment length affects classification results, using pre-defined lengths of 2500, 5000, 7500, 10,000, 12,500, and 15,000. The vibration samples were segmented, denoised, and converted into image data. Then, the deep learning model was used for fault classification. The results of the experiments are summarized in Table 5. More detailed results are presented in Tables 6, 7 and 8. The tables present the bearing fault classification results for different segment lengths using



Denoising Method	HH	IF	OF	BF	CF
Noise-added Data	1.0007	1.0000	0.9997	0.9983	0.9997
VMD Denoised	0.7432	0.6876	0.7389	0.7246	0.7291
WT Denoised	0.4625	0.4473	0.4550	0.4732	0.3940
VMD + WT Denoised	0.4512	0.4085	0.4421	0.4564	0.3761

Denoising Method	HH	IF	OF	BF	CF
Noise-added Data	-0.0023	-0.0037	-0.0100	-0.0153	-0.0025
VMD Denoised	2.2419	3.1489	2.3551	2.5534	2.6077
WT Denoised	6.0551	6.4268	6.2344	5.7868	7.8449
VMD + WT Denoised	6.2538	7.3797	6.5001	6.1019	8.2392

Fig. 14 Average performance metrics of the denoising effects on 1000 random samples of each fault type **a** RMSE, **b** SNR

the original, denoised, and noise-added datasets. It provides performance metrics such as precision, recall, and F1 scores for each bearing fault class. The results demonstrate the impact of segment length on the accuracy of the deep neural network.

Analyzing the results from all three tables, it can be observed that longer segment lengths generally lead to higher accuracy. Table 6 presents the original dataset's bearing fault classification results for different segment lengths. The validation accuracy for the original data increases from 99.46% for the data with a segment length of 2500 to 99.85% for that with a segment length of 15,000. Table 7 shows the classification results of bearing faults for various segment lengths using the denoised dataset. The results reveal that the denoised dataset achieves higher accuracy across various segment lengths than the noise-added datasets. High accuracies ranging from 98.28% for data with a segment length of 2500 to 99.70% for a segment length of 15,000 were observed for the denoised data. The training and validation accuracy of the denoised data is comparable to those of the original vibration data, with a percentage training accuracy error of 0.49% and a percentage validation error of 0.93%. This low error between the model performance for the original vibration data and the denoised vibration data shows that the denoising method effectively separates the vibration signal from the added noise. This indicates that the proposed denoising method effectively improves the quality of the vibration data, resulting in better fault classification performance. The precision, recall and F1 scores consistently indicate superior performance for all the bearing fault classes. This confirms the effectiveness of the proposed denoising method in removing noise from the vibration data, enabling the deep neural network to accurately classify bearing faults.

Table 8 presents the bearing fault classification results for different segment lengths using the noise-added dataset. The results in Table 8 shows that the accuracy of the model decreases when the raw vibration data is supplemented with white Gaussian noise. The precision, recall, and F1 scores indicate a significant decline compared to the original and denoised datasets. This highlights the adverse effect of noise on fault diagnosis accuracy, further emphasizing the importance of the denoising step in the proposed method. The presence of AWGN reduces the model’s performance to accurately classify the faults. The worse performance was observed for the 2500 segment with a 31.23% validation accuracy compared to 95.56% for the 15,000 segment length. So, even though the model performs poorly with the noise-added data, the accuracy improves with increased segment length since the images become more distinguishable at higher segment lengths.

The performance metrics are evaluated using Eqs. (53), (54), (55), and (56).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{58}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{59}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{60}$$

$$F1\_\text{score} = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{61}$$

Another significant observation is that the computational time drops significantly as the segment length increases. The number of segments directly equal to the sum of the number of training and validation images, decreases as the segment length increases. As a result, fewer iterations are required per

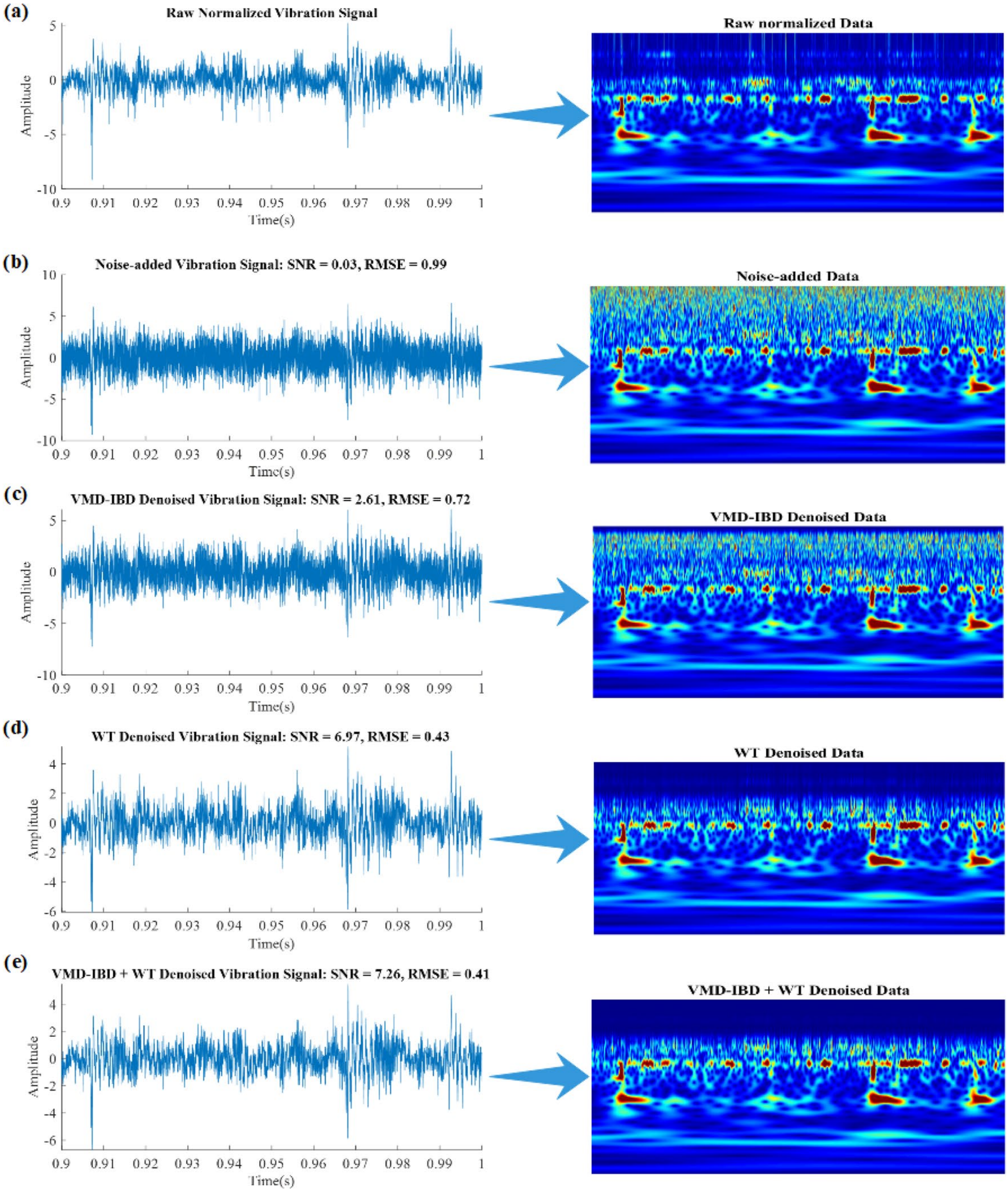


Fig. 15 Denoising results using the proposed method on ball fault bearings vibration signals. **a** Noisy vibration signal. **b** Original vibration signal without added white. **c** Denoised vibration signal using the proposed method

epoch due to the reduced sample size, hence a decrease in computation time observed across all three tables.

Experiments with signals under different speed conditions were conducted to evaluate the robustness and adaptive learning and classification capability of the proposed method. Four shaft speed conditions were tested: increasing, decreasing, increasing then decreasing, and decreasing then increasing. The bearing fault classification results for different operating speed conditions using the denoised dataset are presented in Table 9. It reports the training and validation accuracies and the computation time for each speed profile. It is observed that the increasing speed conditions generally have a slightly higher accuracy in both training and validation phases. This trend is consistent across different segment lengths. The higher accuracies achieved under increasing speed conditions indicate

Table 5 Comparison of bearing classification results for noise-added signal, original signal, and denoised signal

Vibration signal	Performance comparison for different segment lengths											
	2500		5000		7500		10,000		12,500		15,000	
	Train. acc. (%)	Valid. acc. (%)	Train. acc. (%)	Valid. acc. (%)	Train. acc. (%)	Valid. acc. (%)	Train. acc. (%)	Valid. acc. (%)	Train. acc. (%)	Valid. acc. (%)	Train. acc. (%)	Valid. acc. (%)
Noise-added	40.02	31.23	87.68	71.13	91.73	83.37	94.84	87.35	97.45	94.47	95.56	95.56
Original	99.71	99.46	99.91	99.57	99.81	99.91	99.77	99.13	99.99	99.55	99.37	99.85
denoised	98.28	96.09	99.44	98.55	99.58	99.24	99.38	99.25	99.80	99.35	99.57	99.70

acc. = accuracy, Train. = training, Acc. = accuracy

Table 6 Bearing fault classification results for different segment lengths (Original Dataset)

Bearing fault class	Bearing fault class performance metrics for a given segment length (%)																	
	2500			5000			7500			10,000			12,500			15,000		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
HB	98.28	99.02	98.65	98.75	99.04	98.89	99.58	100	99.79	95.81	99.79	97.76	100	97.77	98.87	100	99.24	99.62
IRF	98.30	99.03	98.66	98.85	99.12	98.98	99.54	100	99.77	96.02	99.80	97.88	100	97.79	98.88	100	99.26	99.63
ORF	98.32	99.04	98.68	98.79	99.06	98.92	99.55	100	99.77	96.13	99.81	97.93	100	97.76	98.87	100	99.24	99.62
BF	98.32	99.04	98.68	98.76	99.05	98.90	99.56	100	99.78	95.80	99.79	97.76	100	97.73	98.85	100	99.23	99.61
CF	98.31	99.03	98.67	98.79	99.07	98.83	99.58	100	99.79	96.05	99.80	97.89	100	97.85	98.79	100	99.27	99.64
Average	98.31	99.03	98.65	98.79	99.07	98.90	99.56	100	99.78	95.96	99.80	97.84	100	97.78	98.85	100	99.25	99.62
Comp. Time (mins)	49			34			23			18			11			6		
No. train. imgs	47,952			21,312			13,680			10,080			7968			6576		
No. test imgs	11,988			5328			3420			2520			1992			1644		
Train acc. (%)	99.71			99.91			99.81			99.77			99.99			99.37		
Valid. acc. (%)	99.46			99.57			99.91			99.13			99.55			99.85		

HB = healthy bearing, IRF = inner race fault, ORF = outer race fault, BF = ball fault, CF = combined fault

Table 7 Bearing fault classification results for different segment lengths (Denoised Dataset)

Bearing fault class	Bearing fault class performance metrics for a given segment length (%)																	
	2500			5000			7500			10,000			12,500			15,000		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
HB	91.67	89.93	90.79	97.82	96.44	97.12	96.99	99.44	98.20	96.59	99.59	99.06	97.04	99.75	98.38	99.12	99.41	99.26
IRF	91.55	89.79	90.66	97.96	96.68	97.32	96.75	99.39	98.05	96.76	99.61	98.16	97.13	99.75	98.42	99.02	99.35	99.18
ORF	91.83	90.11	90.96	97.87	96.52	97.19	96.79	99.40	98.08	96.83	99.62	98.21	97.04	99.75	98.37	99.11	99.40	99.26
BF	91.73	90.00	90.85	97.81	96.44	97.12	96.81	99.40	98.09	96.60	99.59	98.07	96.91	99.73	98.30	99.06	99.37	99.21
CF	91.44	89.65	90.53	97.85	96.49	97.17	96.94	99.43	98.17	96.77	96.61	98.17	97.16	99.76	98.44	99.15	99.43	99.29
Average	91.64	89.90	90.76	97.86	96.51	97.18	96.86	99.41	89.12	96.71	99.60	98.33	97.54	99.75	98.38	99.09	99.39	99.24
Comp. Time (mins)	47			35			24			18			10			6		
No. train. imgs	47,925			21,311			13,680			10,080			7968			6576		
No. test imgs	11,988			5327			3420			2520			1992			1644		
Train acc. (%)	98.28			99.44			99.58			99.38			99.80			99.57		
Valid. acc. (%)	96.09			98.55			99.24			99.25			99.35			99.70		

HB = healthy bearing, IRF = inner race fault, ORF = outer race fault, BF = ball fault, CF = combined fault

Table 8 Bearing fault classification results for different segment lengths (Noise-added Dataset)

Bearing fault class	Bearing fault class performance metrics for a given segment length (%)																	
	2500			5000			7500			10,000			12,500			15,000		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
HB	20.03	20.90	20.46	41.67	40.64	41.15	64.79	57.26	60.79	72.25	63.58	67.64	88.89	82.76	85.71	89.78	88.67	89.22
IRF	17.89	18.68	18.28	58.43	75.40	57.91	74.11	67.57	70.69	79.79	72.58	76.02	90.87	85.66	88.19	90.70	89.69	90.19
ORF	14.22	14.88	14.54	47.77	46.71	47.24	71.23	64.32	67.60	77.99	70.38	73.99	90.51	85.13	87.74	90.33	89.28	89.80
BF	13.32	13.95	13.63	48.41	47.75	48.27	68.29	61.05	64.47	76.52	68.60	72.34	88.94	82.83	85.78	89.93	88.83	89.38
CF	8.83	9.27	9.04	51.12	50.06	50.59	72.00	65.18	68.42	79.40	72.10	75.57	90.91	85.71	88.24	90.62	89.59	90.10
Average	14.86	15.54	15.19	49.48	52.11	49.03	70.08	63.08	66.39	77.19	69.45	73.11	90.02	84.42	87.13	90.27	89.21	89.74
Comp. time (mins)	48			33			24			17			11			9		
No. train. imgs	47,925			21,311			13,680			10,080			7968			6576		
No. test imgs	11,988			5327			3420			2520			1992			1644		
Train acc. (%)	40.02			87.68			91.73			94.84			97.45			95.56		
Valid. acc. (%)	31.23			71.13			83.37			87.35			94.47			95.63		

HB = healthy bearing, IRF = inner race fault, ORF = outer race fault, BF = ball fault, CF = combined fault, Comp. = computation, No. = number of, Train. = training, Imgs = images, Valid. = validation, Acc. = accuracy that the deep neural network can better capture the fault patterns present in the vibration data when the speed increases. On the other hand, decreasing speed conditions show slightly lower accuracy than increasing speed conditions. However, the difference in accuracy is relatively small, indicating a certain level of robustness of the model against variations in speed conditions. This suggests the trained deep neural network can generalize well and perform reasonably even when tested under different speed conditions. This could be attributed to the efficient capability of CWT to isolate the distinctive frequency features of each signal under variable conditions. There is also no effect on the computation time since it is largely affected by the size of the dataset. This result shows that the proposed method does not depend on the speed of operation of the shaft and can be used in any speed condition.

The experimental training and validation progress for the original, denoised, and noise-added vibration data are, respectively, shown in Figs. 16, 17 and 20. The poor performance of the noise-added data can be further ascertained by observing the training and validation progress in Fig. 20. In contrast, the original and denoised data have closely similar results. The confusion matrices for all considered segment lengths for the original and denoised data are also shown in Figs. 18 and 19, respectively. The confusion matrices show that the model can effectively classify the test images as opposed to the poor results depicted in Fig. 21 for the noise-added data. The performance of the model for the noise-added vibration data, as observed in the confusion matrices, shows that the presence of the noise significantly affects the accuracy of the results, as seen by the increased number of false positives and false negatives due to confusion between scalogram images of different fault types especially at smaller segment lengths (Figs. 20, 21).

The adaptive learning capability, robustness, and ability of the deep neural network model to generalize were further investigated with signals under different operation conditions for training and testing. Given that there are four operating speed conditions, the model is trained with images generated from vibration data with one operating speed condition and tested with images obtained from vibration signals acquired under the other three for all six segment lengths. The experiments were performed using the vibration data that has been denoised using the proposed denoising method. The bearing fault classification accuracies obtained under these scenarios and averaged over 5 runs are presented in Table 10.

The results show that the choice of training and testing speed conditions has a slight impact on the testing accuracies. Matching the speed conditions between training and testing generally leads to higher accuracies, indicating that consistency in the speed profile contributes to improved fault classification performance. For a segment length of

Table 9 Bearing fault classification results for different operating speed conditions (Denoised Dataset)

Speed profile	2500			5000			7500			10,000			12,500			15,000		
	Train. acc. (%)	Valid. acc. (%)	Comp. time (mins)	Train. acc. (%)	Valid. acc. (%)	Comp. time (mins)	Train. acc. (%)	Valid. acc. (%)	Comp. time (mins)	Train. acc. (%)	Valid. acc. (%)	Comp. time (mins)	Train. acc. (%)	Valid. acc. (%)	Comp. time (mins)	Train. acc. (%)	Valid. acc. (%)	Comp. time (mins)
Increasing	98.35	96.51	54	99.15	98.45	38	99.65	99.26	21	99.75	99.71	17	99.85	99.71	9	99.65	99.91	6
Decreasing	98.62	96.43	52	99.28	98.57	36	99.77	99.34	18	99.87	99.54	16	99.67	99.54	8	99.97	99.94	6
Increasing then decreasing	98.31	96.75	53	99.44	98.74	37	99.51	99.41	22	99.61	99.49	15	99.71	99.49	8	99.81	99.96	7
Decreasing then increasing	98.74	96.86	53	99.52	98.98	34	99.68	99.55	23	99.49	99.87	16	99.78	99.87	10	99.48	99.97	6

Train. = training, Acc. = accuracy, Comp. = computation

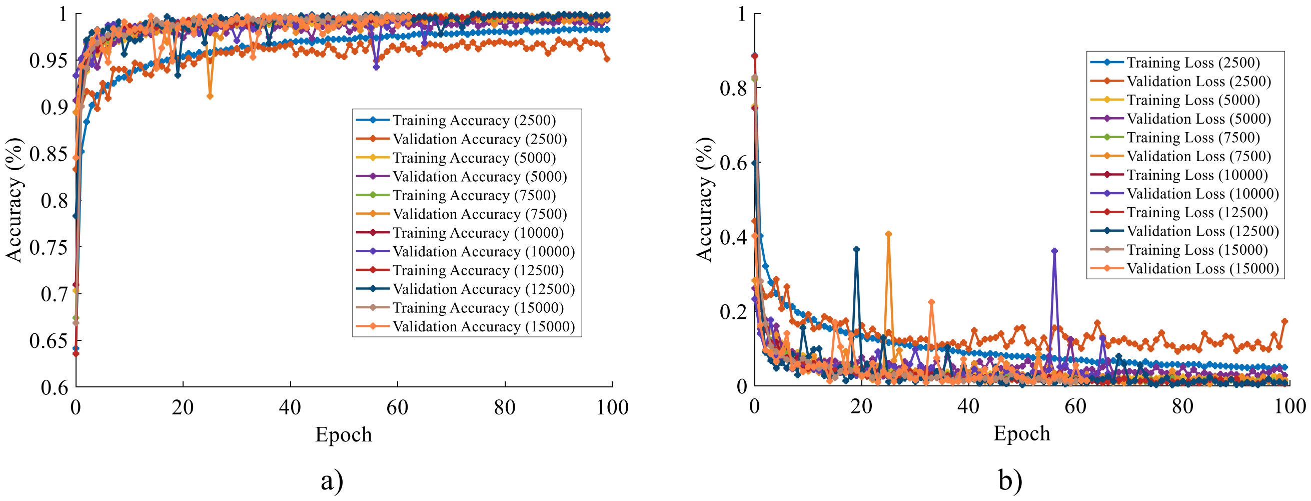


Fig. 16 Results for denoised data **a** Training and Validation accuracy curves. **b** Training and validation loss

2500, the highest testing accuracy of 97.44% is achieved when the model is trained with an increasing speed condition (Inc) and tested with a decreasing then increasing speed condition (DecTI). Similarly, for segment lengths of 5000, 7500, 10,000, 12,500, and 15,000, the highest testing accuracies are achieved with specific combinations of training and testing speed conditions. Interestingly, the testing accuracies are comparable across different conditions for some segment lengths. For example, for a segment length of 7500, the testing accuracies range from 99.22 to 99.88% across various training and testing speed conditions. This suggests that the model's performance is relatively robust to variations in speed conditions and capable of adapting and generalizing. It is important to note that while matching the speed conditions generally leads to higher accuracies, the models still exhibit reasonable accuracy even when tested under different speed conditions from that which they were trained. This indicates a certain level of adaptability and generalization capability of the deep neural network, enabling it to classify bearing faults effectively even in varying operational conditions.

4.3 Comparison with other popular deep learning and machine learning methods

Some popular deep-learning techniques for machinery fault diagnosis have been reported with very good performance. These techniques include CWT+CNN, deep neural networks (DNN), 1D convolutional neural networks (1D-CNN), and long short-term memory (LSTM). Experiments were carried out with these methods on the same dataset, and the results were compared with the proposed signal processing technique presented in this paper. Frequency spectra of the original raw vibration signal were used as input into the DNN and CNN networks, while the raw time series data was used as input to the LSTM network. The noise-added vibration data was directly converted to scalogram images using CWT, and the same CNN-based algorithm in the proposed framework was used. The structure of the DNN in each experiment contained an input layer, 4 hidden layers, and an output layer. The hyperparameters of the DNN network are shown in Table 11. Each CNN contained 3 convolutional-pooling layer units with ReLU activation functions. The convolutional kernel used in all CNNs is of dimension 5, and the max pooling stride of 2. A learning rate of 0.001 and a maximum epoch of 500 was set for both DNNs and CNNs training, while an 80–20% dataset split was used for training and validation datasets. The output layer of DNNs and CNNs has dimension 5, corresponding to the number of classes. Condition monitoring features such as peak value, crest factor, standard deviation, variance measure, kurtosis, skewness, shape factor, root mean square (RMS), entropy, impulse factor, margin factor, root mean square frequency (RMSF), and root variance frequency (RVF) calculated through spectral averaging method as described in [74, 75], were used for bearing fault diagnosis using support vector machine (SVM). The result using this manual feature extraction method is included in Table 11.

The comparison results presented in Table 11 show that the proposed bearing fault diagnosis method can more accurately identify bearing faults from noisy vibration signals than the DNN, 1D-CNN, LSTM, and SVM methods. This shows that the denoising technique makes it easier for the deep learning structure to accurately identify the faults since the frequency features of the vibration signals are not easily mixed up and confused with the noisy components. Compared to other signal processing methods carried out on the same dataset, the proposed method presented in this paper combines signal

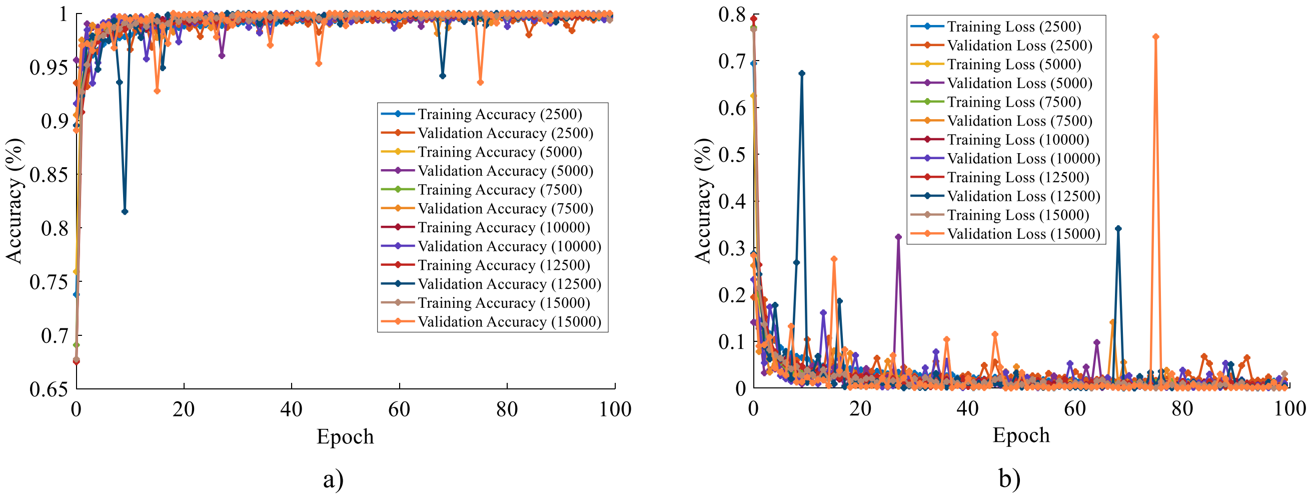


Fig. 17 Results for original data **a** Training and Validation accuracy curves. **b** Training and validation loss

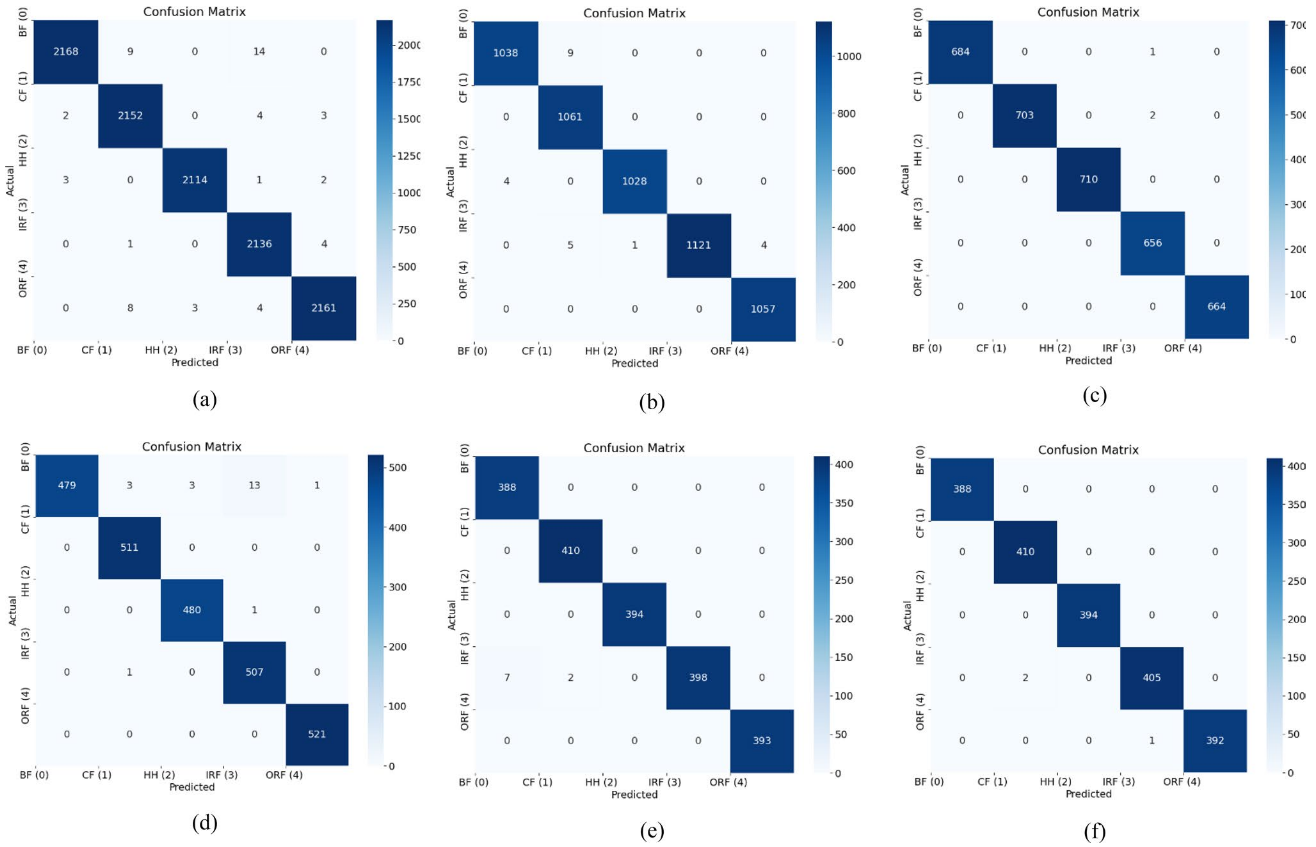
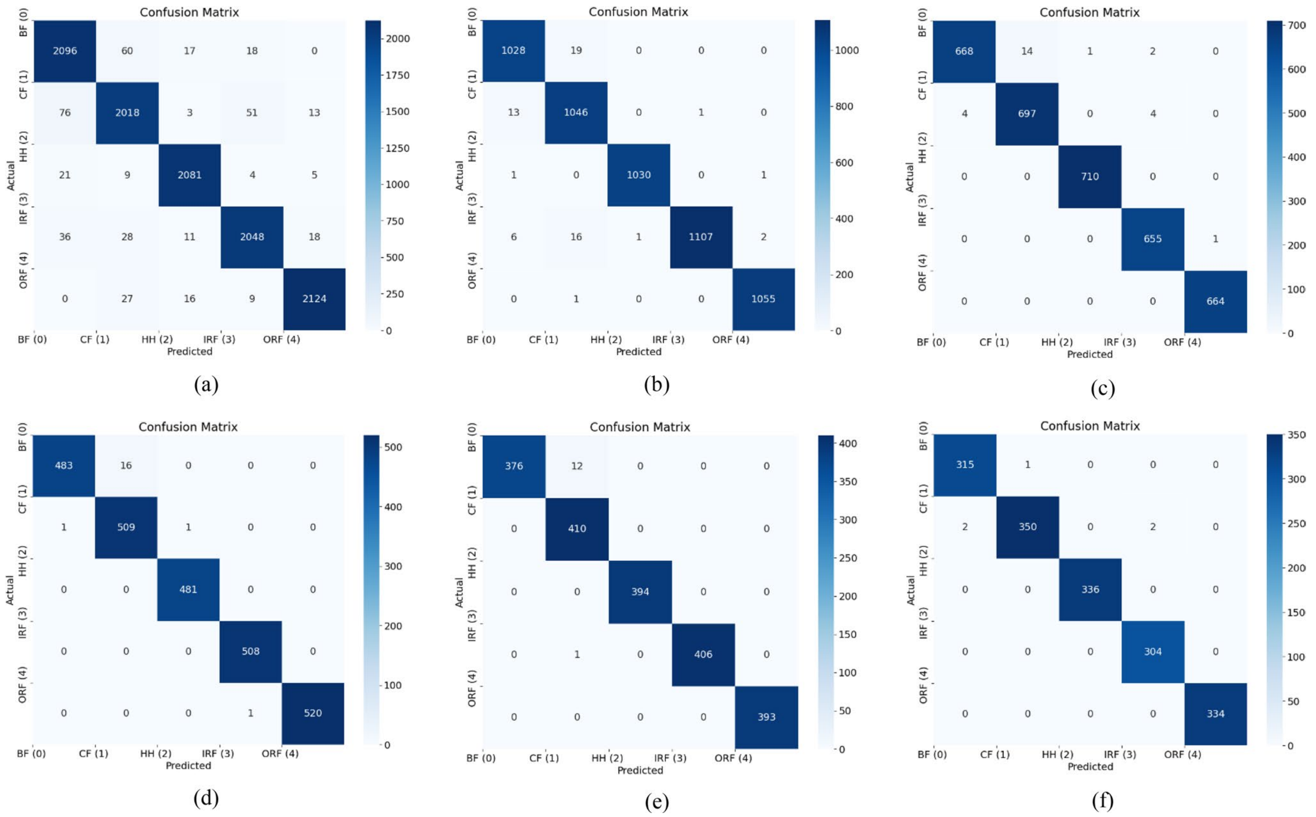


Fig. 18 Confusion matrix for bearing fault detection results for different segment lengths for the original vibration signal **a** 2500, **b** 5000, **c** 7500, **d** 10,000, **e** 12,500

denoising techniques with deep learning based on CNN architecture for feature extraction and fault classification. By so doing, the proposed method presents several advantages over other methods:

(1) It filters out low and high-frequency components and white noise from the data based on the optimized parameters, advanced denoising, and WT techniques. This method reduces the noise level in the acquired



(a) (b) (c)

(d) (e) (f)

Fig. 19 Confusion matrix for bearing fault detection results for different segment lengths for the denoised vibration signal **a** 2500, **b** 5000, **c** 7500, **d** 10,000, **e** 12,500

signal with little changes to the original features of the actual vibration signal profile of the healthy or faulty bearings. This significantly improves the fault diagnosis accuracy.

(2) By using deep learning, the fault diagnosis is automated as opposed to other unreliable manual techniques as they depend on visual inspection or statistical analysis.

(3) Due to the time-varying variable speed condition considered in this study, significant denoising and CWT technique decompose the input signal into a time–frequency representation, making it good for time and frequency localization. Combined with the power of deep learning, a powerful tool is developed to effectively identify subtle changes in the vibration signal due to existing faults.

(4) The weights obtained during deep learning represent changes in pixel depth in the scalogram images, which corresponds to time–frequency characteristics of the vibration signal and can be correlated to the fault features of the bearings. By applying denoising techniques, the effects of identical noise levels that can create confusion in distinguishing the frequency peaks in the time–frequency representation are limited, thereby improving the accuracy of the results. This makes the proposed method effective for bearing fault diagnosis and bearing health monitoring.

LSTM is also based on deep learning but has the lowest accuracy in diagnosing the bearing faults from the raw time-series signals. This could be attributed to the fact that the network cannot accurately identify the distinguishing features from the signals. Also, the classification accuracies from SVM with the manually extracted features through spectral average-based method is significantly lower than that of the proposed method and other deep learning methods, indicating that the proposed framework is more efficient for bearing diagnosis than the regular machine learning methods such as SVM on manually extracted features.

5 Conclusion

This paper presented a time-series signal processing method for denoising and bearing fault diagnosis using vibration signals. The proposed method combines several advanced time-series signal processing and denoising techniques with a deep learning structure for effective and efficient automatic vibration signal processing, feature extraction, and bearing fault diagnosis. The presented

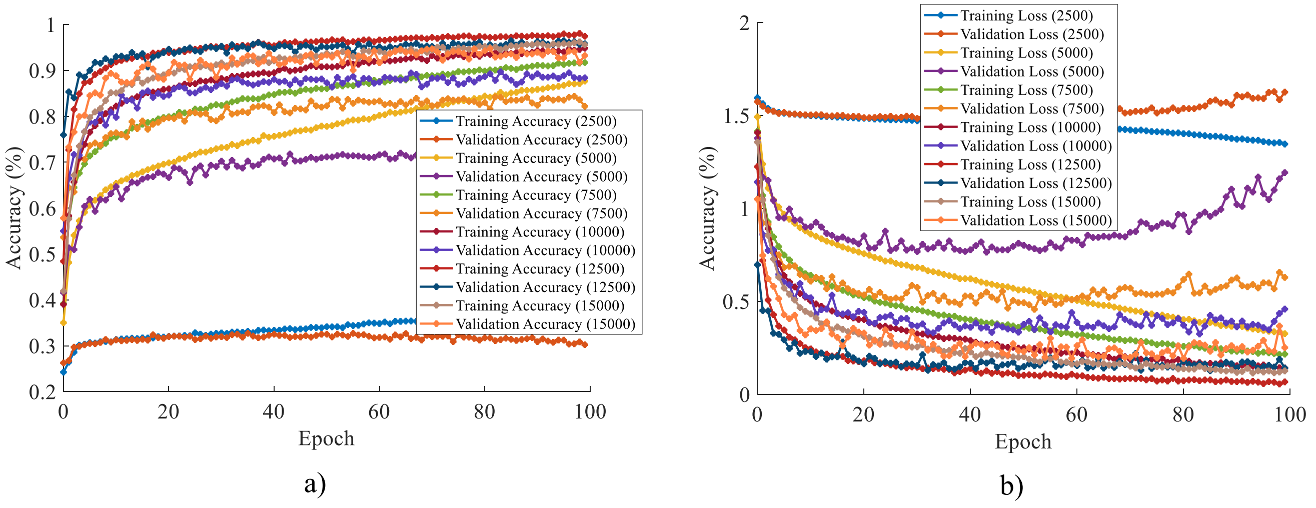


Fig. 20 Results for noise added data **a** Training and Validation accuracy curves. **b** Training and validation loss

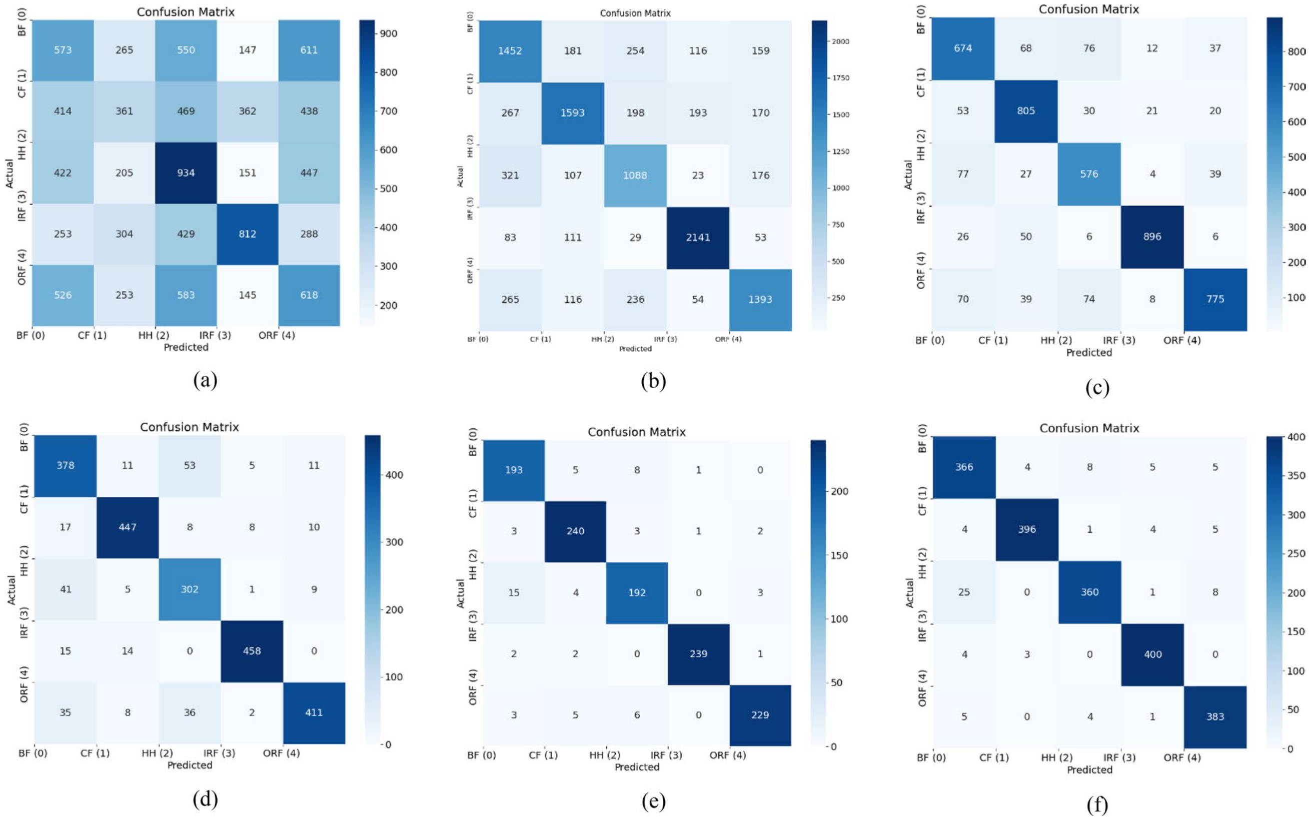


Fig. 21 Confusion matrix for bearing fault detection results for different segment lengths for the noisy vibration signal **a** 2500, **b** 5000, **c** 7500, **d** 10,000, **e** 12,500

method was divided into three crucial steps. First, the time series vibration signals are preprocessed and prepared. In this stage, the time series vibration signals are supplemented with AWGN to simulate the bearing subsystem as part of a larger industrial system whose vibration signals are considered the source of heavy noise. Then the noise-added signal is standardized and split into smaller segments of arbitrary length using an overlapping sliding window approach. Next, the noised added vibration signals are denoised using two successive denoising techniques:

Table 10 Bearing fault classification results for different training and testing conditions

Segment length	Training speed condition	Inc			Dec			IncTD			DecTI		
	Testing speed condition	Dec	IncTD	DecTI	Inc	IncTD	DecTI	Inc	Dec	DecTI	Inc	Dec	IncTD
2500	Testing accuracy (%)	96.86	96.34	97.10	96.23	96.57	96.88	97.12	96.91	96.85	97.44	97.13	96.97
5000	Testing accuracy (%)	98.99	98.76	98.58	98.73	98.39	98.78	98.90	98.61	99.01	98.31	98.92	99.08
7500	Testing accuracy (%)	99.24	99.67	99.53	99.49	99.66	99.50	99.88	99.71	99.22	99.25	99.57	99.43
10,000	Testing accuracy (%)	99.89	99.82	99.86	99.79	99.83	99.80	99.95	99.81	99.85	99.90	99.87	99.89
12,500	Testing accuracy (%)	99.97	100	99.86	99.89	99.76	99.96	99.89	99.69	99.57	99.83	99.92	99.37
1,5000	Testing accuracy (%)	100	99.83	99.68	99.99	99.84	99.43	99.97	99.87	99.64	99.96	100	99.74

Inc = increasing, Dec = decreasing, IncTD = increasing then decreasing, DecTI = decreasing then increasing

VMD-IBD with optimized parameters to remove low and high-frequency noise signals and then wavelet denoising to remove the same frequency noise signal to obtain a reconstructed vibration signal with less noise that resembles the original vibration signal as closely as possible. The final denoised vibration signal was processed using CWT to generate 2D time–frequency domain scalogram images. Finally, the generated image dataset was used to train a CNN deep neural network with a softmax classification layer to automatically identify the bearing fault.

The proposed method was validated using the Ottawa University vibration dataset from a bearing test rig. The validation results showed that the proposed method could diagnose bearing faults using vibration signals mixed with heavy noise signals through unsupervised learning with high accuracy and robustness. The results also showed that segment length and operating speed conditions have little or no significant effect on the accuracy of the results. It was established that the speed conditions for training and testing do not significantly affect the accuracy of the method. As a result, the model can be trained in one-speed conditions and used to identify faults under other speed conditions. By denoising, the noise-added vibration signal, the performance of the deep neural network to accurately identify the fault was significantly improved.

The proposed bearing fault diagnosis method was more accurate than other deep learning and traditional machine learning methods. The proposed bearing fault diagnosis method had the highest accuracy over other deep learning and traditional machine learning methods. It was able to remove the added white noise and traces of noise in the original vibration signal from the bearing test rig. The time–frequency transform of the denoised signal revealed important aspects of the faults, making it easier for the CNN to accurately classify the faults. The method proposed in the paper demonstrated its capability to extract “clean” vibration signals even in the presence of background noise and accurately classify and automatically diagnose the bearing faults. This method could be used to diagnose bearing faults in variable speed conditions and the presence of heavy background noise.

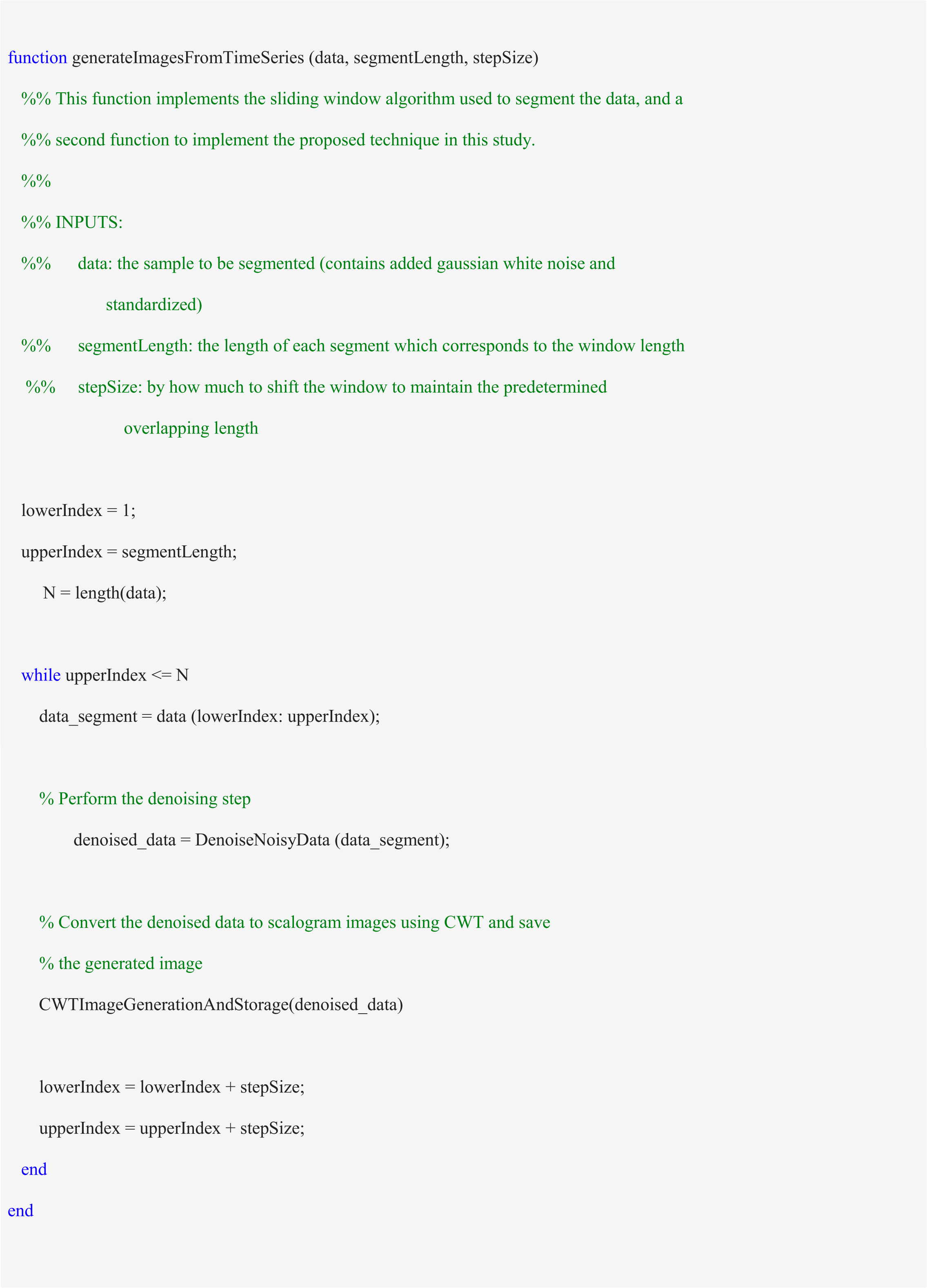
In future work, we will explore the effects of different parameter selections for the denoising stage on the accuracy of diagnosis. Optimizing the parameter selection process is key to improving the denoising process for better performance and robustness. One crucial limitation of this work is that it involves three crucial computationally intensive stages (denoising, time–frequency transformation, and deep learning model training). This limitation makes it difficult to

Table 11 Comparison of bearing fault classification results with other commonly used methods

Segment length	Number of images	Proposed method	Testing Accuracy (%)				
			CWT + CNN (Raw Noisy Signal)	DNN (Raw noisy signal)	LSTM (Raw noisy signal)	1D-CNN (Raw noisy signal)	SVM (Raw noisy signal)
2500	59,940	96.09	31.23	89.76 (2500|2000|1000|500|250|5)	45.70	67.78	80.50
5000	26,638	98.55	71.13	91.87 (5000|2500|1300|700|350|5)	44.80	68.54	82.30
7500	17,100	99.24	83.37	92.89 (7500|3750|2500|1300|650|5)	46.38	69.57	84.61
10,000	12,600	99.25	87.35	93.22 (10,000|5000|2500|1250|500|5)	47.50	68.82	86.10
12,500	9960	99.35	94.47	94.23 (12,500|6250|3125|1500|750|5)	45.89	68.33	88.20
15,000	8220	99.70	95.63	94.71 (1500|7500|3750|2500|1500|5)	46.30	68.98	88.40

Raw Noisy Signal = Original bearing fault signals with added white Gaussian noise use the proposed method for on-the-fly bearing fault diagnosis in real-time applications. Despite these limitations, the proposed method has proven its effectiveness, accuracy, and robustness in many applicable scenarios.

Appendix A



```
function generateImagesFromTimeSeries (data, segmentLength, stepSize)
  %% This function implements the sliding window algorithm used to segment the data, and a
  %% second function to implement the proposed technique in this study.
  %%
  %% INPUTS:
  %%      data: the sample to be segmented (contains added gaussian white noise and
                standardized)
  %%      segmentLength: the length of each segment which corresponds to the window length
   %%     stepSize: by how much to shift the window to maintain the predetermined
                   overlapping length

  lowerIndex = 1;
  upperIndex = segmentLength;
      N = length(data);

  while upperIndex <= N
     data_segment = data (lowerIndex: upperIndex);

     % Perform the denoising step
           denoised_data = DenoiseNoisyData (data_segment);

     % Convert the denoised data to scalogram images using CWT and save
     % the generated image
     CWTImageGenerationAndStorage(denoised_data)

     lowerIndex = lowerIndex + stepSize;
     upperIndex = upperIndex + stepSize;
  end
end
```

Acknowledgements National Natural Science Foundation of China No. (51875266).

Author contributions HH: Conceptualization, Methodology, Software, Validation, Writing—original draft, Writing—review and editing, Visualization. LAN: Supervision, Data collection, Software, Conceptualization, Analysis, Writing—review and editing. ST: Supervision, Conceptualization, Review. JG: Supervision.

Declarations

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

References

1. Li J, Wang H, Wang X, Zhang Y (2020) Rolling bearing fault diagnosis based on improved adaptive parameterless empirical wavelet transform and sparse denoising. Measurement 152:107392. https://doi.org/10.1016/j.measurement.2019.107392
2. Jin Z, He D, Wei Z (2022) Intelligent fault diagnosis of train axle box bearing based on parameter optimization VMD and improved DBN. Eng Appl Artif Intell 110:104713. https://doi.org/10.1016/j.engappai.2022.104713
3. Qian W, Li S, Lu J (2022) Adaptive nearest neighbor reconstruction with deep contractive sparse filtering for fault diagnosis of roller bearings. Eng Appl Artif Intell 111:104749. https://doi.org/10.1016/j.engappai.2022.104749
4. Cheng Y, Hu K, Wu J, Zhu H, Shao X (2021) A convolutional neural network based degradation indicator construction and health prognosis using bidirectional long short-term memory network for rolling bearings. Adv Eng Inform 48:101247. https://doi.org/10.1016/j.aei.2021.101247
5. Wei J, Huang H, Yao L, Hu Y, Fan Q, Huang D (2020) New imbalanced fault diagnosis framework based on cluster-MWMOTE and MFO-optimized LS-SVM using limited and complex bearing data. Eng Appl Artif Intell 96:103966. https://doi.org/10.1016/j.engappai.2020.103966
6. Schwendemann S, Amjad Z, Sikora A (2021) Bearing fault diagnosis with intermediate domain based layered maximum mean discrepancy: a new transfer learning approach. Eng Appl Artif Intell 105:104415. https://doi.org/10.1016/j.engappai.2021.104415
7. Jardine AKS, Lin D, Banjevic D (2006) A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Signal Process 20(7):1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012
8. Dyer D, Stewart RM (1978) Detection of rolling element bearing damage by statistical vibration analysis. J Mech Des 100(2):229–235. https://doi.org/10.1115/1.3453905
9. Collacott RA (1977) Mechanical fault diagnosis and condition monitoring. Springer, Dordrecht. https://doi.org/10.1007/978-94-009-5723-7
10. Tandon N, Choudhury A (1999) A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings. Tribol Int 32(8):469–480. https://doi.org/10.1016/S0301-679X(99)00077-8
11. Rai VK, Mohanty AR (2007) Bearing fault diagnosis using FFT of intrinsic mode functions in Hilbert-Huang transform. Mech Syst Signal Process 21(6):2607–2615. https://doi.org/10.1016/j.ymssp.2006.12.004
12. Lin H, Ye Y-C, Huang B-J, Su J-L (2016) Bearing vibration detection and analysis using enhanced fast Fourier transform algorithm. Adv Mech Eng. https://doi.org/10.1177/1687814016675080
13. Lin H-C, Ye Y-C (2019) Reviews of bearing vibration measurement using fast Fourier transform and enhanced fast Fourier transform algorithms. Adv Mech Eng 11(1):1687814018816751. https://doi.org/10.1177/1687814018816751
14. Randall RB (2021) Some special signal processing techniques. In: Vibration–based condition monitoring. Wiley, New York, pp 147–198. https://doi.org/10.1002/9781119477631.ch5
15. Saidi L, Ben Ali J, Fnaiech F (2014) Application of higher order spectral features and support vector machines for bearing faults classification. ISA Trans. https://doi.org/10.1016/j.isatra.2014.08.007
16. Randall RB (2021) Cepstrum analysis applied to machine diagnostics. In: Vibration–based condition monitoring. Wiley, New York, pp 199–229. https://doi.org/10.1002/9781119477631.ch6
17. Gao R, Yan R (2006) Non-stationary signal processing for bearing health monitoring. IJMR 1:18–40. https://doi.org/10.1504/IJMR.2006.010701
18. Wang WJ, McFadden PD (1993) Early detection of gear failure by vibration analysis I. Calculation of the time-frequency distribution. Mech Syst Signal Process 7(3):193–203. https://doi.org/10.1006/mssp.1993.1008
19. Quinde IR, Sumba JC, Ochoa LE, Guevara Jr. AV, Morales-Menendez R (2019) Bearing fault diagnosis based on optimal time-frequency representation method. IFAC-Pap 52(11):194–199. https://doi.org/10.1016/j.ifacol.2019.09.140
20. Zhou Y, Chen J, Dong GM, Xiao WB, Wang ZY (2011) Wigner–Ville distribution based on cyclic spectral density and the application in rolling element bearings diagnosis. Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci. 225(12):2831–2847. https://doi.org/10.1177/0954406211413215
21. Yan R, Gao RX, Chen X (2014) Wavelets for fault diagnosis of rotary machines: a review with applications. Signal Process 96:1–15. https://doi.org/10.1016/j.sigpro.2013.04.015
22. Peng ZK, Chu FL (2004) Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. Mech Syst Signal Process 18(2):199–221. https://doi.org/10.1016/S0888-3270(03)00075-X
23. Gao Q, Duan C, Fan H, Meng Q (2008) Rotating machine fault diagnosis using empirical mode decomposition. Mech Syst Signal Process 22(5):1072–1081. https://doi.org/10.1016/j.ymssp.2007.10.003
24. Han H, Cho S, Kwon S, Cho S-B (2018) Fault diagnosis using improved complete ensemble empirical mode decomposition with adaptive noise and power-based intrinsic mode function selection algorithm. Electronics 7(2):2. https://doi.org/10.3390/electronics7020016
25. Chen X, Ge D, Liu X, Liu M (2019) Roller bearing fault diagnosis based on empirical mode decomposition and targeting feature selection. IOP Conf. Ser. Mater. Sci. Eng. 630(1):012023. https://doi.org/10.1088/1757-899X/630/1/012023
26. Xuan Z, Ge M (2012) Application of the Hilbert–Huang transform for machine fault diagnostics. Appl Mech Mater 182–183:1484–1488. https://doi.org/10.4028/www.scientific.net/AMM.182-183.1484
27. Thakker HT, Dave V, Vakharia V, Singh S (2020) Fault diagnosis of ball bearing using Hilbert Huang transform and LASSO feature ranking technique. IOP Conf Ser Mater Sci Eng 841(1):012006. https://doi.org/10.1088/1757-899X/841/1/012006
28. Duong BP, Kim JY, Jeong I, Im K, Kim CH, Kim JM (2020) A deep-learning-based bearing fault diagnosis using defect signature wavelet image visualization. Appl Sci 10(24):24. https://doi.org/10.3390/app10248800

29. Gou L, Li H, Zheng H, Li H, Pei X (2020) Aeroengine control system sensor fault diagnosis based on CWT and CNN. Math Probl Eng 2020:e5357146. https://doi.org/10.1155/2020/5357146
30. Gao D, Zhu Y, Wang X, Yan K, Hong J (2018) A fault diagnosis method of rolling bearing based on complex Morlet CWT and CNN. In: 2018 Prognostics and system health management conference (PHM-Chongqing), pp 1101–1105. https://doi.org/10.1109/PHM-Chongqing.2018.00194
31. Xu Y, Li Z, Wang S, Li W, Sarkodie-Gyan T, Feng S (2021) A hybrid deep-learning model for fault diagnosis of rolling bearings. Measurement 169:108502. https://doi.org/10.1016/j.measurement.2020.108502
32. Neupane D, Kim Y, Seok J (2021) Bearing fault detection using scalogram and switchable normalization-based CNN (SN-CNN). IEEE Access 9:88151–88166. https://doi.org/10.1109/ACCESS.2021.3089698
33. Xiao Q, Li S, Zhou L, Shi W (2022) Improved variational mode decomposition and CNN for intelligent rotating machinery fault diagnosis. Entropy 24(7):908. https://doi.org/10.3390/e24070908
34. Zhang X, Wang H, Wu B, Zhou Q, Hu Y (2022) A novel data-driven method based on sample reliability assessment and improved CNN for machinery fault diagnosis with non-ideal data. J Intell Manuf. https://doi.org/10.1007/s10845-022-01944-x
35. Huang NE et al (1998) The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. Proc R Soc Lond Ser Math Phys Eng Sci 454(1971):903–995. https://doi.org/10.1098/rspa.1998.0193
36. Dragomiretskiy K, Zosso D (2014) Variational mode decomposition. IEEE Trans Signal Process 62(3):531–544. https://doi.org/10.1109/TSP.2013.2288675
37. Wu Z, Huang NE (2011) Ensemble empirical mode decomposition: a noise-assisted data analysis method. Adv Adapt Data Anal. https://doi.org/10.1142/S1793536909000047
38. Shannon CE (1949) Communication in the presence of noise. Proc IRE 37(1):10–21. https://doi.org/10.1109/JRPROC.1949.232969
39. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3:1–122. https://doi.org/10.1561/2200000016
40. Bertsekas DP (1995) Nonlinear programming. Athena Scientific, Belmont
41. Variational mode decomposition - MATLAB vmd. https://www.mathworks.com/help/wavelet/ref/vmd.html. Accessed 04 Apr 2023
42. Chen Q, Chen J, Lang X, Xie L, ur Rehman N, Su H (2021) Self-tuning variational mode decomposition. J Frankl Inst 358(15):7825–7862. https://doi.org/10.1016/j.jfranklin.2021.07.021
43. Tang G, Wang X (2015) Parameter optimized variational mode decomposition method with application to incipient fault diagnosis of rolling bearing. J Xi'an Jiaotong Univ 49:73–81. https://doi.org/10.7652/xjtuxb201505012
44. Dou D, Li L, Zhao Y (2011) Fault diagnosis of rolling bearings using EEMD-Renyi entropy and PCA-PNN. J Southeast Uni Nat Sci Edit 41:107–111. https://doi.org/10.3969/j.issn.1001-0505.2011.S1.023
45. Zhang L, Song C, Wang C, Xu T, Yi J (2019) Bearing performance degradation assessment based on a combination of multi-scale entropy and K-medoids clustering, pp 1–6. https://doi.org/10.1109/PHM-Qingdao46334.2019.8943049
46. Sen PK (1996) Anil Kumar Bhattacharyya (1915–1996): a reverent remembrance. Calcutta Stat Assoc Bull 46(3–4):151–158. https://doi.org/10.1177/0008068319960301
47. Mallat S (1989) A theory of multiresolution signal decomposition: the wavelet representation. IEEE Trans Pattern Anal Machine Intell 11:674–693. https://doi.org/10.1109/34.192463
48. Luo G, Zhang D (2012) Wavelet denoising. https://doi.org/10.5772/37424
49. Meyer Y (2002) Matlab help file. The Mathwork inc
50. Gonzalez R, Faisal Z (2019) Digital image processing second edition
51. Daubechies I (1992) Ten lectures on wavelets. Soc Ind Appl Math. https://doi.org/10.1137/1.9781611970104
52. Donoho DL, Johnstone IM (1994) Ideal spatial adaptation by wavelet shrinkage. Biometrika 81(3):425–455. https://doi.org/10.1093/biomet/81.3.425
53. Coifman RR, Donoho DL (1995) Translation-Invariant De-Noising. In: Antoniadis A, Oppenheim G (eds) Wavelets and statistics. Lecture notes in statistics. Springer, New York, pp 125–150. https://doi.org/10.1007/978-1-4612-2544-7_9
54. Donoho DL, Johnstone IM (1994) Ideal denoising in an orthonormal basis chosen from a library of bases. C R Acad Sci Sér I Math 319:1317–1322
55. Stéphane M (2009) Chapter 7—Wavelet bases. In: Stéphane M (ed) A wavelet tour of signal processing, 3rd edn. Academic Press, Boston, pp 263–376. https://doi.org/10.1016/B978-0-12-374370-1.00011-2
56. Johnstone IM, Silverman BW (1997) Wavelet threshold estimators for data with correlated noise. J R Stat Soc Ser B Methodol 59(2):319–351. https://doi.org/10.1111/1467-9868.00071
57. Johnson DH, Dudgeon DE (1992) Array signal processing: concepts and techniques. Simon & Schuster Inc, New York
58. Wang Y, He Z, Zi Y (2010) Enhancement of signal denoising and multiple fault signatures detecting in rotating machinery using dual-tree complex wavelet transform. Mech Syst Signal Process 24(1):119–137. https://doi.org/10.1016/j.ymssp.2009.06.015
59. Ng C-T (2014) On the selection of advanced signal processing techniques for guided wave damage identification using a statistical approach. Eng Struct 67:50–60. https://doi.org/10.1016/j.engstruct.2014.02.019
60. Tary JB, Herrera RH, van der Baan M (2018) Analysis of time-varying signals using continuous wavelet and synchrosqueezed transforms. Philos Trans R Soc Math Phys Eng Sci 376(2126):20170254. https://doi.org/10.1098/rsta.2017.0254
61. Fast Fourier Transform—an overview | ScienceDirect Topics. https://www.sciencedirect.com/topics/engineering/fast-fourier-transform. Accessed 10 Sep 2022
62. Zhou S, Tang B, Chen R (2009) Comparison between non-stationary signals fast Fourier transform and wavelet analysis. In: 2009 International Asia symposium on intelligent interaction and affective computing, pp 128–129.https://doi.org/10.1109/ASIA.2009.31
63. Wavelets and wavelet transform systems and their applications. https://doi.org/10.1007/978-3-030-87528-2. Accessed 12 Sep 2022
64. Lu C, Wang Z, Zhou B (2017) Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. Adv Eng Inform 32:139–151. https://doi.org/10.1016/j.aei.2017.02.005
65. Gu J, Peng Y, Lu H, Chang X, Chen G (2022) A novel fault diagnosis method of rotating machinery via VMD, CWT and improved CNN. Measurement 200:111635. https://doi.org/10.1016/j.measurement.2022.111635
66. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE conference on computer vision and pattern recognition, pp 580–587.https://doi.org/10.1109/CVPR.2014.81
67. Turay T, Vladimirova T (2022) Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: a survey. IEEE Access 10:14076–14119. https://doi.org/10.1109/ACCESS.2022.3147495

68. Alzubaidi L et al (2021) Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J Big Data 8(1):53. https://doi.org/10.1186/s40537-021-00444-8
69. IBM Cloud Education (2021) What are convolutional neural networks? Convolutional neural networks. https://www.ibm.com/cloud/learn/convolutional-neural-networks. Accessed 12 Oct 2022
70. Jin X, Xu C, Feng J, Wei Y, Xiong J, Yan S (2016) Deep learning with S-shaped rectified linear activation units. Proc AAAI Conf Artif Intell 30(1):1. https://doi.org/10.1609/aaai.v30i1.10287
71. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15:1929–1958
72. Huang H, Baddour N (2018) Bearing vibration data collected under time-varying rotational speed conditions. Data Brief 21:1745–1749. https://doi.org/10.1016/j.dib.2018.11.019
73. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. Commun ACM 60(6):84–90. https://doi.org/10.1145/3065386
74. Magar R, Ghule L, Li J, Zhao Y, Farimani AB (2021) FaultNet: a deep convolutional neural network for bearing fault classification. IEEE Access 9:25189–25199. https://doi.org/10.48550/arXiv.2010.02146
75. Van Hecke B, He D, Qu Y (2014) On the use of spectral averaging of acoustic emission signals for bearing fault diagnostics. J Vib Acoust 136(6):061009. https://doi.org/10.1115/1.4028322

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