IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 May 2024, accepted 29 May 2024, date of publication 3 June 2024, date of current version 10 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3408628

RESEARCH ARTICLE

Enhancing Rolling Bearing Fault Diagnosis in Motors Using the OCSSA-VMD-CNN-BiLSTM Model: A Novel Approach for Fast and Accurate Identification

YONG CHANG[1,2] AND GUANGQING BAO[3]

[1]College of Electronic and Electrical Engineering, Lanzhou Petrochemical University of Vocational Technology, Lanzhou 730060, China
[2]College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou 730050, China
[3]School of Electrical Engineering and Information, Southwest Petroleum University, Chengdu 610500, China

Corresponding author: Guangqing Bao (lzlgcyht2008@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51967012, in part by Gansu Youth Science and Technology Fund under Grant 22JR5RA807 and Grant 22JR11RA216, in part by the Natural Science Foundation of Gansu Province under Grant 23JRRA836, and in part by the Innovative Ability Enhancement Project of Gansu Provincial Higher Education under Grant 2023A-199.

ABSTRACT This study addresses the challenges posed by the strong noise and nonstationary characteristics of vibration signals to enhance the efficiency and accuracy of rolling-bearing fault diagnosis in electric motors. A fault diagnosis model is proposed based on improved variational mode decomposition (VMD) and a convolutional neural network bidirectional long short-term memory (CNN-BiLSTM). In the feature extraction stage, the Osprey-Cauchy-Sparrow search algorithm (OCSSA) optimizes the modal number K and penalty coefficient α of the VMD, facilitating the decomposition and reconstruction of the original vibration signals to extract fault features based on the minimum envelope entropy criterion. In the fault diagnosis stage, the mean, variance, peak value, kurtosis, RMS value, peak-to-average ratio (PAR), impulse factors, form factor, and clearance factor were computed from the reconstructed signals. These indicators were used to construct a feature vector for each sample, serving as the input for the OCSSA-VMD-CNN-BiLSTM fault diagnosis model, which quickly and accurately identifies the fault types. Experimental verification confirms that this method enhances the speed and accuracy of rolling-bearing fault identification compared to traditional approaches.

INDEX TERMS Fault diagnosis, variational mode decomposition (VMD), osprey optimization algorithm (OOA), sparrow search algorithm (SSA), bidirectional long short-term memory (BiLSTM).

I. INTRODUCTION

Motors are widely used in various industries as critical devices to convert electrical energy into mechanical energy. Rolling bearings are susceptible to wear, corrosion, and damage when operating under harsh conditions such as high and variable loads. Bearing failure accounts for more than 40% of all motor failures. If rolling bearing failures in motors are not detected promptly, the consequences can range from mild, such as equipment downtime, leading to economic losses to enterprises, to severe, such as threats to human health and life, resulting in negative social impacts. Therefore, it is important to conduct high-precision, intelligent, and low-latency rolling bearing fault diagnosis of rolling bearings. Establishing a reliable fault diagnosis system is essential to ensure the normal operation of motors [1], [2].

In recent years, research on rolling-bearing fault diagnosis has focused on three main areas: signal processing, machine learning, and deep learning. Signal processing techniques include time domain, frequency domain, and time-frequency analysis [3]. Owing to the nonlinear and non-stationary nature of rolling bearing fault vibration signals under complex

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo.

operating conditions, time-frequency analysis methods are often used for fault diagnosis. Notable examples include the short-time Fourier transform (STFT) [4], wavelet analysis [5], empirical mode decomposition (EMD) [6], variational mode decomposition (VMD) [7], and singular value decomposition (SVD) [8]. With a significant increase in the volume of mechanical equipment fault data, relying solely on the subjective expert analysis of signal characteristics can result in suboptimal diagnostic accuracy.

In addition, the continuous advancement of computer technology provides strong support for fault diagnosis, driving the development of rolling bearing fault diagnosis towards more intelligent and automated approaches. Machine learning autonomously learns the fault characteristics, thereby facilitating the diagnosis of rolling bearing faults. Commonly used machine learning algorithms include Support Vector Machines (SVMs) [9], Random Forests (RFs) [10], K-Nearest Neighbors (KNN) [11], and K-means clustering [12]. Although machine learning algorithms have been widely used, they have limited ability to handle nonlinear problems and are sensitive to noise and outliers, which are significant limitations in practical engineering applications. Deep learning, which employs a deep hierarchical structure to represent abstract features, has strong nonlinear representation capability. The relationships between the input and output features were determined through training.

Deep learning algorithms have been widely studied and applied in the diagnosis of rolling-bearing faults. Yu et al. [13] developed a knowledge-based Deep Belief Network (DBN) model by integrating the classification knowledge extracted from data into the classification layer of the DBN. This integration enhances the comprehensive and accurate extraction of bearing failure characteristics. However, the complexity of DBN structures and their tendency to converge to local optima during training limit their engineering applications. Liu et al. [14] proposed an improved Recurrent Neural Network (RNN) model using a Gated Recurrent Unit (GRU)-based autoencoder for signal denoising and deep feature extraction. This model effectively diagnosed the bearing faults under noisy conditions. Yan et al. [15] extracted health information using frequency domain features, which they combined with an enhanced Long Short-Term Memory (LSTM) network for further feature extraction, effectively predicting the remaining life of motor bearings. Wang et al. [16] improved the activation function of Stacked Autoencoders (SAE), addressing gradient vanishing problems and stabilizing model training, thereby improving the accuracy of motor bearing fault diagnosis. Liu et al. [17] used Variational Mode Decomposition (VMD) and Singular Value Decomposition (SVD) to obtain fault characteristics by employing a Convolutional Neural Network (CNN) for fault identification in planetary gears. Hou et al. [18] introduced Diagnosisformer, a novel attention-based multi-feature parallel fusion model for rolling bearing fault diagnosis, utilizing a transformer as the basic network. Tong et al. [19] addressed the signal distortion issues in noise reduction by proposing improved Deep Residual Shrinkage Networks (IDRSN) for fault diagnosis under noisy conditions. Li et al. [20] introduced a fault diagnosis method for rotating machines using Wasserstein Generative Adversarial Networks (WGAN) combined with a fully convolutional long short-term memory network. WGAN was used to generate high-quality data samples for training and fault diagnosis, effectively identifying various motor-bearing faults. Guo et al. [21] proposed an end-to-end fault diagnosis method using an attention-based Convolutional Neural Network and bidirectional LSTM (ACNN-BiLSTM), demonstrating strong generality and superiority over other advanced methods. Fu et al. [22] presented a fault diagnosis model incorporating Continuous Wavelet Transform (CWT), CNN with channel attention, BiLSTM, and a residual module to address the issues of insufficient feature extraction and poor noise resistance. Liu et al. [23] proposed a Siamese CNN-BiLSTM model that effectively addressed the significant impact of small and unbalanced fault samples on the diagnostic performance of models in locating and quantifying rolling-bearing damage in aircraft engines.

Based on the presented analysis, it can be concluded that the VMD algorithm effectively mitigates the problems related to endpoint effects and modal overlap. However, the quality of decomposition is strongly influenced by the number of decomposition layers and penalty factor. Machine learning and deep learning models are widely used in rolling-bearing fault diagnosis; however, their effectiveness is significantly limited by the choice of hyperparameters. Although heuristic algorithms offer a novel method for adaptive hyperparameter optimization, their tendency to fall into local optima can significantly hinder their effectiveness. To address these issues, this study proposes a fault diagnosis model based on improved VMD and CNN-BiLSTM. The model optimizes the VMD by integrating the Osprey-Cauchy-Sparrow Search Algorithm (OCSSA) to determine the optimal parameters for the VMD to effectively separate critical fault features from complex vibration signals. An OCSSA-VMD-CNN-BiLSTM hybrid neural network model was constructed to rapidly and accurately identify motor-bearing fault states, particularly when processing vibration data under complex noise conditions. The main contributions of this study are as follows: A Logistic chaos mapping strategy is employed to increase the diversity of the initial population of particles. In the first phase, the global exploration strategy uses the osprey optimization algorithm instead of the original sparrow algorithm's discoverer position update formula, addressing the excessive reliance of the sparrow algorithm on the previous food position by employing a random check to locate and attack the food source. The Cauchy mutation strategy was used to update the position of the followers in the sparrow algorithm, with individual perturbations applied during the sparrow position updates to expand the search scale and enhance the ability of the algorithm to escape local optima.

The experimental results validate that this method effectively improves the speed and accuracy of fault diagnosis, thereby providing a reference for handling complex engineering optimization problems.

II. VMD

Variational Mode Decomposition (VMD)is a non-recursive, quasi-orthogonal, adaptive signal decomposition algorithm that involves the construction and solution of variational problems [24], [25]. The variational constraint model is constructed as follows:

$$\begin{cases} \min\limits_{\{u_K\}\{\omega_K\}} \{\sum\limits_K \left\| \partial[(\delta(t)+\frac{\mathrm{j}}{\pi t})\cdot u_K(t)]e^{-\mathrm{j}\omega_K t}\right\|_2^2\} \\ s.t.\sum\limits_K u_K = f \end{cases} \tag{1}$$

where $\omega_K(t)$ is the center frequency corresponding to each mode, $\delta(t)$ is the Dirac distribution, and K is the number of modal components.

To convert the constrained variational problem into an unconstrained variational problem, an augmented Lagrangian function is introduced into the solution of the constrained variational model:

$$\begin{aligned} &L(\{u_K\},\{\omega_K\},\lambda) \\ &= \alpha\sum_K \left\| \partial_t[(\delta(t)+\frac{\mathrm{j}}{\pi t})\cdot u_K(t)e^{-\mathrm{j}\omega_K t}]\right\|^2 \\ &\quad + \left\| f(t)-\sum_K u_K(t)\right\|_2^2 + \left\langle \lambda(t), f(t)-\sum_K u_K(t)\right\rangle \end{aligned} \tag{2}$$

where α is the quadratic penalty factor that ensures the reconstruction accuracy of the signal. $f(t)$ under the influence of Gaussian noise, and $\lambda(t)$ is the Lagrange multiplier operator.

VMD algorithm steps:

1. Initialize $\{\hat{u}_K^1\}$, $\{\omega_K^1\}$, $\{\hat{\lambda}\}$ and n, $n=0$.
2. $n=n+1$ to start the whole loop.
3. Update the spectrum for each mode.

$$\hat{u}_K^{n+1}(\omega) = \frac{\hat{f}(\omega)-\sum\limits_{i\neq K}\hat{u}_i(\omega)-\hat{\lambda}(\omega)/2}{1+2\alpha(\omega-\omega_K)} \tag{3}$$

4.Frequency of updating centers.

$$\omega_K^{n+1} = \frac{\int_0^\infty \omega\left|\hat{u}_K(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{u}_K(\omega)\right|^2 d\omega} \tag{4}$$

5. Updating Lagrange multipliers.

$$\hat{\lambda}^{n+1}(\omega) \leftarrow \hat{\lambda}^n(\omega)+\tau[\hat{f}(x)-\sum_K \hat{u}_K^{n+1}(\omega)] \tag{5}$$

where τ is the Lagrange multiplier update parameter

6. Repeat steps 2 to 5 for a given discriminative accuracy $e>0$, until the iteration condition is satisfied:

$$\sum_K \left\|\hat{u}_K^{n+1}-\hat{u}_K^n\right\| \Big/ \left\|\hat{u}_K^n\right\|_2^2 < e \tag{6}$$

K narrowband eigenmode function components were obtained, and the iterations were completed. If the iteration condition is not satisfied, return to Step (2).

III. OSPREY-CAUCHY-SPARROW SEARCH ALGORITHM (OCSSA)

A. SPARROW SEARCH ALGORITHM (SSA)

In the SSA, the sparrow population is divided into discoverers and followers. Discoverers are individuals with higher fitness values who are responsible for leading the flock toward promising regions in the search space. In contrast, followers observe the movements of the discoverers and follow them to explore the search space. However, detection and warning mechanisms are often required because of the threats posed by natural enemies. In sparrow populations, discoverers with high fitness values are prioritized for food, mainly because they search for food and lead the movement of the entire population. Consequently, they can acquire food faster than the other sparrows. The equation for updating the positions of the discoverers is as follows.

$$X_{i,j}^{k+1} = \begin{cases} X_{i,j}^k\cdot\exp\left(\dfrac{-i}{\kappa\cdot \mathrm{Iter}_{\max}}\right), & R_2 < ST \\ X_{i,j}^k + Q\cdot\mathbf{L}, & R_2 \geq ST \end{cases} \tag{7}$$

where $X_{i,j}^k$ is the i^{th} sparrow in the j^{th} dimensional position at iteration k, $\kappa\in(0,1]$ is a random number, $\mathrm{Iter}_{\max}$ is the maximum number of iterations, $R_2\in[0,1]$ and $ST\in[0.5,1]$ are the warning value and safety value, respectively, Q is a random number obeying the standard normal distribution, and $\mathbf{L}$ is a matrix of $1\times d$ whose elements are all one matrix.

The positions of the followers are updated below:

$$X_{i,j}^{k+1} = \begin{cases} Q\cdot\exp\left(\dfrac{X_{\text{worst}}^k - X_{i,j}^k}{i^2}\right), & i > \dfrac{N_s}{2} \\ X_p^{k+1} + \left|X_{i,j}^k - X_p^{k+1}\right|\cdot\mathbf{A}^+\cdot L, & i \leq \dfrac{N_s}{2} \end{cases} \tag{8}$$

where X_{worst}^k is the current global worst position, N_s is the total number of sparrows, X_p is the optimal position of the discoverer, $\mathbf{A}$ is a matrix with $1\times d$, and the elements are randomized to 1 or -1, $\mathbf{A}^+ = \mathbf{A}^{\mathrm{T}}\left(\mathbf{A}\mathbf{A}^{\mathrm{T}}\right)^{-1}$.

For their own safety and to successfully obtain food, 10% to 20% of the total sparrow population is responsible for scouting vigilance, and their locations are updated as follows:

$$X_{i,j}^{k+1} = \begin{cases} X_{\text{best}}^k + \beta\left|X_{i,j}^k - X_{\text{best}}^k\right|, & f_i > f_g \\ X_{i,j}^k + \lambda\left(\dfrac{X_{i,j}^k - X_{\text{worst}}^k}{(f_i-f_w)+\varepsilon}\right), & f_i = f_g \end{cases} \tag{9}$$

where X_{best}^k is the global optimal position under the current iteration number k, β is the step length correction coefficient and obeys the standard normal distribution, f_i is the current fitness value of the sparrow, f_w and f_g represent the current overall worst fitness value and the optimal fitness value, respectively, $\lambda\in[0,1]$. $\varepsilon = 10E-60$.

B. LOGISTIC CHAOS MAPPING STRATEGY

Because SSA suffers from a loss of population diversity at the late stage of the optimization search, leading to local extremes, a logistic chaos mapping strategy is introduced to increase the population diversity and improve the global search ability of the algorithm [26].

Logistic chaotic mapping is a typical method for complex nonlinear behavior, and is formulated as follows:

$$x_{i,j}^{k+1} = \mu x_{i,j}^{k}(1 - x_{i,j}^{k}) \tag{10}$$

where μ is parameter that determines the map behavior. $x_{i,j}^{k+1}$ is the chaotic mapping between $x_{i,j}^{k}$.

C. CAUCHY MUTATION

During foraging, followers typically hunt around the optimal discoverer. Food fights may occur during this process, leading followers to compete and become new discoverers. A Cauchy variation strategy is introduced into the follower update formula to prevent the algorithm from falling into the local optima. This strategy enhanced the global search capability of the algorithm. The new positions of the followers are updated as follows.

$$X_{i,j}^{k+1} = X_{\text{best}}^{k} + cauchy(0, 1) \oplus X_{\text{best}}^{k} \tag{11}$$

where $cauchy(0, 1)$ is the standard Cauchy distribution function, and $\oplus$ indicates multiplication.

The Cauchy distribution function is a continuous probability distribution function used to describe certain types of random variable. The probability density function (PDF) is given by:

$$f(x; x_0, \gamma) = 1 \Big/ \pi\gamma[1 + (\frac{x - x_0}{\gamma})^2] \tag{12}$$

where x_0 is the location parameter representing the center of the distribution, and γ is the scale parameter representing the width of the distribution.

D. OSPREY OPTIMIZATION ALGORITHM (OOA)

The OOA consists of two phases: the first phase involves osprey determining the location of the fish and catching it (global exploration), and the second phase involves bringing the fish to a safe location to eat (local exploitation) [27].

1) POPULATION INITIALIZATION

The OOA is inspired by the hunting behavior of ospreys, using search and predation strategies to find optimal solutions for engineering problems. In the OOA, each osprey represents a potential solution with its position in the search space corresponding to the variable values of the problem. Each osprey is represented by a vector, with each element corresponding to a problem variable value. The algorithm searches the entire solution space to obtain the optimal solution. The osprey population was mathematically modeled as a matrix (Eq. (13)). Initially, the positions of the ospreys were randomly initialized using Eq. (14) to distribute them throughout the search space, thereby enhancing the search diversity.

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \\ \vdots \\ \mathbf{X}_N \end{bmatrix}_{N\times M} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,m} & \cdots & x_{1,M} \\ \vdots & \ddots & \vdots & \cdot & \vdots \\ x_{n,1} & \cdots & x_{n,m} & \cdots & x_{n,M} \\ \vdots & \cdot & \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,m} & \cdots & x_{N,M} \end{bmatrix} \tag{13}$$

$$x_{n,m} = lb_m + r_{n,m} \cdot (ub_m - lb_m) \tag{14}$$

where $n = 1, 2, \cdots, N, m = 1, 2, \cdots, M$, $\mathbf{X}$ is the population matrix of the osprey locations, $\mathbf{X}_n$ is the n^{th} osprey (candidate solution), $x_{n,m}$ is the m^{th} dimension (problem variable), N_o is the number of ospreys, M is the dimensionality of the problem variable, $r_{n,m} \in [0, 1]$ is a random number, lb_m and ub_m are the lower and upper bounds of the m^{th} problem variable, respectively.

The fitness value of each osprey item is calculated based on the corresponding objective function values. Eq. (15) represents the fitness values of all ospreys, which are used to assess the quality of each solution.

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}_1 \\ \vdots \\ \boldsymbol{F}_n \\ \vdots \\ \boldsymbol{F}_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} \boldsymbol{F}(\boldsymbol{X}_1) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{X}_n) \\ \vdots \\ \boldsymbol{F}(\boldsymbol{X}_N) \end{bmatrix}_{N\times 1} \tag{15}$$

where $\boldsymbol{F}$ is the vector of objective function values and $\boldsymbol{F}_n$ is the vector of the objective function values for the n^{th} osprey.

2) GLOBAL EXPLORATION

After locating the fish, the osprey launched an attack to catch them. In the OOA, this hunting behavior is modeled in the first stage of population renewal. Through simulation, the positions of the ospreys in the population were significantly changed, enhancing the exploratory ability of the algorithm to identify optimal regions and escape from local optima. In the OOA design, the set of locations for each osprey is determined according to Eq. (16).

$$\boldsymbol{FP}_n = \{\boldsymbol{X}_v | \, v \in \{1, 2, \cdots, N\} \wedge \boldsymbol{F}_k < \boldsymbol{F}_n\} \cup \{\boldsymbol{X}_{\text{best}}\} \tag{16}$$

where $\boldsymbol{FP}_n$ is the set of locations where fish are locked for the n^{th} osprey and $\boldsymbol{X}_{\text{best}}$ is the osprey with the best location.

The osprey randomly locate the position of the fish and initiate an attack. The movement of the osprey toward the fish was simulated, and the position of the osprey was calculated using Eqs. (17) and (18), respectively. If the new position improves the objective function value, the position of the osprey is updated according to Eq. (19).

$$x_{n,m}^{P1} = x_{n,m} + r_{n,m} \cdot (SF_{n,m} - I_{n,m} \cdot x_{n,m}) \tag{17}$$

$$x_{n,m}^{P1} = \begin{cases} x_{n,m}^{P1}, & lb_m \le x_{n,m}^{P1} \le ub_m \\ lb_m, & x_{n,m}^{P1} < lb_m \\ ub_m, & x_{n,m}^{P1} < ub_m \end{cases} \tag{18}$$

$$X_n = \begin{cases} x_n^{P1}, & F_n^{P1} < F_n \\ X_n, & else \end{cases} \tag{19}$$

where x_n^{P1} is the new position of the n^{th} osprey in the first stage, $x_{n,m}^{P1}$ is the new position of the n^{th} osprey in the first stage, in its m^{th} dimension, F_n^{P1} is the objective function in its m^{th} dimension, SF_n is the fish selected by the n^{th} osprey, $SF_{n,m}$ is its m^{th} dimension, $r_{n,m} \in [0, 1]$ is a random number, and $I_{n,m} \in \{1, 2\}$ is a random number.

3) LOCALIZED EXPLOITATION

In the OOA design, a new random position is computed for each individual in the population using Eq. (20) and (21), which represent the location of a suitable predatory fish. If the objective function value improves at this new position, the previous position of osprey is updated according to Eq. (22).

$$x_{n,m}^{P2} = x_{n,m} + \frac{lb_m + r \cdot (ub_m - lb_m)}{I_k} \tag{20}$$

where I_k is the iteration counter of the algorithm, $I_k = 1, 2, \cdots T$ and T is the total number of iterations.

$$x_{n,m}^{P2} = \begin{cases} x_{n,m}^{P2}, & lb_m \le x_{n,m}^{P2} \le ub_m \\ lb_m, & x_{n,m}^{P2} < lb_m \\ ub_m, & x_{n,m}^{P2} < ub_m \end{cases} \tag{21}$$

$$X_n = \begin{cases} x_n^{P2}, & F_n^{P2} < F_n \\ X_n, & else \end{cases} \tag{22}$$

where x_n^{P2} is the new position of the n^{th} osprey, $x_{n,m}^{P2}$ is the new position of the n^{th} osprey in the m^{th} dimension, F_n^{P2} is the objective function, SF_n is the fish selected by the n^{th} osprey, $SF_{n,m}$ is its m^{th} dimension, and $r_{n,m} \in [0, 1]$ is a random number.

E. OCSSA ALGORITHM FLOW

Step 1: Execute the logistic chaotic mapping algorithm to initialize the sparrow population. We set the population size P, maximum number of iterations $Iter_{max}$, proportion of discoverers PD, proportion of scouts SD, alert threshold R_2, security threshold ST and so etc.

Step 2: The individual fitness values of each sparrow were calculated and ranked to determine the global optimal and worst fitness values.

Step 3: Utilize Eqs. (17), (18), and (19) instead of Eq. (7) to update the discoverer location.

Step 4: The follower position update is performed using Eq. (11) instead of Eq. (8).

Step 5: Update the vigilante position according to Eq. (9).

Step 6: The sparrow population fitness value was calculated and reordered to update its position.

Step 7: Determine whether the current number of iterations reaches the maximum number of iterations or meets the solution accuracy requirement; If so, proceed to the next step otherwise; Otherwise skip to Step 2 until the end condition is met.

Step 8: The program ends and outputs the optimal fitness value and the best position. The flow of the OCSSA algorithm is illustrated in Fig. 1.

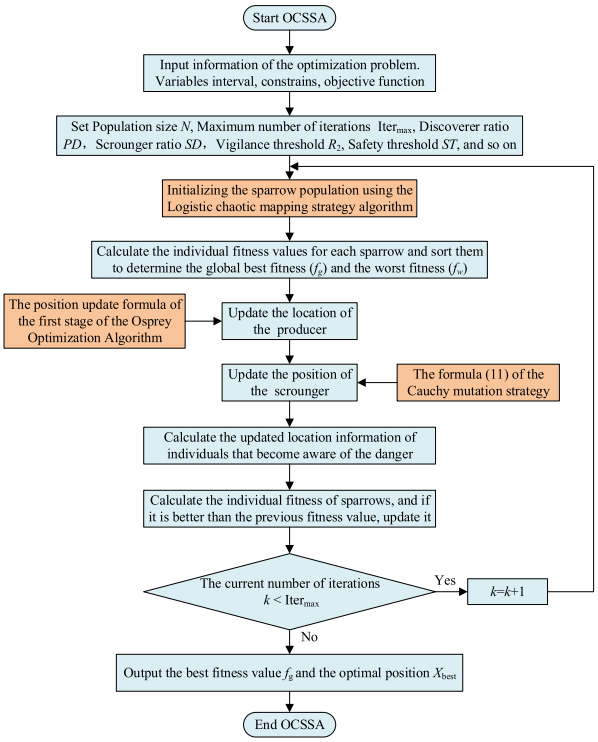


FIGURE 1. Flowchart of the proposed OCSSA.

IV. CNN-BiLSTM

A. LONG SHORT-TERM MEMORY (LSTM)

LSTM is a special recurrent neural network architecture proposed by Sepp Hochreiter and Jürgen Schmidhuber in 1997 [28]. Its internal structure consists of a memory cell, an input gate, a forget gate, and an output gate. The structure of the LSTM cell with gating is illustrated in Fig. 2. Ultimately, the combination of these two layers and the forget gate updates the LSTM storage cell according to Eq. (26).

$$\begin{cases} i_t = \sigma(w_{ix}x_t + w_{ih}h_{t-1} + b_i) \\ f_t = \sigma(w_{fx}x_t + w_{fh}h_{t-1} + b_f) \\ \tilde{c}_t = \tanh(w_{cx}x_t + w_{ch}h_{t-1} + b_c) \\ o_t = \sigma(w_{ox}x_t + w_{oh}h_{t-1} + b_o) \end{cases} \tag{23}$$

where i_t, f_t, and o_t are the input, forget, and output gates at time step t of an LSTM cell, respectively, $\tilde{c}_t$ is the candidate cell state characterizing the long-term memory at time step t, h_{t-1} is the output of the previous time step $t-1$, x_t is the input of the current time step t, b_i is the bias term of the input gate, σ is the sigmoid activation function, w_{ix} is the weight factor of the input gate x_t, w_{ih} is the weight factor of the previous hidden state h_{t-1}, b_f is the bias term of the forget

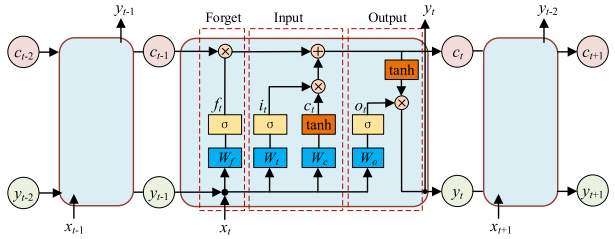


FIGURE 2. LSTM model.

gate, w_{fx} is the weight factor of the forgetting gate x_t, w_{fh} is the previous hidden state h_{t-1}, b_c is the bias term, w_{cx} is the weight coefficient of cell x_t, w_{ch} is the weight coefficient of cell h_{t-1}, b_o is the bias term of the output gate, w_{ox} is the weight coefficient of the output gate x_t, w_{oh} is the weight coefficient of the output gate h_{t-1}.

Write Eq. (23) in vector form as follows

$$\begin{bmatrix} i_t \\ f_t \\ o_t \end{bmatrix} = \sigma\left(\begin{bmatrix} w_{ix}w_{ih} \\ w_{fx}w_{fh} \\ w_{ox}w_{oh} \end{bmatrix} \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + \begin{bmatrix} b_i \\ b_f \\ b_o \end{bmatrix}\right) \tag{24}$$

$$\begin{bmatrix} \tilde{c}_t \end{bmatrix} = \tanh\left([w_{cx}w_{ch}] \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + [b_c]\right) \tag{25}$$

Eqs. (24) and (25) can be written as

$$\begin{cases} \mathbf{y}_1 = \sigma(\mathbf{w}_1\mathbf{x}_1 + \mathbf{b}_1) \\ \mathbf{y}_2 = \tanh(\mathbf{w}_1\mathbf{x}_1 + \mathbf{b}_1) \end{cases} \tag{26}$$

where $\mathbf{y}_1 = [i_t f_t o_t]^T$, $\mathbf{w}_1 = \begin{bmatrix} w_{ix}w_{fx}w_{ox} \\ w_{ih}w_{fh}w_{oh} \end{bmatrix}^T$, $\mathbf{x}_1 = [x_t h_{t-1}]^T$, $\mathbf{b}_1 = \begin{bmatrix} b_i b_f b_o \end{bmatrix}^T$, $\mathbf{y}_2 = \begin{bmatrix} \tilde{c}_t \end{bmatrix}$, $\mathbf{w}_2 = [w_{cx}w_{ch}]^T$, $\mathbf{x}_2 = [x_t h_{t-1}]^T$,, $\mathbf{x}_2 = [x_t h_{t-1}]^T$, and $\mathbf{b}_2 = [b_c]$.

The output of the LSTM cell for each time step is

$$\begin{cases} c_t = f_t * c_{t-1} + i_t * \tilde{c}_t \\ h_t = o_t \tanh(c_t) \end{cases} \tag{27}$$

where c_t is the cell state at time step t and c_{t-1} is the cell state from the previous time step $t-1$.

B. STRUCTURE OF CNN-BILSTM

The structure of the BiLSTM network is shown in Fig. 3, where x_t is the input data at time step t, $\vec{h}_t = (\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_n)$ is the output of the forward LSTM implicit layer at time step t, $\overleftarrow{h}_t = (\overleftarrow{h}_1, \overleftarrow{h}_2, \cdots, \overleftarrow{h}_n)$ is the output of the reverse LSTM implicit layer at time step t, $y_t = (y_1, y_2, \cdots y_n)$ is the output of the BiLSTM network at time step t, and the final output vector is the combined effect of both forward and reverse information flow$y_t = f(\vec{h}_t, \overleftarrow{h}_t)$. The mathematical model of the BiLSTM neural network system is as follows.

$$\begin{cases} \vec{h}_t = \sigma(w_{\vec{h}x}x_t + w_{\vec{h}\vec{h}}h_t + b_{\vec{h}}) \\ \overleftarrow{h}_t = \sigma(w_{\overleftarrow{h}x}x_t + w_{\overleftarrow{h}\overleftarrow{h}}h_t + b_{\overleftarrow{h}}) \\ y_t = w_{y\vec{h}}\vec{h}_t + w_{y\overleftarrow{h}}\overleftarrow{h}_t + b_y \end{cases} \tag{28}$$

The CNN-BiLSTM model combines the advantages of CNNs and BiLSTM for text categorization. As illustrated in

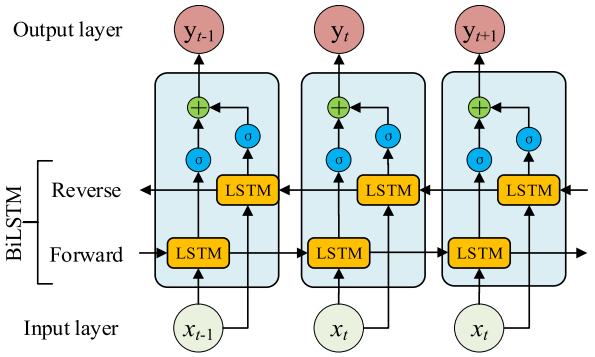


FIGURE 3. BiLSTM Structure.

Fig. 4, a CNN is employed to capture the local features of the input text, progressively reducing the size and amount of feature information through a series of convolutional and pooling layers. Subsequently, the BiLSTM network extracts global feature information from the CNN output by considering the overall structure and long-term dependencies of text. The forward and backward networks of BiLSTM separately process the output features of the CNN, retaining the key information of the text sequence through the memory unit and gating mechanism to obtain the final feature information. Finally, the BiLSTM output is passed to a fully connected layer for text categorization. By integrating the local feature extraction capability of the CNN with the global information processing ability of BiLSTM, the CNN-BiLSTM model effectively enhances the accuracy and efficiency of text classification.

V. TESTS AND COMPARISONS

A. OCSSA

In this study, we introduce the OCSSA algorithm and assess its performance by comparing it with nine other optimization algorithms, namely, the Sparrow Search Algorithm (SSA) [29], Dung Beetle Optimization Algorithm (DBO) [30], Subtractive Averaging Based Optimization Algorithm (SABO) [31], Artificial Gorilla Troop Optimization Algorithm (GTO) [32], Particle Swarm Optimization Algorithm (PSO) [33], Gray Wolf Optimization Algorithm (GWO) [34], Marine Predator Algorithm (MPA) [35], Raccoon Optimization Algorithm (COA) [36], and Sand Cat Swarm Optimization Algorithm (SCSO) [37]. These algorithms were selected based on their relevance and popularity in optimization.

To conduct a comprehensive evaluation, we employed CEC2005 test functions [38], which are widely recognized benchmarks in the optimization community. The test suite comprises ten classical functions with unimodal and multimodal functions, as detailed in Table 1. This classification allowed us to assess the performance of the algorithms across diverse optimization landscapes. The main parameter settings for each algorithm are listed in Table 2, ensure a fair and consistent comparison. Our experiments were conducted on a Windows 11 64-bit operating system with an Intel® Core™i7-11800H processor running at 2.30 GHz. All

TABLE 1. Test function.

Function	D	Initial range	Optimum
$f_1(x)=\sum_{i=1}^{n}x_i^2$	30	$[-100,100]^D$	0
$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i}^{n}\lvert x_i\rvert$	30	$[-10,10]^D$	0
$f_3(x)=\sum_{i}^{n}(\sum_{j-1}^{i}x_j)^2$	30	$[-100,100]^D$	0
$f_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	30	$[-100,100]^D$	0
$f_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	$[-5.12,5.12]^D$	0
$f_6(x)=\sum_{i=1}^{n}\left(\lfloor x_i+0.5\rfloor\right)^2$	30	$[-32,32]^D$	0
$f_7(x)=\sum_{i=1}^{n}ix_i^4+random[0,1]$	30	$[-100,100]^D$	0
$f_8(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	$[-10,10]^D$	0
$f_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$	30	$[-500,500]^D$	0
$f_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	$[-5,10]^D$	0

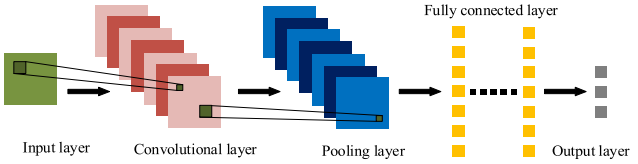


FIGURE 4. 1D CNN structure.

TABLE 2. Main parameter settings.

Algorithm	Parameter setting
SSA	PD=0.2, SD=0.1, R_2=0.8, ST=0.8
DBO	γ=0.1, k=0.1, u=0.3, s=0.5
SABO	i=1, t=1
GTO	p=0.03, β=3, w=0.8
PSO	c_1=c_2=1.5, w=0.8
GWO	a decreases linearly from 2 to 0, $r_1,r_2\in[0,1]$
MPA	P=0.5, EFIs=0.2
COA	a decreases linearly from 2 to 0
SCSO	SM=2
OCSSA	PD=0.2, SD=0.1, R_2=0.8, ST=0.8

algorithms were implemented in MATLAB 2020a using M language. By comparing the OCSSA algorithm with the nine established algorithms across the CEC2005 test functions, we aimed to demonstrate its effectiveness and efficiency in solving complex optimization problems. The results of this comparative analysis are presented in the following sections.

To ensure fairness of the comparison, we standardized the experimental conditions for all ten algorithms by setting the population number to $P = 100$, the number of dimensions $D = 30$, and the maximum number of iterations $\text{Iter}_{max} = 1000$. The additional parameter settings are listed in Table 2. The convergence performance of the algorithms under the CEC2005 test functions $f_1 \sim f_{10}$ is shown in Fig. 5. The convergence curves demonstrate the progress of the algorithms toward the theoretical optimal solution. When a curve ceases to progress with an increase in iterations, it indicates that the algorithm has reached the theoretical optimum. For single-peak test functions (Fig. 5(a) and (g)), the OCSSA algorithm exhibited superior convergence performance compared to the PSO, GWO, MPA, DBO, SABO, GTO, SSCSO, COA, and SSA algorithms. Notably, PSO, GWO, MPA, and COA exhibited stagnation in their convergence curves, indicating lower optimization search accuracy and a tendency to fall into local optima. In contrast, OCSSA significantly improves the convergence speed and accuracy, outperforming SSA and other algorithms in reaching a theoretically optimal solution. This improvement highlights the effectiveness of the OOA and Cauchy variants in escaping local optima and accelerating the search process. For the multipeak test functions (Fig. 5(h) and (j)), the OCSSA continues to show significant advantages in terms of both convergence speed and accuracy compared with the other algorithms. The rapid decline of its curve and achievement of the theoretical optimal solution by all algorithms underscore the superior performance of the OCSSA in complex landscapes. Regarding the complexity of the algorithm, the PSO algorithm had the lowest overall time consumption. However, the OCSSA algorithm requires more time owing to the introduction of an improved strategy that expands the SSA optimization range. Despite the increased time consumption, the optimization accuracy of the OCSSA was significantly enhanced. The findings presented in Fig. 5 and Table 3 verify the effectiveness of the proposed improvement strategy. The superior performance of the OCSSA on both single-peak and multipeak test functions and its ability to achieve high accuracy despite increased complexity demonstrate its potential as a powerful tool for task optimization.

B. OCSSA-VMD-CNN-BILSTM MODEL

1) EXPERIMENT 1: CWRU BEARING DATASET

The experimental data used in this study were sourced from the Case Western Reserve University (CWRU) database of motor bearing failures [39]. As shown in Fig. 6, a test bench

TABLE 3. Calculation time of ten algorithms for testing functions in ten.

Function	Time(s)									
	PSO	GWO	MPA	DBO	SABO	GTO	SSCSO	COA	SSA	OCSSA
f_1	0.217	0.370	0.602	0.292	1.255	0.769	4.325	1.250	0.341	0.808
f_2	0.227	0.371	0.623	0.307	1.282	0.770	4.332	1.252	0.351	0.816
f_3	0.728	0.910	1.739	0.869	1.861	1.864	5.139	1.853	0.996	1.483
f_4	0.220	0.366	0.614	0.301	1.238	0.771	4.333	1.278	0.358	0.814
f_5	0.280	0.438	0.746	0.364	1.337	0.873	4.370	1.369	0.424	0.900
f_6	0.222	0.369	0.590	0.295	1.244	0.761	4.359	1.297	0.342	0.814
f_7	0.543	0.707	1.317	0.657	1.600	1.426	4.767	1.600	0.753	1.245
f_8	0.280	0.445	0.744	0.383	1.380	0.888	4.516	1.357	0.424	0.935
f_9	0.261	0.415	0.676	0.350	1.324	0.806	4.470	1.712	0.396	0.877
f_{10}	0.249	0.381	0.645	0.334	1.266	0.817	4.351	1.336	0.401	0.902

TABLE 4. Division results of fault dataset.

Fault type	Normal	Inner ring fault			Rolling ball fault			Outer ring fault		
Fault label	1	2	3	4	5	6	7	8	9	10
Fault diameter (d/inch)	-	0.007	0.014	0.021	0.007	0.014	0.021	0.007	0.014	0.021
Training sample	90	90	90	90	90	90	90	90	90	90
Test sample	30	30	30	30	30	30	30	30	30	30
Sample data size	2048	2048	2048	2048	2048	2048	2048	2048	2048	2048

was employed to simulate the bearing failures with a vibration signal sampling frequency of 1024 Hz. The experiment utilized a deep-groove ball-bearing model SKF6205 with failures simulated via electrical discharge machining (EDM). For diagnostic purposes, ten fault categories were established, encompassing four types of states: normal bearing, rolling element fault, outer ring fault, and inner ring fault. Each fault state was further categorized based on fault diameters 0.007, 0.014, and 0.021 in, resulting in nine distinct states representing varying degrees of failure. The dataset for each fault type consists of 120 samples, with each sample containing 2048 data points. In the experimental setup, 90 samples were allocated for model training and 30 samples were reserved for model testing. The divisions of the fault dataset are presented in Table 4.

In the Variational Mode Decomposition (VMD) of the original vibration signals of rolling bearings, determining the relevant parameters is crucial. Specifically, the number of modal components K and penalty parameter α significantly influenced the decomposition results. The value of K determines the number of decomposed modal components. If K is set too low, under-decomposition occurs, leading to a loss of key information and an inability to extract essential features. Conversely, if K is set too high, it results in an excessive number of modal components, causing overlapping center frequencies and making signal feature differentiation difficult. The penalty parameter α primarily affects the bandwidth of each modal component with an appropriate value that enhances the accuracy of the reconstructed signal. In this study, we propose an approach that optimizes VMD using an improved sparrow search algorithm and combines it with a CNN-BiLSTM architecture for the fault diagnosis of motor bearings. This method effectively extracts fault features from the original vibration signals. After the feature extraction process, each of the ten states was represented by 120 samples, resulting in a 1200×9 matrix. Each row of this matrix is labeled with numbers 1-10 to indicate different fault types. The fault-diagnosis process implemented in this study is shown in Fig. 7.

Fig. 8 illustrates the convergence and parameter variation curves obtained during the optimization of the VMD parameters using the OCSSA. As shown in Fig. 8, the convergence process shows that the algorithm tends to stabilize after 14 iterations, indicating convergence. At this point, the minimum envelope entropy achieved is 4.7828, with the optimal parameter combinations for the number of decomposition modes (K) and penalty factor (α) being (5, 3508).

Fig. 9 illustrates the impact of the VMD parameters on the fault diagnosis accuracy, specifically focusing on the number of decomposition modes K and penalty factor α. From Fig. 9, it is evident that when K is too low, the accuracy is reduced owing to under-decomposition, implying that key signal features might be missed. Conversely, when K is too high, the accuracy again decreases owing to over-decomposition, which leads to overlapping center frequencies that make distinguishing signal features difficult. The penalty factor α influences the bandwidth of each modal component. Fig. 9 shows that the fault diagnosis accuracy is sensitive to the

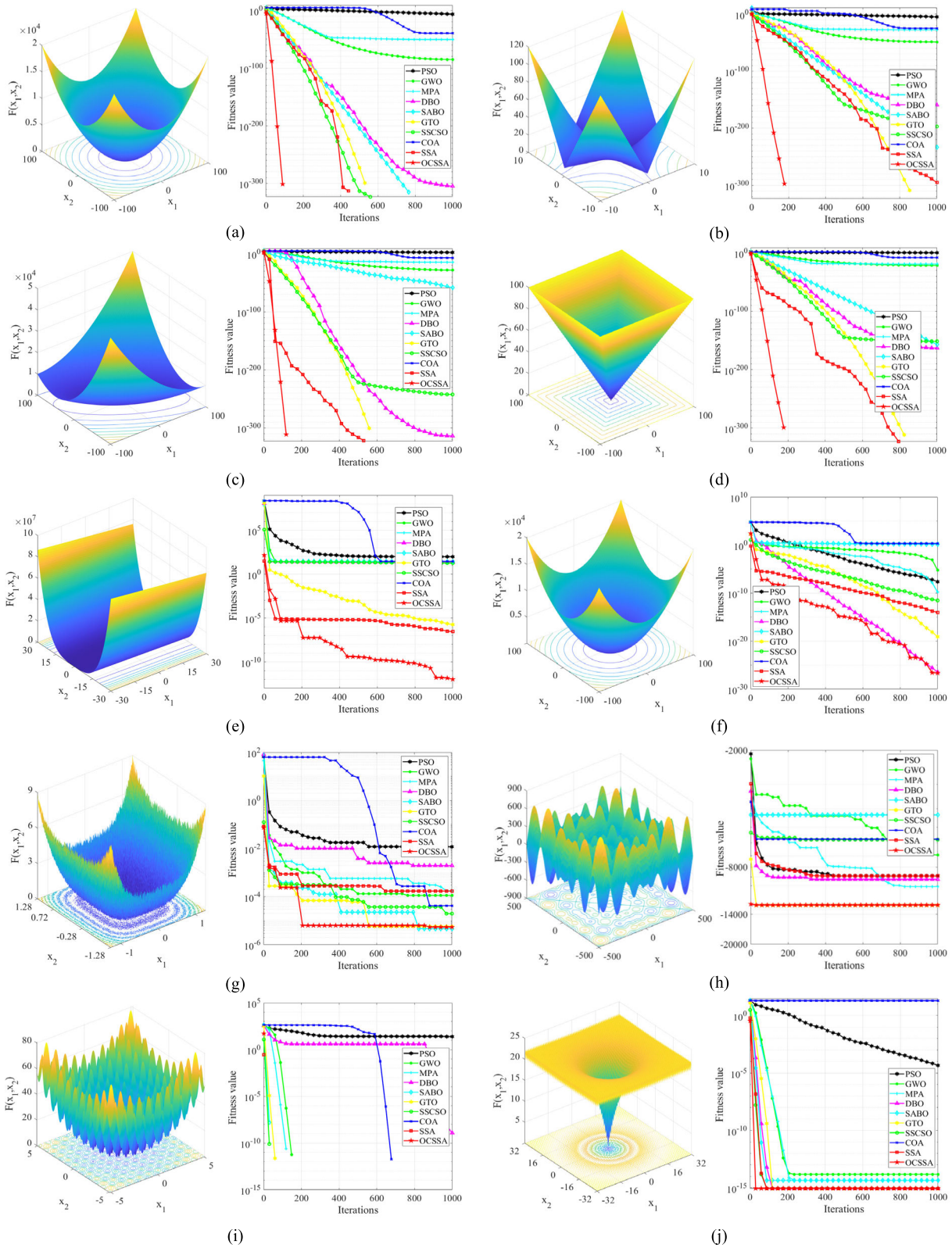


FIGURE 5. 10 classical test functions and convergence curves in CEC2005 test function: (a) f_1; (b) f_2; (c) f_3; (d) f_4; (e) f_5; (f) f_6; (g) f_7; (h) f_8; (i) f_9; (a) f_{10}.

value of α. If α is too low, the components may be too broad, and if they are too high, they may be too narrow. The optimal α value, as shown in the plot, was approximately 3500, when the accuracy reached its maximum.

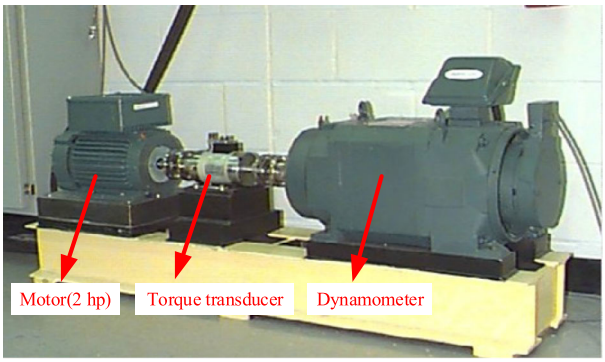


FIGURE 6. Bearing fault simulation test bench.

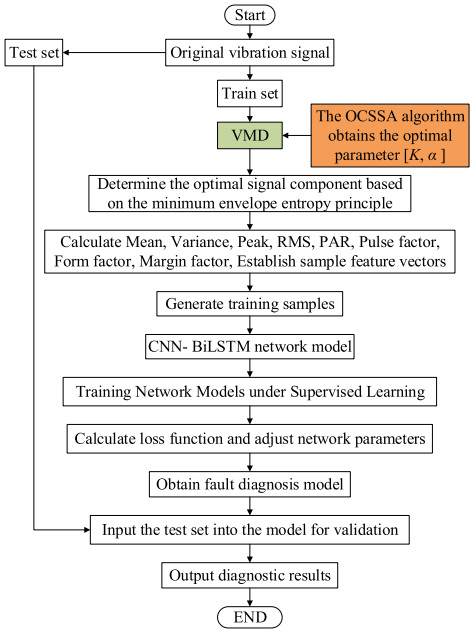


FIGURE 7. OCSSA-VMD-CNN-BiLSTM model fault diagnosis flow.

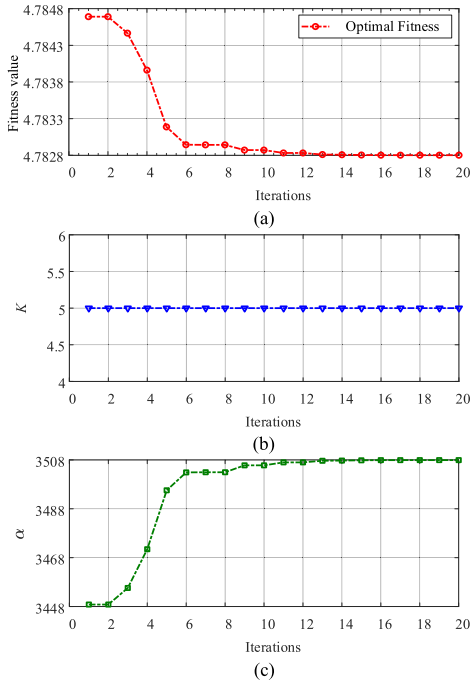


FIGURE 8. K and α parameter optimization: (a) OCSSA evolution curve; (b) optimization process curve of K; and (c) optimization curve of penalty factor α.

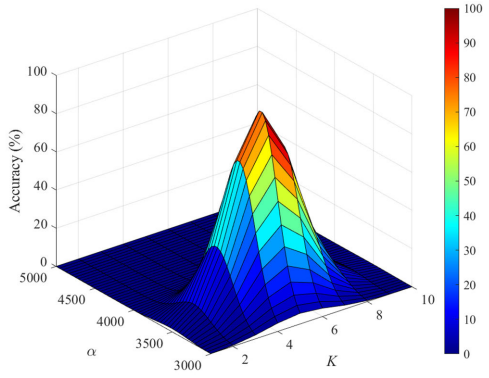


FIGURE 9. Effect of VMD parameters on fault diagnosis accuracy.

Fig. 10(a)-(d) show time-domain comparison diagrams before and after signal decomposition and reconstruction for the bearing normal, inner ring, outer ring, and rolling body faults, respectively. These figures demonstrate that after the optimized VMD reconstruction, the time-domain signals exhibited significant noise reduction, and the impact characteristics of the fault signals became more pronounced.

Fig. 11 illustrates the accuracy curve of a fault diagnosis model based on the OCSSA-VMD-CNN-BiLSTM using the CWRU dataset. The model performed excellently during both the training and testing phases, stabilizing after approximately 150 iterations and achieving a testing accuracy of up to 99.34% at 300 iterations.

Fig. 12 shows the loss function curve of a fault diagnosis model based on the OCSSA-VMD-CNN-BiLSTM using the CWRU dataset. The model demonstrated excellent performance, with both the training and testing losses rapidly decreasing and stabilizing at low values after approximately 50 iterations.

To evaluate the performance of OCSSA-VMD-CNN-BiLSTM, the proposed model was compared with other state-of-the-art (SOTA) models [40], [41], [42], [43], [44], [45], [46]. The confusion matrix and classification results for fault diagnosis across the eight models are shown in Fig. 13. After repeated validation, the OCSSA-VMD-CNN-BiLSTM model demonstrated superior performance, achieving a recognition accuracy of 96.7% for normal faults and fault diameters of 0.021 inches, and 100% for all other fault types. The overall recognition accuracy for the ten types of faults reached 99.34%. Furthermore, the recognition speed of the OCSSA-VMD-CNN-BiLSTM model was significantly improved, demonstrating its effectiveness in fault diagnosis.

Deep learning methods, such as CNN-LSTM, CNN-BiLSTM, and VMD-CNN-BiLSTM, perform exceptionally well in fault recognition and generally achieve higher accuracy than traditional machine learning methods. The

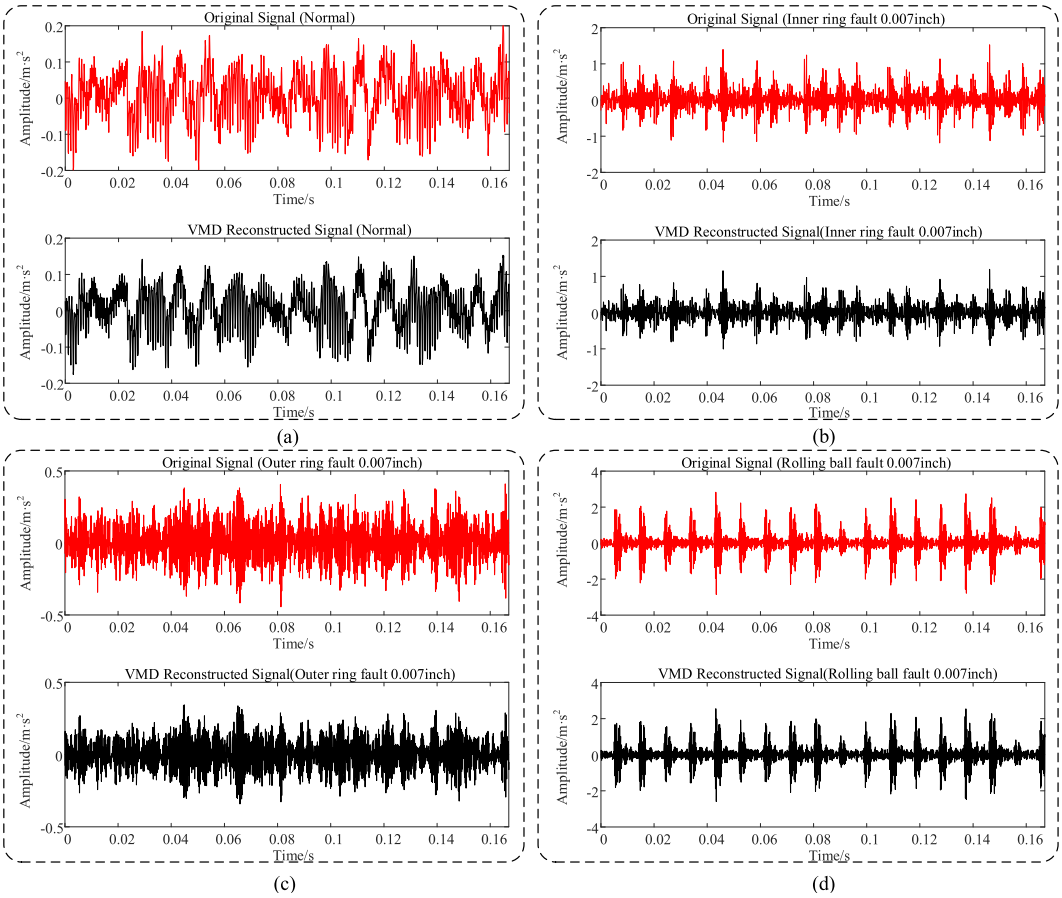


FIGURE 10. Time domain comparison between the original and the reconstructed signals of different faults: (a) Normal; (b) Inner ring fault diameter 0.007 inches; (c) Outer ring fault diameter 0.007 inches; (d) Rolling ball fault diameter 0.007 inches.

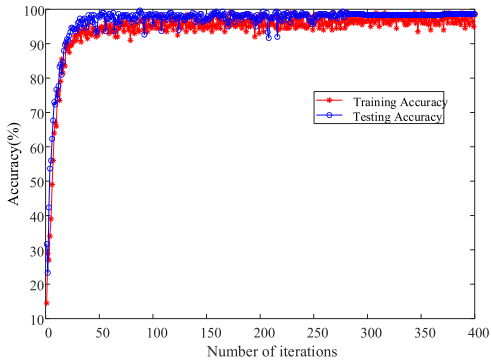


FIGURE 11. Accuracy curve of the fault diagnosis model based on OCSSA-VMD-CNN-BiLSTM using the CWRU dataset.

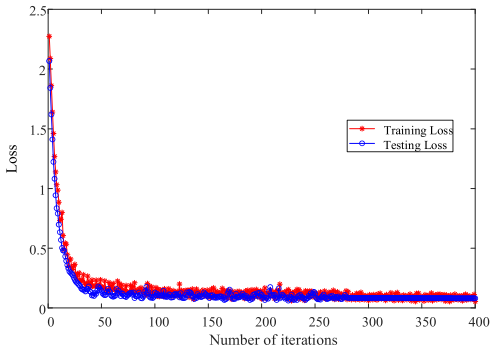
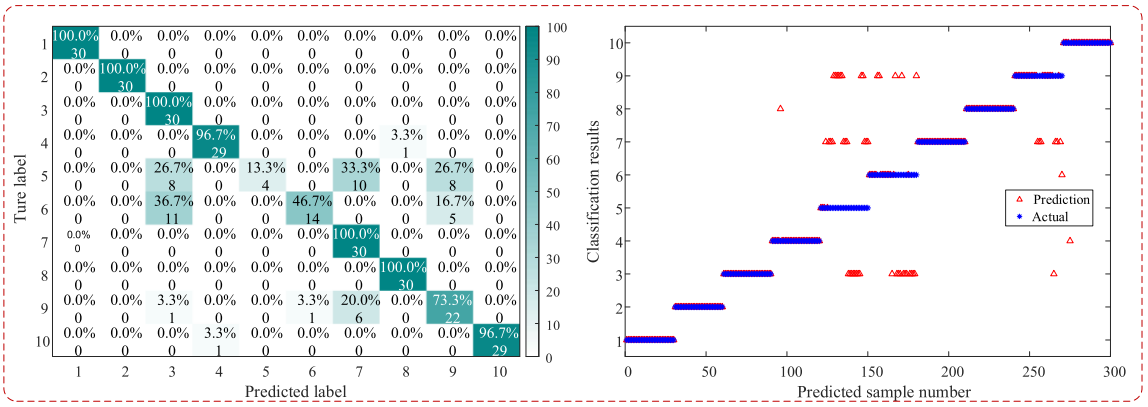


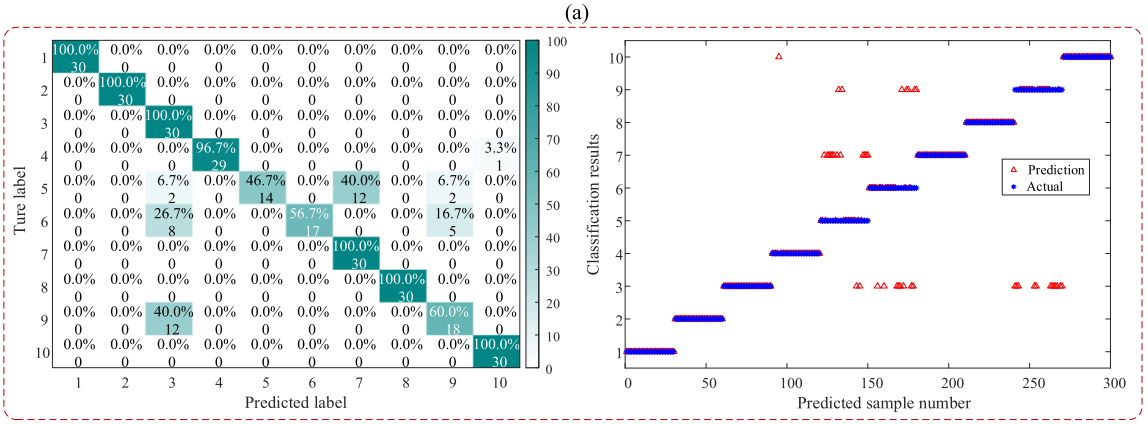
FIGURE 12. Loss function curve of the fault diagnosis model based on OCSSA-VMD-CNN-BiLSTM using the CWRU dataset.

incorporation of techniques such as VMD and OCSSA further enhances the accuracy of fault recognition. This analysis indicates that methods combining deep learning and optimization techniques have significant advantages in multiclass fault recognition tasks, providing strong support for practical engineering applications.

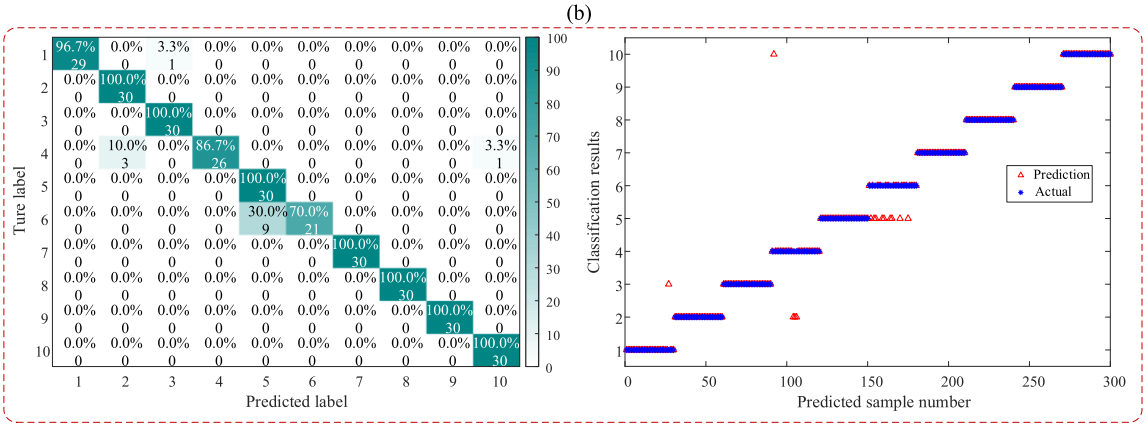
Fig. 14 shows the diagnostic accuracy of the different experimental methods using the CWRU dataset. The figure presents the accuracy and average processing time of eight different fault diagnosis models applied to ten types of faults. The OCSSA-VMD-CNN-BiLSTM model stands out as the most effective method for fault diagnosis, achieving 100% accuracy for all faults except one, with an overall accuracy of 99.34%. It maintains a very short processing time of 6.67 seconds, second only to the CNN-SVM, which has a significantly lower accuracy. The consistently high accuracy across the different fault types demonstrates robustness. The combination of optimization (OCSSA) and advanced feature



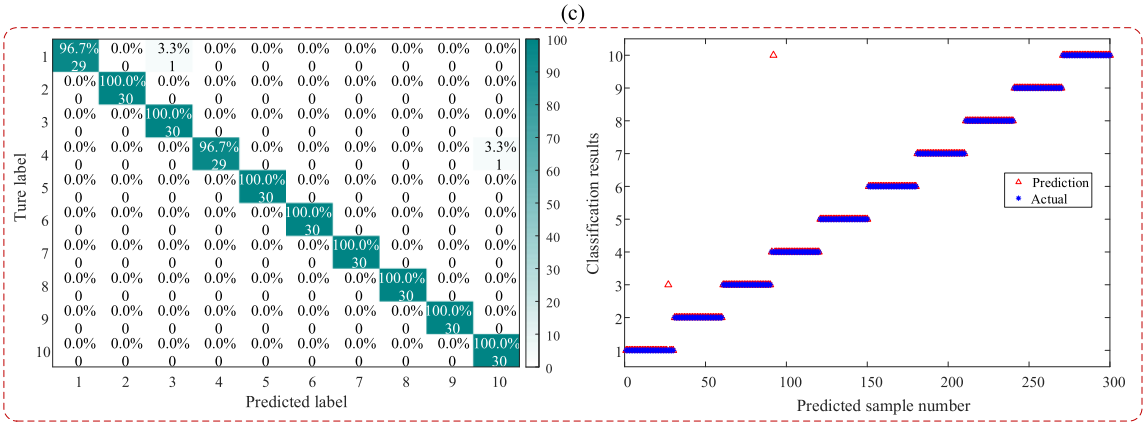
(a)



(b)

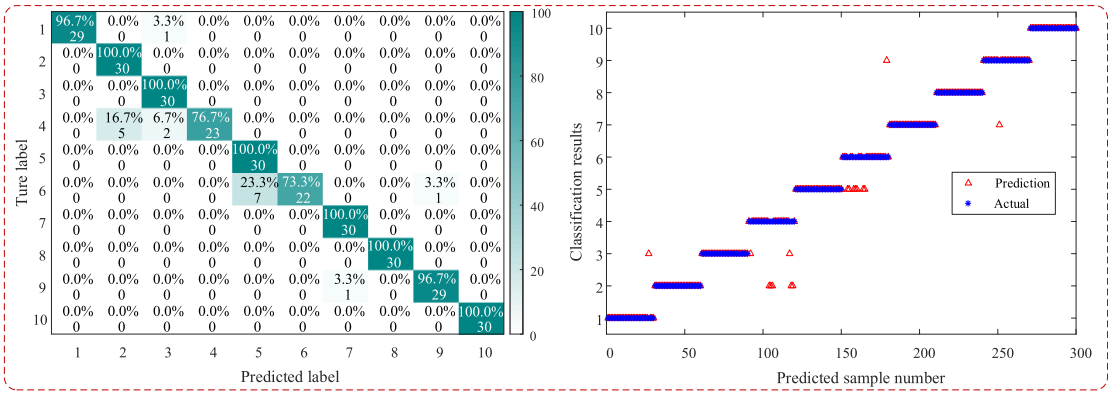


(c)

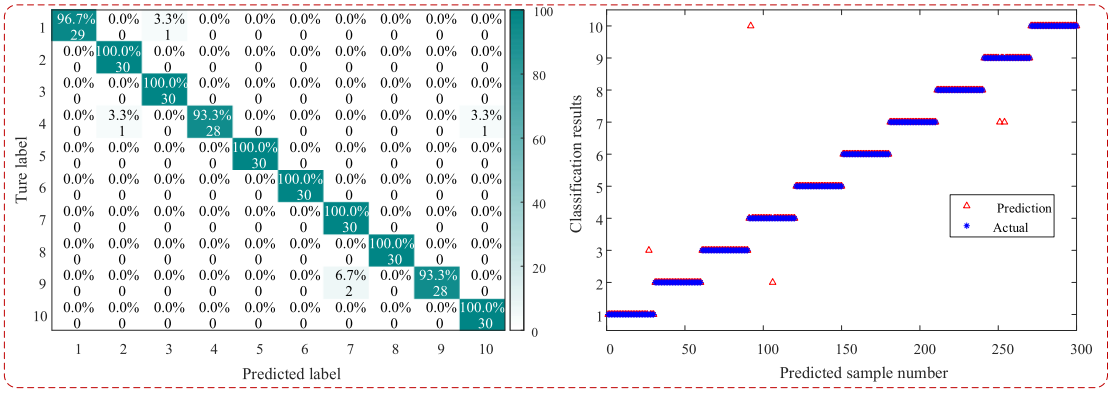


(d)

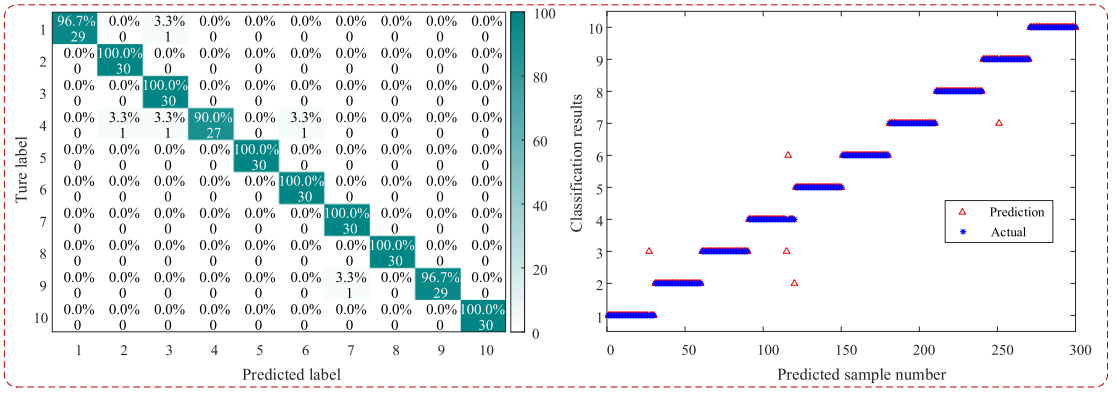
FIGURE 13. The accuracy of the test results for the eight methods: (a) CNN-LSTM; (b) CNN-BiLSTM; (c) VMD-CNN-BiLSTM; (d) OCSSA-VMD-CNN-BiLSTM.



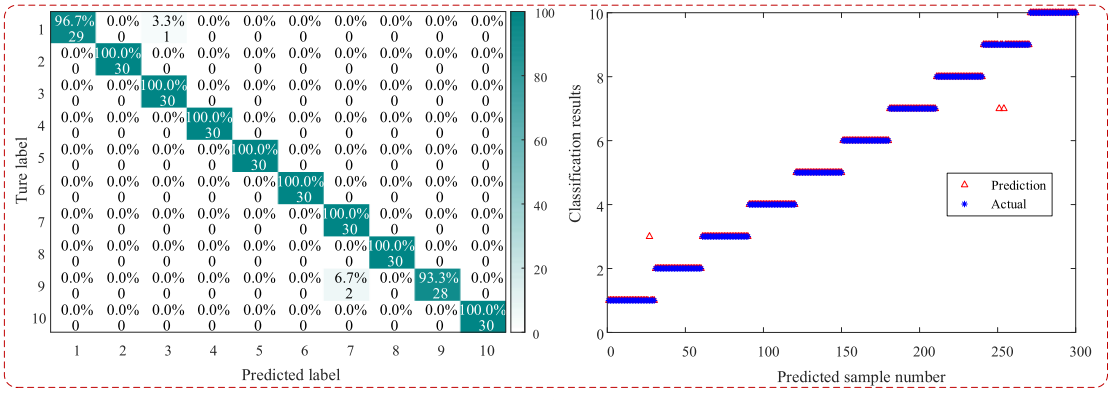
(e)



(f)

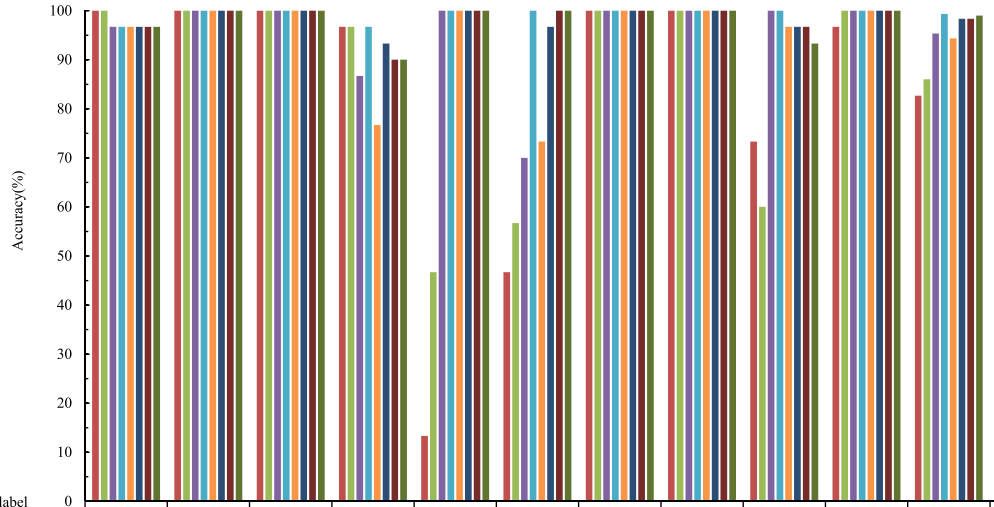


(g)



(h)

FIGURE 13. ***(Continued.)*** **The accuracy of the test results for the eight methods: (e) BiGRU; (f) CNN-BiLSTM-Attention; (g) CNN-BiGRU-Attention; (h) CNN-SVM.**



	1	2	3	4	5	6	7	8	9	10	Avg.	Time(s)
CNN-LSTM	100	100	100	96.7	13.3	46.7	100	100	73.3	96.7	82.67	106.83
CNN-BiLSTM	100	100	100	96.7	46.7	56.7	100	100	60	100	86.01	148.07
VMD-CNN-BiLSTM	96.7	100	100	86.7	100	70	100	100	100	100	95.34	6.47
OCSSA-VMD-CNN-BiLSTM	96.7	100	100	96.7	100	100	100	100	100	100	99.34	6.67
BiGRU	96.7	100	100	76.7	100	73.3	100	100	96.7	100	94.34	21.08
CNN-BiLSTM-Attention	96.7	100	100	93.3	100	96.7	100	100	96.7	100	98.34	8.59
CNN-BiGRU-Attention	96.7	100	100	90	100	100	100	100	96.7	100	98.34	33.86
CNN-SVM	96.7	100	100	90	100	100	100	100	93.3	100	99	5.7

FIGURE 14. Diagnostic accuracy of different experimental methods using the CWRU dataset.

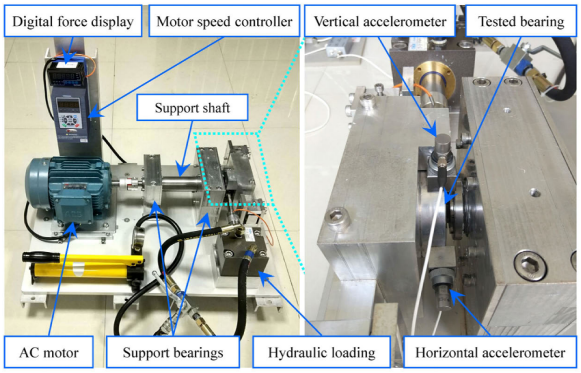


FIGURE 15. XJTU rolling bearing accelerated life experimental device.

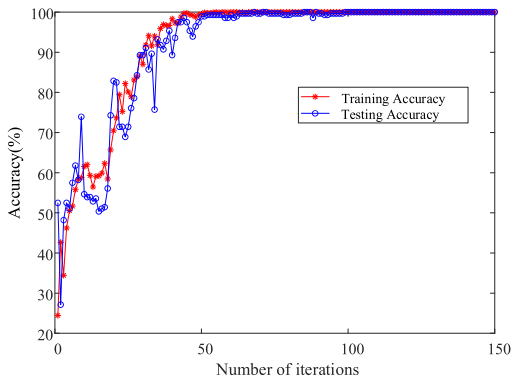


FIGURE 16. Accuracy curve of the fault diagnosis model based on OCSSA-VMD-CNN-BiLSTM using XJTU dataset.

extraction (VMD) significantly enhanced the performance of the CNN-BiLSTM model, making it the best choice for

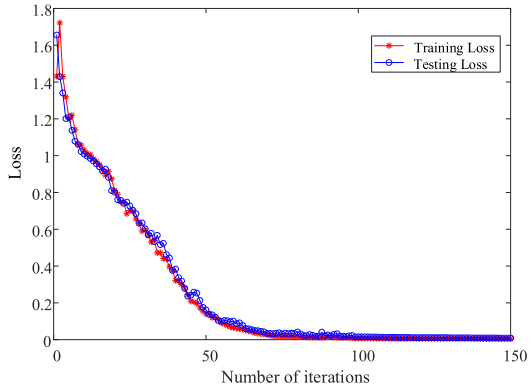


FIGURE 17. Loss function curve of the fault diagnosis model based on OCSSA-VMD-CNN-BiLSTM using XJTU dataset.

practical applications that require high accuracy and efficiency in fault diagnosis.

2) EXPERIMENT 2: XJUT BEARING DATASET

To verify the reliability, accuracy, and universality of the OCSSA-VMD-CNN-BiLSTM model, this study reanalyzes a rolling bearing accelerated life experimental dataset from Xi'an Jiaotong University (XJUT) was used. This dataset was provided by the Institute of Design Science and Basic Components at Xi'an Jiaotong University in Shaanxi, China and Sumyoung Technology Co., Ltd. in Changxing, Zhejiang [47]. The bearing test bench is depicted in Fig. 15 and comprises an alternating current (AC) induction motor, motor speed controller, support shaft, two support bearings, and

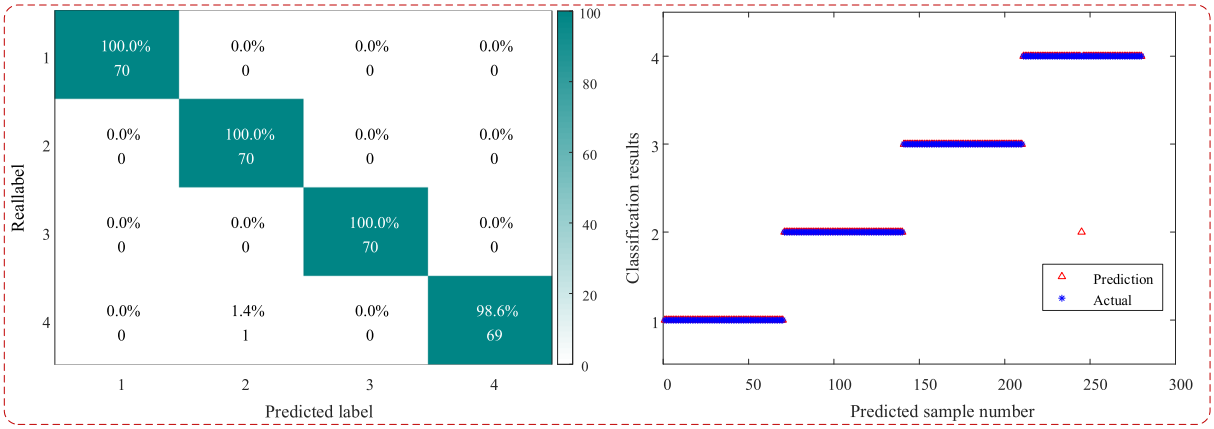
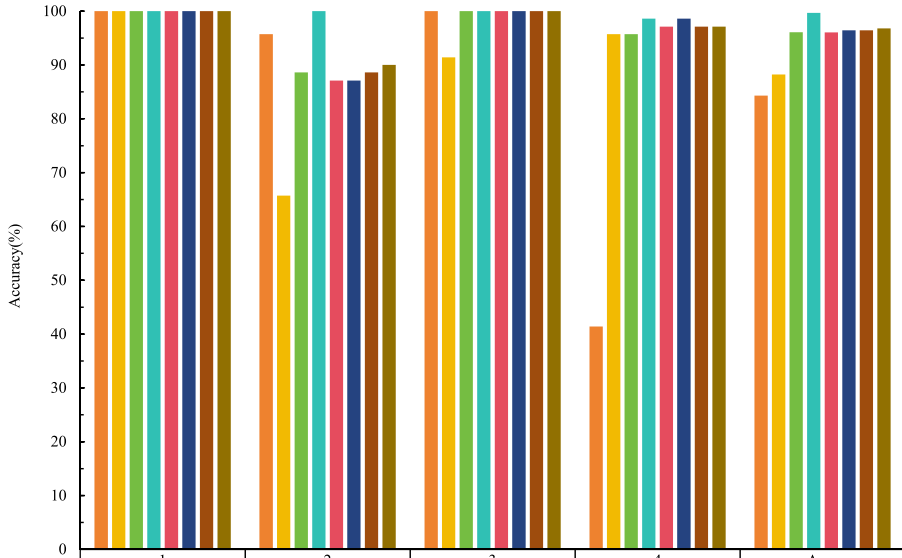


FIGURE 18. The accuracy of the test results for the OCSSA-VMD-CNN-BiLSTM using the XJTU dataset.



	1	2	3	4	Avg.	Time(s)
CNN-LSTM	100	95.7	100	41.4	84.28	136.53
CNN-BiLSTM	100	65.7	91.4	95.7	88.21	152.39
VMD-CNN-BiLSTM	100	88.6	100	95.7	96.07	7.62
OCSSA-VMD-CNN-BiLSTM	100	100	100	98.6	99.65	8.47
BiGRU	100	87.1	100	97.1	96.05	21.48
CNN-BiLSTM-Attention	100	87.1	100	98.6	96.43	9.67
CNN-BiGRU-Attention	100	88.6	100	97.1	96.43	49.34
CNN-SVM	100	90	100	97.1	96.78	11.18

FIGURE 19. Diagnostic accuracy of different experimental methods using the CWRU dataset.

hydraulic loading system. The sampling period was 1 min at 2250 rpm (37.5Hz) and 11 kN, with each sampling period lasting 1.28 seconds. The last six CSV files from Bearing2_1 (inner race fault), Bearing2_2 (outer race fault), and Bearing2_3 (cage fault) were selected as rolling bearing fault data for this analysis. All faults occur naturally during regular operations, thereby enhancing the realism of the data. Specifically, files 100-105 from Bearing2_3 were used as data to represent the healthy rolling bearing conditions. The dataset recorded four operating states of the bearings and used overlapping sampling for data augmentation, with each sample being 1600 in length and an overlap of 1200. The dataset divisions are listed in Table 5.

Fig. 16 shows the accuracy curve of a fault diagnosis model based on the OCSSA-VMD-CNN-BiLSTM using the XJTU dataset. The training accuracy started at approximately 30% and rapidly increased to approximately 70% by the 20th iteration. As the number of iterations increases, the fluctuations decrease, and the training accuracy reaches approximately 90% by the 50th iteration, eventually approaching 100%, indicating that the model has nearly perfectly learned the training data. Similarly, the testing accuracy started at

TABLE 5. XJTU dataset division results.

Fault type	Normal	Inner ring	Outer ring	Cage
Fault label	1	2	3	4
Training sample	350	350	350	350
Testing sample	70	70	70	70

approximately 30% and reached approximately 60% by the 20th iteration. Despite larger fluctuations, it followed a similar trend, reaching approximately 85% by the 50th iteration and eventually approaching 100%, demonstrating an excellent generalization to unseen data. The accuracy curve indicates that the model learns efficiently and generalizes well, making it suitable for practical fault-diagnosis applications.

Fig. 17 presents the loss function curve of the fault diagnosis model based on the OCSSA-VMD-CNN-BiLSTM using the XJTU dataset. Both the training and testing losses start at approximately 1.6 and quickly decrease, stabilizing near zero after approximately 50 iterations. This indicates that the model learns rapidly during the initial phase and then gradually stabilizes, thereby demonstrating effective learning and strong generalization capabilities. The close alignment between the training and testing losses signifies that the model performs similarly well on unseen data, highlighting its high stability and reliability.

The results shown in Fig. 18 highlight the high accuracy and reliability of the OCSSA-VMD-CNN-BiLSTM model for fault diagnosis using the XJTU dataset. The confusion matrix demonstrated a near-perfect classification for all classes, with only a single misclassification. The scatter plot reinforced this finding, showing a strong correlation between the predicted and actual labels. These findings indicate the suitability of the model for practical applications in fault diagnosis, offering a precise and dependable performance.

Fig. 19 shows the diagnostic accuracy and average processing time of the eight fault diagnosis models on the CWRU dataset. The OCSSA-VMD-CNN-BiLSTM model continued to perform exceptionally well, achieving nearly 100% accuracy for all fault types with an average accuracy of 99.65%. The processing time was short, 8.47 seconds, demonstrating high efficiency and a high accuracy. The model maintains high accuracy across different fault types, demonstrating its robustness.

The accuracy curve of the OCSSA-VMD-CNN-BiLSTM fault diagnosis model demonstrated efficient learning and robust generalization for the XJTU dataset. The model achieved high accuracy for both the training and testing data, with stability and minimal fluctuations as the number of iterations increased. This indicated the suitability of the model for practical applications in fault diagnosis.

VI. CONCLUSION

To address the challenges of noise susceptibility and difficulty in extracting weak fault features from rolling bearing signals, this study proposes a motor bearing fault diagnosis model based on OCSSA-VMD-CNN-BiLSTM. The optimization process begins by initializing the population diversity using logistic chaotic mapping. The global exploration strategy of the OOA replaces the discoverer position updating formula of the original sparrow algorithm, whereas the Cauchy mutation strategy is used to replace the follower position updating formula. The minimum envelope entropy serves as the fitness function that guides the adaptive selection of the modal number K and penalty coefficient α. This approach improves the decomposition quality of VMD, enabling the effective extraction of real components containing fault characteristic information. The proposed OCSSA-VMD-CNN-BiLSTM model was validated using the CWRU and XJTU datasets, and compared with eight other state-of-the-art (SOTA) models. The results demonstrate that the proposed method achieves faster fault recognition speed and higher fault recognition accuracy.

CONFLICT OF INTEREST STATEMENT

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

REFERENCES

[1] M. Alonso-González, V. G. Díaz, B. L. Pérez, B. C. P. G-Bustelo, and J. P. Anzola, "Bearing fault diagnosis with envelope analysis and machine learning approaches using CWRU dataset," *IEEE Access*, vol. 11, pp. 57796–57805, 2023.

[2] Z. Ke, C. Di, and X. Bao, "Adaptive suppression of mode mixing in CEEMD based on genetic algorithm for motor bearing fault diagnosis," *IEEE Trans. Magn.*, vol. 58, no. 2, pp. 1–6, Feb. 2022.

[3] L. Wan, Y. Chen, H. Li, and C. Li, "Rolling-element bearing fault diagnosis using improved LeNet-5 network," *Sensors*, vol. 20, no. 6, p. 1693, Mar. 2020.

[4] Q. Zhang and L. Deng, "An intelligent fault diagnosis method of rolling bearings based on short-time Fourier transform and convolutional neural network," *J. Failure Anal. Prevention*, vol. 23, no. 2, pp. 795–811, Feb. 2023.

[5] B. B. Etana, A. A. Dawud, B. Malengier, W. Sitek, W. F. Gemechu, J. Krishnamoorthy, and L. V. Langenhove, "Discrete wavelet transform based processing of embroidered textile-electrode electromyography signal acquired with load and pressure effect," *J. Ind. Textiles*, vol. 54, pp. 1–21, Feb. 2024.

[6] R. A. Rios and R. F. D. Mello, "Applying empirical mode decomposition and mutual information to separate stochastic and deterministic influences embedded in signals," *Signal Process.*, vol. 118, pp. 159–176, Jan. 2016.

[7] K. Geetha and M. K. Hota, "Microseismic signal denoising based on variational mode decomposition with adaptive non-local means filtering," *Pure Appl. Geophysics*, vol. 180, no. 11, pp. 3709–3731, Nov. 2023.

[8] H. Li, T. Liu, X. Wu, and Q. Chen, "Research on bearing fault feature extraction based on singular value decomposition and optimized frequency band entropy," *Mech. Syst. Signal Process.*, vol. 118, pp. 477–502, Mar. 2019.

[9] W. Tuerxun, X. Chang, G. Hongyu, J. Zhijie, and Z. Huajian, "Fault diagnosis of wind turbines based on a support vector machine optimized by the sparrow search algorithm," *IEEE Access*, vol. 9, pp. 69307–69315, 2021.

[10] J. Ma, Y. Li, L. Wang, J. Hu, H. Li, J. Fei, L. Li, and G. Zhao, "Stator ITSC fault diagnosis for EMU induction traction motor based on Goertzel algorithm and random forest," *Energies*, vol. 16, no. 13, p. 4949, Jun. 2023.

[11] J. Lu, W. Qian, S. Li, and R. Cui, "Enhanced K-nearest neighbor for intelligent fault diagnosis of rotating machinery," *Appl. Sci.*, vol. 11, no. 3, p. 919, Jan. 2021.

[12] L. M. Elshenawy, C. Chakour, and T. A. Mahmoud, "Fault detection and diagnosis strategy based on k-nearest neighbors and fuzzy C-means clustering algorithm for industrial processes," *J. Franklin Inst.*, vol. 359, no. 13, pp. 7115–7139, Sep. 2022.

[13] J. Yu and G. Liu, "Knowledge extraction and insertion to deep belief network for gearbox fault diagnosis," *Knowledge-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105883.

[14] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," *ISA Trans.*, vol. 77, pp. 167–178, Jun. 2018.

[15] H. Yan, Y. Qin, S. Xiang, Y. Wang, and H. Chen, "Long-term gear life prediction based on ordered neurons LSTM neural networks," *Measurement*, vol. 165, Dec. 2020, Art. no. 108205.

[16] X. Wang, Y. Qin, Y. Wang, S. Xiang, and H. Chen, "ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis," *Neurocomputing*, vol. 363, pp. 88–98, Oct. 2019.

[17] C. Liu, G. Cheng, X. Chen, and Y. Pang, "Planetary gears feature extraction and fault diagnosis method based on VMD and CNN," *Sensors*, vol. 18, no. 5, p. 1523, May 2018.

[18] Y. Hou, J. Wang, Z. Chen, J. Ma, and T. Li, "Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved transformer," *Eng. Appl. Artif. Intell.*, vol. 124, Sep. 2023, Art. no. 106507.

[19] J. Tong, S. Tang, Y. Wu, H. Pan, and J. Zheng, "A fault diagnosis method of rolling bearing based on improved deep residual shrinkage networks," *Measurement*, vol. 206, Jan. 2023, Art. no. 112282.

[20] Y. Li, W. Zou, and L. Jiang, "Fault diagnosis of rotating machinery based on combination of Wasserstein generative adversarial networks and long short term memory fully convolutional network," *Measurement*, vol. 191, Mar. 2022, Art. no. 110826.

[21] Y. Guo, J. Mao, and M. Zhao, "Rolling bearing fault diagnosis method based on attention CNN and BiLSTM network," *Neural Process. Lett.*, vol. 55, no. 3, pp. 3377–3410, Oct. 2022.

[22] G. Fu, Q. Wei, Y. Yang, and C. Li, "Bearing fault diagnosis based on CNN-BiLSTM and residual module," *Meas. Sci. Technol.*, vol. 34, no. 12, Sep. 2023, Art. no. 125050.

[23] X. Liu, G. Chen, H. Wang, and X. Wei, "A Siamese CNN-BiLSTM-based method for unbalance few-shot fault diagnosis of rolling bearings," *Meas. Control*, vol. 57, no. 5, pp. 551–565, May 2024.

[24] Y. Chang, G. Bao, S. Cheng, T. He, and Q. Yang, "Improved VMD-KFCM algorithm for the fault diagnosis of rolling bearing vibration signals," *IET Signal Process.*, vol. 15, no. 4, pp. 238–250, Apr. 2021.

[25] C. Yong, H. Ting, and C. Peng, "Enhancing sparrow search algorithm with OCSSA: Integrating osprey optimization and Cauchy mutation for improved convergence and precision," *Electron. Lett.*, vol. 60, no. 4, pp. 1–3, Feb. 2024.

[26] N. Jia and T. Wang, "A GCM neural network with piecewise logistic chaotic map," *Symmetry*, vol. 14, no. 3, p. 506, Mar. 2022.

[27] D. Mohammad and T. Pavel, "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *Frontiers Mech. Eng.*, vol. 8, pp. 1–43, Jan. 2023.

[28] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[29] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.

[30] J. Xue and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *J. Supercomput.*, vol. 79, no. 7, pp. 7305–7336, May 2023.

[31] P. Trojovský and M. Dehghani, "Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems," *Biomimetics*, vol. 8, no. 2, p. 149, Apr. 2023.

[32] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, and S. Mirjalili, "Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems," *Int. J. Intell. Syst.*, vol. 36, no. 10, pp. 5887–5958, Jul. 2021.

[33] O. M. O. Patrick, O. Nicholas, and O. Winston, "Performance analysis of an improved particle swarm optimization and the standard particle swarm optimization," *Int. J. Eng. Adv. Technol.*, vol. 13, no. 1, pp. 37–42, Oct. 2023.

[34] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[35] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Exp. Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.

[36] M. Dehghani, Z. Montazeri, E. Trojovská, and P. Trojovský, "Coati optimization algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems," *Knowledge-Based Syst.*, vol. 259, Jan. 2023, Art. no. 110011.

[37] A. Seyyedabbasi and F. Kiani, "Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems," *Eng. Comput.*, vol. 39, no. 4, pp. 2627–2651, Aug. 2023.

[38] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Natural Comput.*, vol. 2005, pp. 341–357, Jan. 2005.

[39] Case Western Reserve University Case School of Engineering Bearing Data Center. [Online]. Available: https://engineering.case.edu/bearingdatacenter

[40] C. Qin, Y. Jin, Z. Zhang, H. Yu, J. Tao, H. Sun, and C. Liu, "Anti-noise diesel engine misfire diagnosis using a multi-scale CNN-LSTM neural network with denoising module," *CAAI Trans. Intell. Technol.*, vol. 8, no. 3, pp. 963–986, Sep. 2023.

[41] J. Guo, J. Wang, Z. Wang, Y. Gong, J. Qi, G. Wang, and C. Tang, "A CNN-BiLSTM-bootstrap integrated method for remaining useful life prediction of rolling bearings," *Qual. Rel. Eng. Int.*, vol. 39, no. 5, pp. 1796–1813, Jul. 2023.

[42] R. Bazi, T. Benkedjouh, H. Habbouche, S. Rechak, and N. Zerhouni, "A hybrid CNN-BiLSTM approach-based variational mode decomposition for tool wear monitoring," *Int. J. Adv. Manuf. Technol.*, vol. 119, nos. 5–6, pp. 3803–3817, Mar. 2022.

[43] D. She and M. Jia, "A BiGRU method for remaining useful life prediction of machinery," *Measurement*, vol. 167, Jan. 2021, Art. no. 108277.

[44] T. Ma, G. Xiang, Y. Shi, and Y. Liu, "Horizontal in situ stresses prediction using a CNN-BiLSTM-attention hybrid neural network," *Geomechanics Geophysics Geo-Energy Geo-Resources*, vol. 8, no. 5, pp. 1–26, Sep. 2022.

[45] P. Ma, G. Li, H. Zhang, C. Wang, and X. Li, "Prediction of remaining useful life of rolling bearings based on multiscale efficient channel attention CNN and bidirectional GRU," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, 2024.

[46] W. Gong, H. Chen, Z. Zhang, M. Zhang, R. Wang, C. Guan, and Q. Wang, "A novel deep learning method for intelligent fault diagnosis of rotating machinery based on improved CNN-SVM and multichannel data fusion," *Sensors*, vol. 19, no. 7, p. 1693, Apr. 2019.

[47] B. Wang, Y. Lei, N. Li, and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings," *IEEE Trans. Rel.*, vol. 69, no. 1, pp. 401–412, Mar. 2020.

YONG CHANG was born in China. He received the B.E. and M.S.E. degrees from Lanzhou University of Technology, Lanzhou, China, in 2010 and 2014, respectively, where he is currently pursuing the Ph.D. degree with the College of Electrical Engineering and Information Engineering. His current research interests include electric drive, fault diagnosis, and signal processing.

GUANGQING BAO was born in China. She received the Ph.D. degree in power electronics and electric drives from Shanghai University, Shanghai, China, in 2006. She was engaged in the research of power conversion with The University of Manchester, U.K., from 2007 to 2008. Since 2022, she has been with Southwest Petroleum University, where she is currently a Professor with the School of Electronics and Information Engineering. Her research interests include electrical machine systems and renewable energy.