

Article

A Rolling Bearing Fault Diagnosis Method Based on ICEEMDAN and AHO-SVM

Liping Wang, Yaozheng Zhao, Yan Chen and Guangyong Xi *

School of Computer Science and Artificial Intelligence, Zhengzhou University of Light Industry, Zhengzhou 450000, China; kinglp2026@163.com (L.W.); 542307130434@zzuli.edu.cn (Y.Z.); 2008043@zzuli.edu.cn (Y.C.)

* Correspondence: xigy@zzuli.edu.cn

Abstract

To address the issues of rolling bearing fault vibration signals being susceptible to noise interference and the support vector machine (SVM) relying on manual parameter settings, this paper proposes a fault diagnosis method based on ICEEMDAN and AIHO-SVM. Firstly, the Hippopotamus Optimization Algorithm is improved by incorporating Chebyshev chaotic mapping, refraction opposite learning, dynamic weighting, adaptive step size, and guided learning strategies, thereby enhancing convergence accuracy and speed. Secondly, ICEEMDAN is employed to decompose vibration signals for noise reduction, and the effective intrinsic mode function (IMF) components are selected according to the mutual information criterion to reconstruct the signal. Nine time-domain statistical features are then extracted to construct the fault feature vector. Thirdly, AIHO is used to collaboratively optimize the penalty factor and kernel parameters of SVM, establishing an AIHO-SVM classification model. Finally, the proposed method is validated on bearing datasets from Case Western Reserve University and Huazhong University of Science and Technology. Experimental results show that the average diagnostic accuracies on the two datasets reach 99.07% and 99.09%, respectively, demonstrating the effectiveness of the proposed method for rolling bearing fault diagnosis.

Keywords: SVM; ICEEMDAN; AIHO; fault diagnosis; rolling bearing

1. Introduction

Rolling bearings are indispensable core components of rotating mechanical equipment and are widely used in critical fields such as automobiles, high-speed railways, and aerospace engines [1]. Bearings operate in harsh environments over long periods, subject to alternating loads and complex stresses, which make them highly susceptible to typical faults such as wear, pitting, and spalling [2]. According to statistics, about 45–55% of rotating machinery faults originate from rolling bearings, and in the field of electric motors, this proportion even exceeds 40% [3]. If these damages are not identified and intervened in a timely and accurate manner, the faults will accumulate progressively, ultimately leading to equipment failure and even major safety accidents. Therefore, achieving fault diagnosis of rolling bearings is of great significance for ensuring stable equipment operation and implementing predictive maintenance.

During the operation of rolling bearings, their faults are often accompanied by abnormal vibration signals, and such signals tend to exhibit nonlinear and non-stationary

Academic Editors: Muhammad Farooq Siddique and Izaz Raouf

Received: 17 June 2026

Revised: 11 July 2026

Accepted: 17 July 2026

Published: 19 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

characteristics. During the acquisition process, they are highly susceptible to interference from environmental noise and the equipment's own vibrations, causing fault features to be masked [4]. Therefore, it is usually necessary to process the original vibration signals before conducting bearing fault diagnosis. Empirical Mode Decomposition (EMD), as a classical signal processing method, has good adaptability to non-stationary and nonlinear vibration signals and has been widely applied in the field of mechanical fault diagnosis [5]. However, the EMD method is prone to mode mixing when processing complex signals, which interferes with subsequent feature extraction. To alleviate this problem, Wu et al. [6] proposed the Ensemble Empirical Mode Decomposition (EEMD) method, which adds Gaussian white noise to the original signal to make the distribution of signal extrema more uniform, thereby suppressing mode mixing, but significant noise residue still remains in the reconstructed signal. To address these shortcomings, Torres et al. [7] proposed the Complete Ensemble EMD with Adaptive Noise (CEEMDAN) method, which adaptively adds white noise at each decomposition stage and uses the residual signal to compute intrinsic mode functions layer by layer, achieving signal reconstruction with near-zero error. However, noise still remains in the modes of CEEMDAN, and mode mixing tends to occur in the early stages of decomposition. To solve these problems, Colominas et al. [8] developed the Improved CEEMDAN (ICEEMDAN), which further optimizes the noise addition mechanism and residual noise processing, enabling better preservation of effective impact components in the original signal while suppressing mode mixing. Li et al. [9] applied ICEEMDAN to power load forecasting and demonstrated that this method can effectively improve the accuracy of short-term load forecasting. Gao et al. [10] combined ICEEMDAN with flexible wavelet thresholding for fault diagnosis of wind turbine gearbox bearings and achieved good diagnostic results. Given that research on ICEEMDAN in the field of rolling bearing fault diagnosis is still relatively scarce, this paper adopts ICEEMDAN for noise reduction preprocessing of the original vibration signals of rolling bearings.

With the deepening of industrial intelligence, data-driven intelligent fault diagnosis methods are playing an increasingly important role in condition monitoring of rotating machinery. Deep learning has made remarkable progress in the field of fault diagnosis, with its powerful automatic feature extraction capability offering clear advantages when handling large-scale data. However, deep models typically rely on massive labeled samples and high computational resources, are time-consuming to train, and suffer from poor interpretability. In practical rolling bearing fault diagnosis, sample scarcity and difficulty in labeling fault data are common constraints, under which the advantages of deep learning are hardly exploited. In contrast, machine learning methods require relatively fewer samples and computational resources, offer better interpretability, and are more suitable for current diagnostic scenarios. Commonly used machine learning methods include artificial neural networks [11], random forests [12], and support vector machines (SVM) [13]. Among them, SVM, based on the principle of structural risk minimization, exhibits stronger generalization ability and robustness in handling high-dimensional, non-linear, and small-sample pattern recognition problems, and thus has been widely applied in bearing fault diagnosis [14,15]. However, the classification performance of SVM is highly dependent on the appropriate selection of the penalty factor and kernel function parameters; improper parameter settings can directly lead to reduced diagnostic accuracy, while manual parameter tuning based on experience involves considerable randomness. Therefore, using intelligent optimization algorithms to automatically optimize SVM parameters has become an effective approach to improving fault diagnosis performance. For example, Yan et al. [16] adopted the particle swarm optimization (PSO) algorithm to adaptively tune SVM parameters, combined with multi-domain feature extraction and Laplacian

score feature selection, to construct a rolling bearing fault diagnosis model, effectively improving fault identification accuracy under different operating conditions. Fan et al. [17] used a genetic algorithm-enhanced whale optimization algorithm to optimize SVM, achieving a diagnostic accuracy of 94.05% on 627 DGA samples, and the accuracy decreased by only about 5% under 10% noise disturbance. Zhang et al. [18] employed the seagull optimization algorithm (SOA) for parameter optimization of SVM to build an SOA-SVM diagnostic model, and combined it with variational mode decomposition optimized by the whale optimization algorithm and multi-scale permutation entropy (MPE), achieving high-precision rolling bearing fault diagnosis. Ding et al. [19] proposed a rolling bearing fault diagnosis method based on CMEGDE2D and CSO-SVM, using two-dimensional composite multi-scale integrated Gramian discrete entropy to extract fault features and employing the chicken swarm optimization algorithm to optimize the support vector machine for pattern recognition, significantly improving diagnostic accuracy and stability. However, due to the inherent limitations of swarm intelligence algorithms themselves, these algorithms still suffer from common problems such as slow convergence, susceptibility to local optima, and poor stability during the iteration process.

The Hippopotamus Optimization (HO) algorithm is a swarm intelligence optimization algorithm proposed by Amiri et al. [20] in 2024. It simulates the behaviors of hippopotamuses in terms of group positioning in water areas, predator defense, and escape, offering advantages such as high solution accuracy and fast convergence speed. However, as the complexity of practical problems increases, HO still suffers from shortcomings such as insufficient population diversity and a tendency to fall into local optima. To address these issues, some scholars have attempted to improve HO. Han et al. [21] introduced sine chaotic mapping and pinhole imaging reverse learning strategies for population initialization, which improved convergence speed and solution accuracy, but did not fully account for the algorithm's global search capability. Sachin Minocha et al. [22] employed iterative cosine operators and opposition-based learning to update hippopotamus positions, enhancing the exploration and exploitation abilities of the algorithm, yet the convergence accuracy remains limited. The above studies indicate that there is still considerable room for improving HO. To this end, this paper proposes a multi-strategy improved hippopotamus optimization algorithm, aiming to preserve the bionic advantages of HO while overcoming its engineering adaptability bottlenecks, so as to meet the specific requirements of SVM hyperparameter optimization.

In summary, to address the noise problem in rolling bearing fault diagnosis and the shortcomings of traditional intelligent algorithms used for optimizing SVM, this paper proposes a rolling bearing fault diagnosis method based on the combination of ICEEMDAN and AIHO-SVM. The main contributions of this paper are as follows:

- (1) ICEEMDAN is adopted to adaptively decompose the original vibration signals, and the optimal IMF components are selected based on the mutual information criterion, effectively suppressing mode mixing and noise interference.
- (2) By incorporating chaotic refraction initialization strategy, dynamic weight strategy, adaptive step size strategy, and guided learning strategy, an Adaptive Improved Hippopotamus Optimization (AIHO) algorithm is proposed. This algorithm enhances the population quality, search accuracy, and ability to escape local optima of the traditional HO algorithm.
- (3) A fault diagnosis model based on AIHO-SVM is established. By using AIHO to adaptively optimize the kernel function and penalty factor of SVM, the accuracy of bearing fault diagnosis is effectively improved.

2. Fundamental Principles

2.1. SVM

The SVM is a classic classifier based on the principle of structural risk minimization, and it has significant advantages in high-dimensional, nonlinear, and small-sample pattern recognition problems [23]. Its core idea is to map the input samples into a high-dimensional feature space via a kernel function, and to construct an optimal hyperplane that maximizes the classification margin in that space, thereby achieving good generalization ability.

For the nonlinear inseparable problem in actual rolling bearing fault diagnosis, SVM introduces slack variables ξ_i and penalty factor c to construct a soft-margin classifier, and its original optimization problem can be described as follows:

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + c \sum_{i=1}^l \xi_i \quad (1)$$

$$\text{s.t. } y_i [\mathbf{w}^T \phi(\mathbf{x}_i) + b] \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, 2, \dots, l \quad (2)$$

where $\phi(\mathbf{x}_i)$ is the nonlinear mapping function; $y_i \in \{-1, +1\}$ is the class label. By introducing Lagrange multipliers, the above problem can be transformed into its dual form for solution.

The kernel function is the key to achieving nonlinear mapping. This paper adopts the Radial Basis Function (RBF) kernel, and its expression is as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-g \|\mathbf{x}_i - \mathbf{x}_j\|^2), \quad g > 0 \quad (3)$$

where g is the kernel parameter. Research shows that the classification performance of SVM is significantly affected by the penalty factor c and the kernel parameter g . To improve the classification accuracy, the AIHO algorithm is adopted to collaboratively optimize c and g .

2.2. Principle of the ICEEMDAN Algorithm

EMD can decompose a signal into a series of IMFs and a residual sequence, but it is prone to mode mixing, which distorts the decomposition results. To suppress mode mixing, EEMD and CEEMDAN have been successively developed. However, the modal components of CEEMDAN still contain residual noise and are prone to generating spurious modes. To address this issue, the ICEEMDAN modifies the generation and addition of noise to reduce the noise content in the residual, making the decomposed modal components closer to the true modes and effectively reducing the occurrence of spurious modes. For ease of representation, the k -th mode obtained through EMD is defined as operator $E_k(\cdot)$, and the local mean of the signal obtained by $x - E_k(x)$ is defined as operator $M(\cdot)$.

ICEEMDAN constructs noisy signals. The first-order noise component is added to the original signal x , generating K reconstructed signals:

$$\mathbf{x}^{(i)} = \mathbf{x} + \beta_0 E_1(\mathbf{w}^{(i)}), \quad i = 1, 2, \dots, K \quad (4)$$

where $\mathbf{w}^{(i)}$ is the Gaussian white noise with zero mean and unit variance; β_0 is the noise standard deviation at the first decomposition. Then, the first residual component is calculated. The local mean of each reconstructed signal is computed, and then all results are averaged:

$$r_1 = \frac{1}{K} \sum_{i=1}^K M(\mathbf{x}^{(i)}) \quad (5)$$

The first modal component is extracted as:

$$\mathbf{d}_1 = \mathbf{x} - r_1 \quad (6)$$

For $k = 2, 3, \dots$, successively calculate the k -th residual component. Add the k -th order noise component to the previous residual:

$$r_k = \frac{1}{K} \sum_{i=1}^K M(r_{k-1} + \beta_{k-1} E_k(\mathbf{w}^{(i)})) \quad (7)$$

Extract the k -th modal component:

$$\mathbf{d}_k = r_{k-1} - r_k \quad (8)$$

Let $k = k + 1$ repeat the previous two steps until the number of extreme points of the residual r_k is not greater than 2 (i.e., the residual is a monotonic function) or the preset maximum number of decompositions is reached. At this point, the decomposition is terminated, and the original signal can be expressed as:

$$\mathbf{x} = \sum_{k=1}^m \mathbf{d}_k + r_m \quad (9)$$

where m is the total number of IMFs obtained from the decomposition.

2.3. Principle of the Hippopotamus Optimization Algorithm

The Hippopotamus Optimization (HO) algorithm [20] finds the optimal solution by simulating the behaviors of hippopotamuses in terms of water-area positioning, predator defense, and escape.

(1) Population initialization

HO first randomly generates an initial hippopotamus population in the solution space. Let the population size be N and the decision variable dimension be m ; the initial position of the i -th hippopotamus in the j -th dimension is initialized according to the following formula:

$$x_{ij} = lb_j + r_1 \cdot (ub_j - lb_j), \quad i = 1, 2, \dots, N, j = 1, 2, \dots, m \quad (10)$$

where lb_j, ub_j are the lower and upper bounds of the c -th dimension, respectively, and $r_1 \in [0, 1]$ is a uniformly distributed random number. All individuals form a position matrix X , and the corresponding objective function value vector is denoted as F .

(2) River/pond position update (exploration phase)

In this phase, the hippopotamus population updates its position based on the dominant individual and the mean of a random subgroup. Define the temperature parameter $\theta = \exp(-t / T_{\max})$, where t is the current iteration number and $T_{\max}$ is the maximum iteration number. The first $\lceil n/2 \rceil$ individuals (male hippopotamuses) of the population are updated according to the following rules:

$$x_{i,j}^{\text{Mh}} = x_{i,j} + r_2 \cdot (Dh_j - I_1 x_{i,j}), \quad i = 1, 2, \dots, \lceil n/2 \rceil \quad (11)$$

where $r_2 \in [0,1]$ is a random number, Dh_j is the coordinate of the dominant individual (current optimal solution) in the j -th dimension, and $I_1 \in \{1,2\}$ is a random integer. Female hippopotamuses and calves within the same range select different update strategies according to the value of θ :

$$x_{i,j}^{\text{FMh}} = \begin{cases} x_{i,j} + h_1 \cdot (Dh_j - I_2 M_{i,j}), & \theta > 0.6 \\ x_{i,j} + h_2 \cdot (M_{i,j} - Dh_j), & \theta \leq 0.6 \text{ 且 } r_3 > 0.5 \\ \text{lb}_j + r_4(\text{ub}_j - \text{lb}_j), & \text{else} \end{cases} \quad (12)$$

where $I_2 \in \{1,2\}$, $M_{i,j}$ is the mean of a randomly selected hippopotamus subgroup in the j -th dimension, $r_3, r_4 \in [0,1]$, and h_1, h_2 is a vector or value selected from the following five random modes:

$$h \in \{I_2 \bar{r}_5 + (1 - \rho_1), 2\bar{r}_6 - 1, \bar{r}_7, I_1 \bar{r}_8 + (1 - \rho_2), r_9\} \quad (13)$$

where $\rho_1, \rho_2 \in [0,1]$ is an integer random number, $\bar{r}_5, \bar{r}_6, \bar{r}_7, \bar{r}_8 \in \mathbb{R}^{\text{lxm}}$, $r_9 \in [0,1]$. If the fitness of the new position is better than that of the original position, the update is accepted; otherwise, the original position is retained:

$$X_i = \begin{cases} X_i^{\text{new}}, & F_i^{\text{new}} < F_i \\ X_i, & \text{else} \end{cases} \quad (14)$$

(3) Defense against predators (exploration phase)

This phase simulates the response of hippopotamuses when encountering predators. A predator position $P_j = \text{lb}_j + \bar{r}_{10} \cdot (\text{ub}_j - \text{lb}_j)$ is randomly generated. The distance $D_{i,j} = |P_j - x_{i,j}|$ between the hippopotamus and the predator is calculated. Based on the comparison between the predator's fitness F_p and the hippopotamus's fitness F_i , different update formulas are adopted:

$$X_{i,j}^{\text{HR}} = \begin{cases} \overline{RL}_{i,j} \oplus P_j + \gamma \cdot \frac{1}{D_{i,j}}, & F_p < F_i \\ \overline{RL}_{i,j} \oplus P_j + \gamma \cdot \frac{1}{2D_{i,j} + r_{11}}, & \text{else} \end{cases} \quad (15)$$

where $\overline{RL}$ is a random vector following the Lévy distribution, $\oplus$ denotes element-wise multiplication, and γ is the distance coefficient (usually taken as 0.05), $r_{11} \in [0,1]$. The greedy selection criterion is then adopted to decide whether to replace the original position.

(4) Escape from predators (exploitation phase)

When hippopotamuses are unable to defend against predators, the local refined search after escape is simulated. First, the local boundaries that shrink with iterations are defined as follows:

$$\text{lb}_j^{\text{local}} = \frac{\text{lb}_j}{t}, \text{ub}_j^{\text{local}} = \frac{\text{ub}_j}{t} \quad (16)$$

Then new positions of the hippopotamus population are generated:

$$x_{i,j}^{\text{E}} = x_{i,j} + r_{12}(\text{lb}_j^{\text{local}} + s \cdot (\text{ub}_j^{\text{local}} - \text{lb}_j^{\text{local}})) \quad (17)$$

where $s \in [-1, 1]$ is a random perturbation, $r_{12} \in [0, 1]$. Finally, the greedy selection from Equation (4) is used again to decide whether to accept the new position.

2.4. Improved Hippopotamus Optimization Algorithm

The Hippopotamus Optimization Algorithm, as a population-based optimization algorithm, has demonstrated good potential in various optimization problems since its proposal in 2024. However, when dealing with high-dimensional, multimodal, and complex constrained engineering problems, the original HO algorithm still has obvious limitations: on the one hand, its population initialization mechanism has limited coverage in high-dimensional solution spaces, which affects search diversity; on the other hand, its transition strategy between global exploration and local exploitation is relatively simplistic, which tends to cause the algorithm to fall into local optima, and its convergence accuracy and stability need to be improved. To address these issues, this paper proposes an Adaptive Improved Hippopotamus Optimization Algorithm (AIHO).

2.4.1. Chaotic Refraction Initialization of Population

The standard HO algorithm usually initializes the population by random means, which makes it difficult to ensure the uniform distribution of individuals in the solution space, thus affecting the global exploration capability of the algorithm. To improve this issue, this paper introduces the Chebyshev chaotic map [24] to initialize the hippopotamus population, making the initial position distribution more uniform and enhancing the ergodicity of the algorithm at the early stage of iterations. The expression of the Chebyshev chaotic map is as follows:

$$X_i(y+1) = \cos(k \cdot \arccos(X_i(y))), k \geq 2 \quad (18)$$

where y is the current iteration number; $X_i(y)$ is the position information of the i -th hippopotamus at the y -th iteration; and k is the chaotic order, which is usually taken as $k=4$ to obtain better chaotic characteristics.

After generating the Chebyshev chaotic sequence, it is mapped to the solution space of the original problem. To further improve the diversity of the initial population and prevent the algorithm from falling into local optima prematurely, the refraction opposite learning strategy is introduced. The refraction function is defined as follows:

$$X_i(y+1) = \frac{L_b + U_b}{2} + \frac{L_b + U_b}{2R} - \frac{X_i(y)}{R} \quad (19)$$

where L_b and U_b are the lower and upper bounds of the optimization problem, respectively; R is the refraction coefficient.

Obviously, a fixed refraction coefficient may cause the algorithm's search step size to be too large or too small, which is not conducive to balancing global exploration and local exploitation. To this end, this paper adopts a dynamic refraction coefficient to replace the fixed value R , allowing it to decrease rapidly as the number of iterations increases, thereby maintaining strong global search capability at the early stage of iterations and enhancing local refined search capability at the later stage. The expression of the dynamic refraction coefficient is as follows:

$$\omega = \exp\left(-2 \times \frac{y}{Y_m}\right) \quad (20)$$

where Y_m is the maximum number of iterations of the algorithm.

2.4.2. Dynamic Weighting Strategy

In the standard HO, the exploitation intensity remains fixed throughout the entire iteration process, leading to a decline in search capability in the later stages and making it difficult to conduct sufficient refined search near the optimal solution. To address this issue, inspired by the particle swarm optimization algorithm, this paper introduces an adaptive weight strategy, enabling the algorithm to use a larger weight factor at the early stage of iterations to enhance global search capability, and to promote local refined search through a gradually decreasing weight factor in the later stage, thereby accelerating convergence speed. The formula for the weight factor is as follows:

$$w(t) = e^{-\left(\frac{t}{t_{\max}}\right)^{0.5}} \quad (21)$$

This weight is applied to the first phase of the HO Algorithm (river/pond position update), resulting in the improved position update formula for male hippopotamuses as follows:

$$x_{i,j}^{\text{Mh}} = w(t) \cdot x_{i,j} + r_2 \cdot (Dh_j - I_1 \cdot x_{i,j}) \quad (22)$$

2.4.3. Adaptive Step Size Strategy

In the defense against predators phase, the algorithm step size is completely determined by a random mechanism, leading to unstable optimization results. In fact, at the early stage of iterations, a larger step size can expand the defensive search range of hippopotamuses, enhance global exploration capability, and help the algorithm quickly locate promising solution regions; while at the later stage, a smaller step size facilitates refined local search near the optimal solution, improving the accuracy of the final solution. Therefore, the regular variation in step size plays a guiding role in the defensive behavior of individual hippopotamuses. To this end, this paper introduces an adaptive step size control factor to dynamically adjust the Lévy step size [20]. Its expression is as follows:

$$\alpha(t) = 0.5 \times \left(1 + \cos\left(\frac{\pi t}{T_{\max}}\right) \right) \quad (23)$$

Multiply $\alpha(t)$ by the Lévy flight step size vector $\overline{RL}$, yielding the improved position update formula for the second phase:

$$X_{i,j}^{\text{HR}} = \begin{cases} \alpha(t) \cdot \overline{RL}_{i,j} \oplus P_j + \gamma \cdot \frac{1}{D_{i,j}}, & F_p < F_i \\ \alpha(t) \cdot \overline{RL}_{i,j} \oplus P_j + \gamma \cdot \frac{1}{2D_{i,j} + r_{11}}, & \text{else} \end{cases} \quad (24)$$

2.4.4. Guided Learning Strategy (GLS)

In the original HO, the third phase (escape from predators) adopts a fixed random boundary perturbation mechanism, which lacks the ability to perceive the current search state of the algorithm. As a result, in the later stages, it cannot adaptively strengthen convergence toward the global optimal solution according to the degree of population aggregation, and thus tends to fall into local optima. To address this issue, this paper introduces a Guided Learning Strategy (GLS), which accumulates the distribution information of historical optimal individuals and dynamically adjusts the update mode of individuals, thereby achieving effective following of superior solutions during the exploitation phase.

The core of GLS lies in utilizing a queue with a capacity of L (taken as $L=5$ in this paper) to store the optimal individual positions of the most recent L generations, and based on this, the population dispersion V_0 is calculated as follows:

$$V_0 = \text{std}(S_t) \cdot \frac{200}{\text{ub}_j^{\text{local}} - \text{lb}_j^{\text{local}}} \quad (25)$$

where S_t is the historical optimal position queue, and $\text{std}(\cdot)$ is the standard deviation.

To avoid premature convergence caused by excessive guidance, GLS introduces a dynamic guidance threshold C_{Gui} based on the Ebbinghaus forgetting curve [25]:

$$C_{\text{Gui}} = \left(1 - \frac{t}{T_{\text{max}}}\right) \exp\left(-\frac{t - \alpha_{\text{th}}}{T_{\text{max}}}\right) + \sum_{n=0}^k \frac{1}{\alpha_{\text{th}}} \quad (26)$$

where $\alpha_{\text{th}} = 50$ is the guidance intensity parameter, and k is the total number of times guidance has been triggered by the algorithm. This threshold dynamically adjusts the guidance frequency through a forgetting term (simulating memory decay) and an accumulation term (suppressing continuous excessive guidance).

The improved position update formula for the third phase is as follows:

$$x_{i,j}^{\text{new}} = \begin{cases} Dh_j + \tan(R\pi) \cdot \frac{\text{ub}_j^{\text{local}} - \text{lb}_j^{\text{local}}}{V_0}, & \alpha_{\text{Gui}} > C_{\text{Gui}} \\ x_{i,j} + r_{12} \cdot (\text{lb}_j^{\text{cur}} + s \cdot (\text{ub}_j^{\text{local}} - \text{lb}_j^{\text{local}})), & \text{else} \end{cases} \quad (27)$$

The flowchart and algorithm of AIHO are shown in Figure 1 and Table 1, respectively.

Table 1. The pseudocode for the AIHO is as follows.

Algorithm: Adaptive Improved Hippopotamus Optimization Algorithm

1. Input: maximum number of iterations T_{max} , hippopotamus population size N , decision variable dimension m , lower and upper bounds lb , ub .
 2. Output: global optimal solution X_{best} and its fitness value
 3. Generate the initial hippopotamus population according to Equations (18) and (19)
 4. Calculate the fitness value F_i of each hippopotamus, determine the initial dominant hippopotamus X_{best}
 5. Initialize the historical optimal queue S (with length $L = 5$)
 6. While $t < T_{\text{max}}$ do
 7. Phase 1: River/pond position update (exploration phase)
 8. For $i = 1$ to $[N/2]$ do
 9. Calculate the new position and fitness F_i^{Mh} of male hippopotamuses according to Equation (22)
 10. Update the position and fitness of the i -th hippopotamus between F_i^{Mh} and F_i according to the greedy principle
 11. End for
 12. For $i = [N/2] + 1$ to N do
 13. Calculate the new position and fitness F_i^{FMh} of female/juvenile hippopotamuses according to Equation (12)
 14. Update the position and fitness of the i -th hippopotamus between F_i^{FMh} and F_i according to the greedy principle
 15. End for
 16. Phase 2: Defense against predators (exploration phase)
-

17. For $i = 1$ to N do
18. Randomly generate the predator position P_j , and calculate the distance $D_{i,j}$
19. Calculate the new position and fitness F_i^{HR} of the i -th hippopotamus according to Equation (24)
20. Update the position and fitness of the i -th hippopotamus between F_i^{HR} and F_i according to the greedy principle
21. End for
22. Phase 3: Escape from predators (exploitation phase)
23. Calculate the local contraction boundary under the current iteration according to Equation (16)
24. Calculate the population dispersion V_0 and the guidance threshold C_{Gui} according to Equations (25) and (26), respectively
25. For $i = 1$ to N do
26. Calculate the new position and fitness F_i^{E} of the i -th hippopotamus according to Equation (27)
27. Update the position and fitness of the i -th hippopotamus between F_i^{E} and F_i according to the greedy principle
28. End for
29. Update the historical optimal queue and save the current global optimal solution
30. $t = t + 1$
31. End while
32. Output the global optimal solution X_{best} and its fitness value

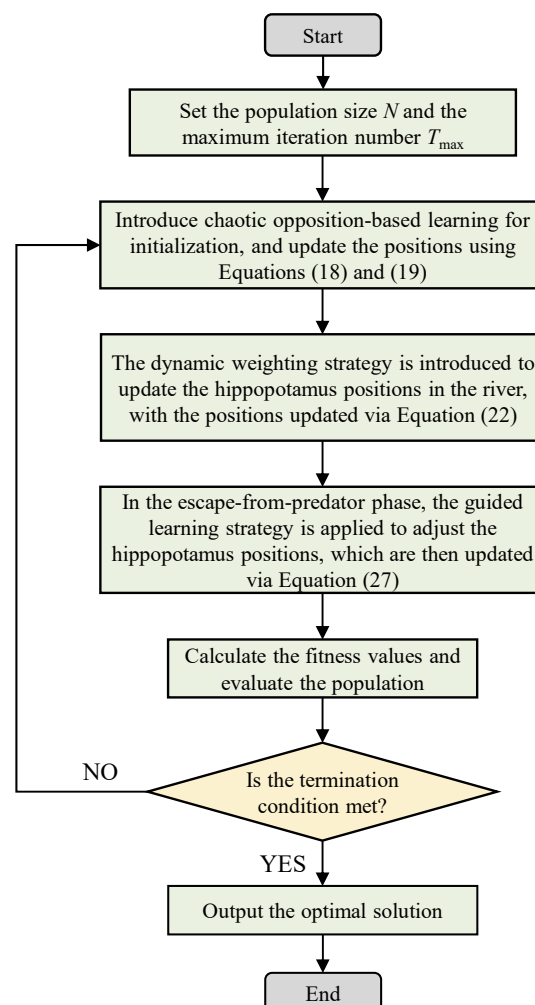


Figure 1. Flowchart of the AIHO algorithm.

3. Fault Diagnosis Model Based on ICEEMDAN–AIHO–SVM

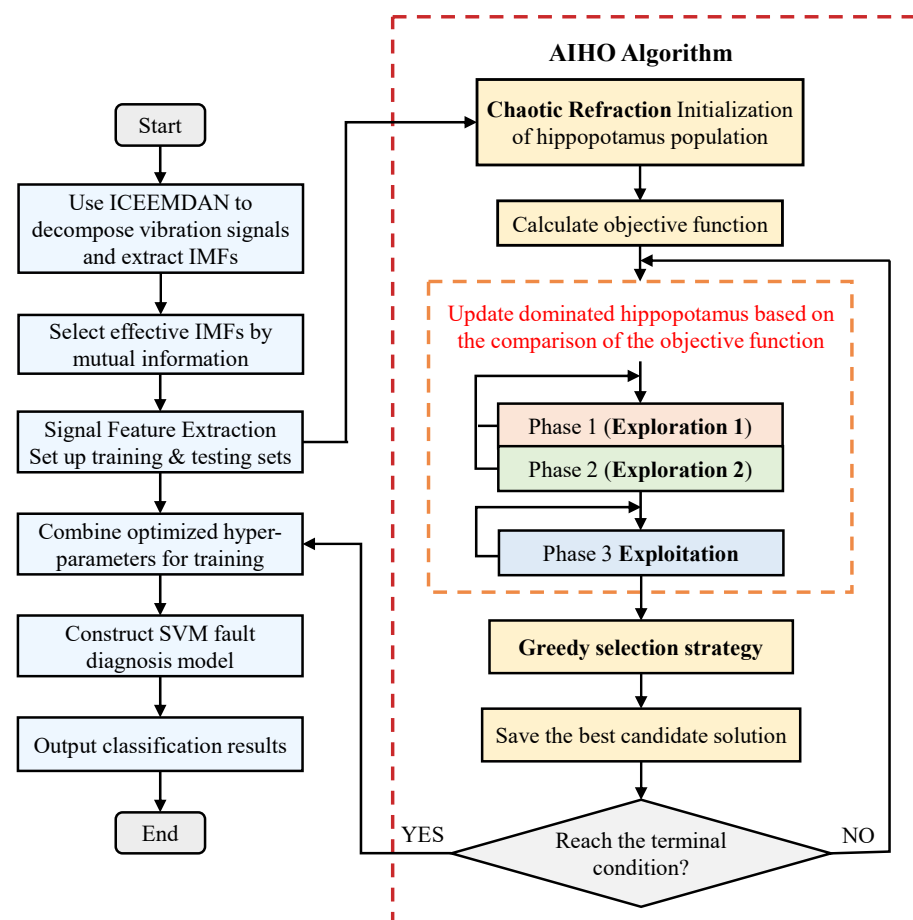
When constructing an SVM-based classification model for rolling bearing fault diagnosis, the selection of the penalty factor c and the kernel parameter g has a significant impact on the diagnostic accuracy. To mitigate this effect, this study employs the improved hippopotamus optimization algorithm to adaptively optimize (c, g) . The optimization process is shown in Figure 2, and the specific implementation steps are as follows.

1. The bearing vibration signals are segmented and randomly divided into training set and test set according to a specified ratio. The ICEEMDAN algorithm is applied to denoise and decompose the vibration signals, yielding their IMF components.

2. The IMF components with the strongest correlation to the original signal are selected based on the mutual information criterion for signal reconstruction, and nine time-domain features are extracted to obtain the fault feature vector.

3. By incorporating the chaotic refraction initialization strategy, dynamic weight strategy, adaptive step size strategy, and guided learning strategy, AIHO is used to optimize the parameters and performance of SVM, establishing the AIHO–SVM classification framework.

4. The fault feature vector is fed into the optimized AIHO–SVM model for fault classification.

**Figure 2.** Flowchart of the AIHO–SVM fault diagnosis model.

4. Experimental Results and Analysis

4.1. Performance Validation of the AIHO Algorithm

To verify the performance of the AIHO algorithm, experimental evaluations were conducted on the Matlab R2022b platform using 12 standard benchmark functions from the CEC2005 test suite [26] and the CEC2019 test suite [27]. All experiments were performed on a Windows 11 system equipped with an AMD Ryzen 7 6800H with Radeon Graphics @ 3.20 GHz processor and 16 GB RAM. Meanwhile, the AIHO algorithm was compared with the original HO algorithm, as well as PSO, whale optimization algorithm (WOA) [28], Harris hawks optimization (HHO) [29], and dung beetle optimizer (DBO) [30]. The parameter settings of the comparative algorithms are listed in Table 2.

Table 2. Parameter settings of comparative algorithms.

No.	Algorithm	Parameter Settings
1	PSO	$c_1 = c_2 = 2, w_{\max} = 0.9, w_{\min} = 0.4$
2	WOA	$b = 1, p = 0.5$
3	HHO	$E_1 = 2 - 2 \times \frac{t}{\text{Max_iter}}$
4	DBO	$k = 0.1, b = 0.3$
5	HO	$I_1, I_2 \in [1, 2]$

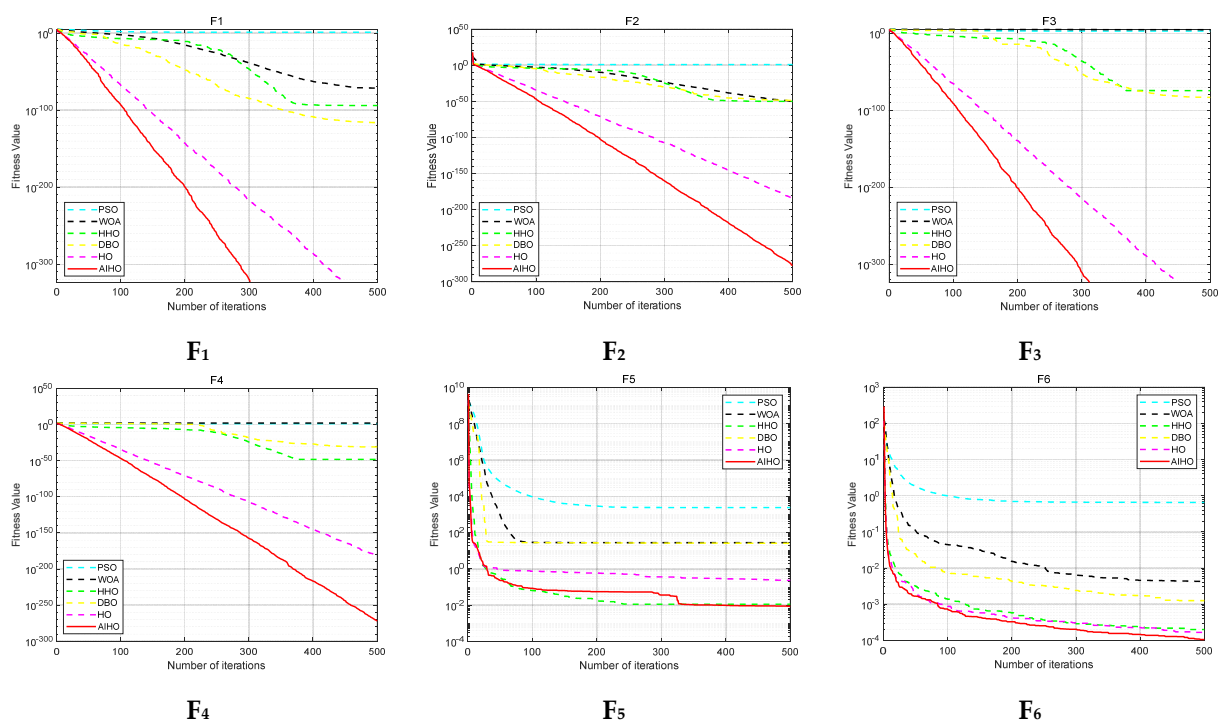
First, 12 benchmark functions from the CEC2005 test suite are used to evaluate the performance of the AIHO algorithm. The expressions of these test functions are shown in Table 3. Among them, F1–F6 are unimodal functions with only one global optimum, which are used to test the local exploitation capability of the algorithm; F7–F12 are complex multimodal functions with multiple local extrema, which are used to test the global exploration capability and the ability to escape local optima. The experimental parameters are set as follows: dimension $D = 30$, population size $N = 30$, maximum number of iterations $T_{\max} = 500$, and 30 independent runs are conducted for each function.

Table 3. CEC2005 benchmark functions.

Test Function Expression	Dimension	Search Range	Optimal Value
$F_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]$	0
$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]$	0
$F_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^l x_j \right)^2$	30	$[-100, 100]$	0
$F_4(x) = \max_i \{ x_i , 1 \leq i \leq n \}$	30	$[-100, 100]$	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]$	0
$F_6(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	30	$[-1.28, 1.28]$	0
$F_7(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]$	−418.9829 D
$F_8(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e$	30	$[-32, 32]$	0
$F_9(x) = \frac{\pi}{n} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$	30	$[-50, 50]$	0

$F_{10}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65.53, 65.53]$	0.998
$F_{11}(x) = -\sum_{i=1}^4 c_i \exp \left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right)$	6	$[0, 1]$	-3.32
$F_{12}(x) = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	$[0, 10]$	-10.56

To more intuitively demonstrate the advantages of the AIHO algorithm in terms of optimization accuracy and convergence speed, Figure 3 presents the convergence curves of different algorithms on the 12 benchmark functions. It can be observed that on unimodal functions such as F1–F4, AIHO exhibits significant convergence advantages, with its fitness value declining rapidly at the early stage of iterations and converging to the global optimum at an extremely fast speed, far outperforming the comparison algorithms such as PSO and WOA. On F7, AIHO also demonstrates outstanding optimization capability, with its final convergence accuracy being significantly higher than that of other algorithms. On complex functions such as F5, F6, and F10, AIHO maintains relatively fast convergence speed and superior solution accuracy; especially on F6, its final convergence value is notably lower than that of most comparison algorithms. On functions such as F8, F11, and F12, although the final convergence results of all algorithms are relatively close, AIHO still exhibits a faster descent rate at the early stage of convergence and can quickly stabilize near the optimal value. For F9, AIHO and some comparison algorithms can both achieve relatively good convergence accuracy. The experimental results demonstrate that the AIHO algorithm outperforms the comparison algorithms overall in terms of convergence speed and solution accuracy, and exhibits good optimization performance across different types of test functions.



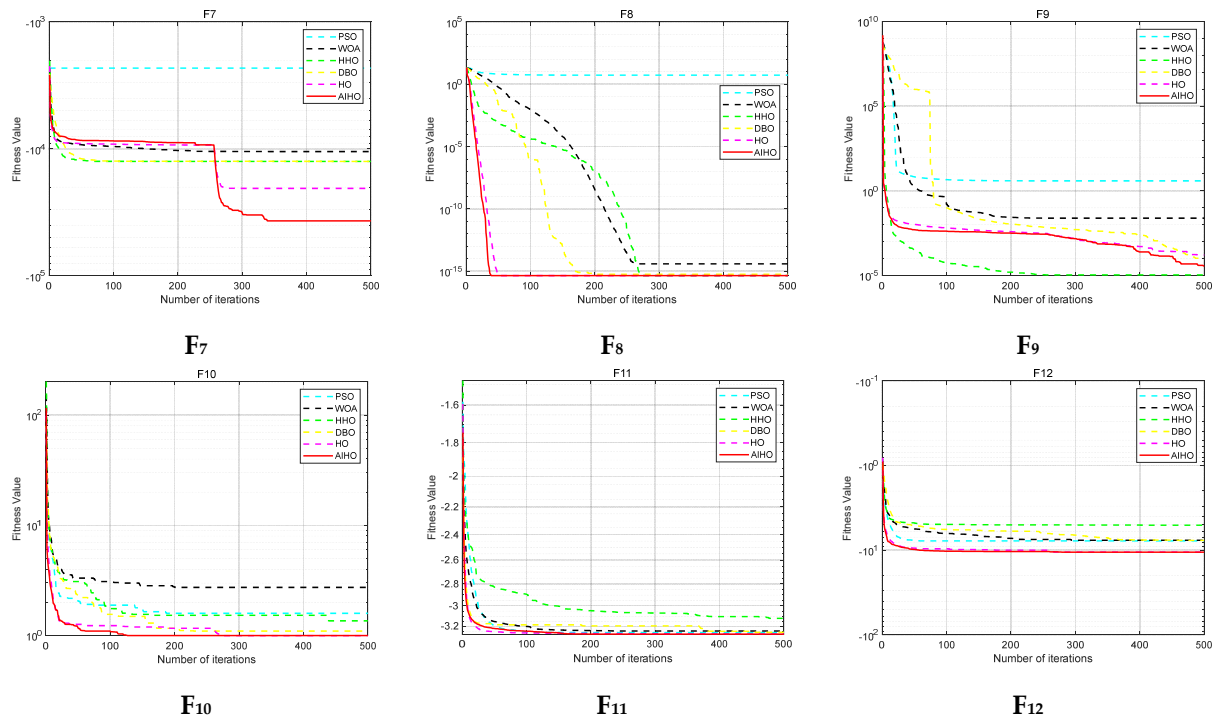


Figure 3. Convergence curves of each algorithm on the 12 benchmark functions.

From the experimental results in Table 4, it can be seen that the AIHO algorithm exhibits excellent comprehensive performance across different types of test functions. For unimodal functions (F1–F6), AIHO achieves or approaches the theoretical optimal values on F1 to F4, among which it converges with zero error on F1 and F3, and its convergence accuracy on F2 and F4 is significantly superior to all comparison algorithms. On F5 and F6, AIHO leads in terms of optimal value, mean value, and standard deviation, fully demonstrating its extremely high convergence accuracy and optimization capability on unimodal functions. For complex multimodal functions (F7–F9), AIHO achieves results far superior to other algorithms on F7, exhibiting extremely strong global search capability; on F8, it ties with several algorithms, all reaching the theoretical optimal value; on F9, it achieves the best convergence accuracy, but its mean value is slightly inferior to that of HHO. For fixed-dimensional multimodal functions (F10–F12), AIHO reaches the theoretical optimal value on F10, with its mean and standard deviation being far superior to other algorithms; on F11 and F12, it obtains results very close to the theoretical optimal solution, with the smallest standard deviation among all algorithms, demonstrating excellent stability. Overall, AIHO holds a leading position on the vast majority of test functions, verifying the effectiveness of the proposed multi-strategy improvement mechanism in enhancing the algorithm’s global optimization capability, convergence accuracy, and robustness.

Table 4. Comparison results of six algorithms on benchmark functions (bold indicates the best results).

Function	Metric	PSO	WOA	HHO	DBO	HO	AIHO
F1	Best	5.1291	8.2343×10^{-86}	4.975×10^{-115}	1.9405×10^{-163}	0	0
	Mean	10.9318	4.2309×10^{-72}	9.586×10^{-95}	7.1094×10^{-117}	0	0
	Std	4.6723	2.3168×10^{-71}	5.249×10^{-94}	3.8925×10^{-116}	0	0
F2	Best	6.876	2.8828×10^{-57}	1.515×10^{-61}	8.1555×10^{-81}	4.5079×10^{-195}	3.5197×10^{-288}
	Mean	11.6013	6.8046×10^{-51}	9.7169×10^{-51}	2.2299×10^{-49}	8.8185×10^{-185}	3.818×10^{-278}
	Std	3.1605	1.9702×10^{-50}	3.0078×10^{-50}	1.2214×10^{-48}	0	0

F3	Best	218.663	18,458.2236	1.473×10^{-97}	1.2332×10^{-131}	0	0
	Mean	764.1017	45,820.5322	5.1692×10^{-75}	8.9792×10^{-84}	0	0
	Std	444.5968	12,314.2733	2.2943×10^{-74}	4.8306×10^{-83}	0	0
F4	Best	12.6186	22.9939	5.7261×10^{-59}	2.8214×10^{-79}	1.4358×10^{-196}	2.0961×10^{-293}
	Mean	15.3909	82.0102	5.4772×10^{-49}	8.2905×10^{-32}	2.7528×10^{-182}	1.1989×10^{-272}
	Std	1.3608	21.1621	2.7334×10^{-48}	4.5398×10^{-31}	0	0
F5	Best	828.7334	27.0544	1.7486×10^{-5}	24.9798	0.0013661	7.1693×10^{-8}
	Mean	2407.4237	27.9238	0.011455	25.745	0.23373	8.4974×10^{-3}
	Std	1370.3505	0.49612	1.4926×10^{-2}	0.22823	0.48664	1.5084×10^{-2}
F6	Best	0.14233	4.2553×10^{-5}	8.2265×10^{-6}	8.0866×10^{-5}	2.5383×10^{-5}	6.7374×10^{-6}
	Mean	0.65269	0.0042775	0.00019924	0.0012571	0.00016629	0.0001.0524
	Std	0.37622	0.0071353	0.00026414	0.0010045	0.00017126	0.00011085
F7	Best	-4291.351	-12,567.055	-12,569.485	-12,569.486	-32,477.330	-98383.7799
			2	4	6	6	
	Mean	-2315.0382	-10,523.678	-12,569.076	-12,568.642	-20,512.869	-36999.2487
F8			9	6	3	4	
	Std	586.4491	1809.3132	0.61913	2.6482	3268.6457	19752.1036
	Best	3.943	4.4409×10^{-16}	4.4409×10^{-16}	4.4409×10^{-16}	4.4409×10^{-16}	4.4409×10^{-16}
F9	Mean	5.2513	3.8784×10^{-15}	4.4409×10^{-16}	5.6251×10^{-16}	4.4409×10^{-16}	4.4409×10^{-16}
	Std	0.8611	2.5523×10^{-15}	0	6.4863×10^{-16}	0	0
	Best	0.86363	0.00234	1.8575×10^{-8}	2.2683×10^{-7}	1.93×10^{-9}	6.0198×10^{-11}
F10	Mean	3.9045	0.024609	1.0995×10^{-5}	9.7483×10^{-5}	0.00016488	3.6394×10^{-5}
	Std	1.6565	0.026962	1.5282×10^{-5}	0.00032949	0.00043055	0.00011766
	Best	0.998	0.998	0.998	0.998	0.998	0.998
F11	Mean	1.591	2.7345	1.3599	1.0974	0.998	0.998
	Std	1.4083	2.933	1.2551	0.30331	3.921×10^{-13}	2.4377×10^{-13}
	Best	-3.322	-3.3212	-3.2718	-3.322	-3.322	-3.322
F12	Mean	-3.2625	-3.2465	-3.1207	-3.2604	-3.2781	-3.2784
	Std	0.060463	0.081473	0.13497	0.085343	0.063488	0.058275
	Best	-10.5364	-10.5364	-10.3761	-10.5364	-10.5364	-10.5364
	Mean	-7.2367	-7.8317	-5.2999	-8.2071	-10.5364	-10.5364
	Std	3.8702	3.2459	0.95874	2.706	4.8524×10^{-6}	4.5382×10^{-6}

In addition, to further verify the capability of the AIHO algorithm in handling complex problems, this paper also selects the CEC2019 test function suite for 30 independent experiments. The CEC2019 suite contains 10 highly challenging and complexly constructed synthetic functions, all of which are minimization problems [24]. This suite covers different types of functions ranging from unimodal to multimodal and from simple to complex, among which Q1–Q3 are unimodal functions used to examine the algorithm's convergence capability, while Q4–Q10 are multimodal functions used to test the algorithm's ability to escape local optima. These diverse function types enable the CEC2019 benchmark suite to comprehensively evaluate the overall performance of optimization algorithms. The specific information of the CEC2019 function suite is shown in Table 5. The experimental parameters are set as follows: population size $N = 30$, maximum number of iterations $T_{\max} = 500$.

Table 5. CEC2019 test functions.

NO.	Functions	Dimension	Domain	Optimal Value
Q1	Storn's Chebyshev Polynomial Fitting Problem	9	$[-8192, 8192]$	1
Q2	Inverse Hilbert Matrix Problem	16	$[-16, 384, 16, 384]$	1
Q3	Lennard–Jones Minimum Energy Cluster	18	$[-4, 4]$	1
Q4	Rastrigin's Function	10	$[-100, 100]$	1

Q5	Griewangk's Function	10	[−100,100]	1
Q6	Weierstrass Function	10	[−100,100]	1
Q7	Modified Schwefel's Function	10	[−100,100]	1
Q8	Expanded Schaffer's F6 Function	10	[−100,100]	1
Q9	Happy Cat Function	10	[−100,100]	1
Q10	Ackley Function	10	[−100,100]	1

Table 6 presents the experimental results of AIHO and the comparison algorithms on the CEC2019 test functions. It can be observed that the AIHO algorithm achieves better results than the comparison algorithms on most test functions. Specifically: on Q1, AIHO, together with PSO, HHO, DBO, and HO, reaches the theoretical optimal value of 1; on Q4, Q6, Q7, Q8, and Q9, the optimal values, mean values, and standard deviations of AIHO are significantly superior to those of other comparison algorithms; on Q2, the optimal value of AIHO is slightly inferior to that of DBO, but its mean value is far smaller than those of other comparison algorithms; on Q3, AIHO achieves the best mean value and standard deviation, with its optimal value being only slightly inferior to DBO; on Q5, AIHO has the smallest standard deviation, demonstrating the best stability; and on Q10, AIHO obtains the optimal value, and its mean value is comparable to that of HO. It can thus be concluded that AIHO also possesses certain advantages when solving complex functions.

Table 6. Comparison results of six algorithms on CEC2019 functions (bold indicates the best results).

Function	Metric	PSO	WOA	HHO	DBO	HO	AIHO
Q1	Best	1	8711.0445	1	1	1	1
	Mean	41.7744	12,132,391.6358	1	140,982.628 1	1	1
	Std	136.6927	20,433,787.7744	0	327,821.139	0	0
Q2	Best	5.3301	602.4827	5	4.2336	5	4.979
	Mean	10.9825	7338.8339	5	452.1405	5	4.9993
	Std	7.0536	3294.2427	0	1018.7096	0	0.0038396
Q3	Best	2.1042	3.7132	2.5318	1	1.4098	1.3531
	Mean	6.1715	6.6961	5.0786	4.3054	3.0534	2.8685
	Std	1.8359	1.5246	1.2318	1.7647	1.1124	0.93268
Q4	Best	17.9397	12.3751	16.1882	13.9345	10.951	9.9967
	Mean	44.8295	39.8248	40.2873	40.214	35.3821	27.9208
	Std	14.1184	17.3065	11.804	13.4408	14.1808	9.9634
Q5	Best	1.1185	1.5404	1.5155	1.0788	1.2834	1.2315
	Mean	1.543	2.3879	2.0201	1.3172	1.5502	1.5621
	Std	0.41358	0.57799	0.21092	0.24124	0.26676	0.18141
Q6	Best	6.4241	6.1427	6.63	4.2509	3.7175	3.5045
	Mean	9.6408	9.93	10.037	7.9827	7.9973	6.1084
	Std	1.6171	1.7779	1.7658	1.6618	1.7567	1.5868
Q7	Best	278.4017	587.1635	274.9168	486.9581	622.393	97.0404
	Mean	1054.8105	1274.725	957.6788	861.0514	932.0171	641.7425
	Std	268.7813	292.073	311.3657	224.3321	212.1009	225.891
Q8	Best	3.717	3.6318	3.6867	3.1575	3.4008	2.4306
	Mean	4.3867	4.4935	4.7027	4.2761	4.1746	4.0187
	Std	0.35112	0.3602	0.38	0.33389	0.29974	0.47372
Q9	Best	1.1945	1.225	1.1485	1.2194	1.1289	1.0816
	Mean	1.4301	1.5015	1.4851	1.4583	1.5066	1.3271
	Std	0.15086	0.19656	0.22986	0.14959	0.23913	0.14905

Q10	Best	21.1029	21.0172	21.0593	21.2296	21.0259	21.0051
	Mean	21.2752	21.268	21.2277	21.3908	21.0932	21.0998
	Std	0.10474	0.12449	0.10917	0.11652	0.046688	0.047757

Figure 4 presents the average convergence curves of AIHO and the comparison algorithms on selected CEC2019 test functions. It can be observed that for functions Q3, Q4, and Q6, the convergence speed of AIHO is significantly superior to that of the comparison algorithms. Benefiting from its strong ability to escape local optima, the convergence curve of AIHO declines sharply within the first 100 iterations, demonstrating excellent convergence performance. For functions Q7 and Q8, AIHO maintains a favorable downward trend throughout the iteration process, ultimately converging to a higher accuracy than the other comparison algorithms. For function Q9, the convergence advantage of AIHO is relatively modest, but it still exhibits the best convergence performance among all comparison algorithms.

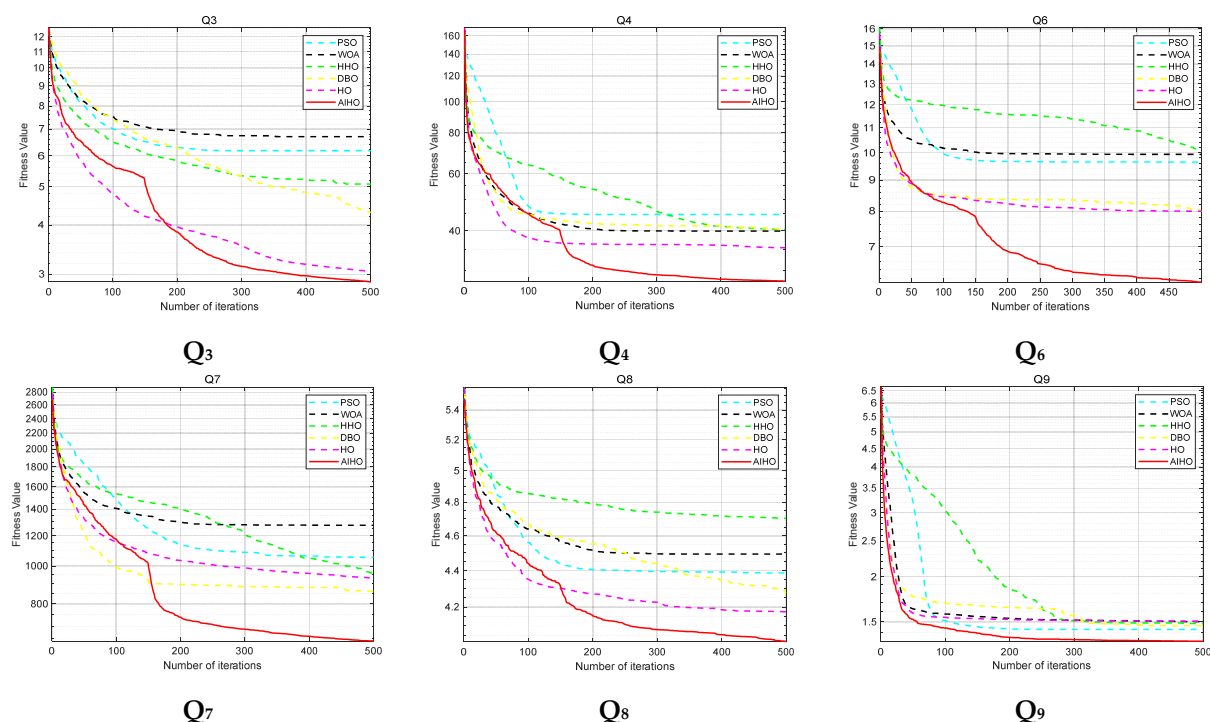


Figure 4. Convergence curves of AIHO algorithm and comparison algorithm.

4.2. Validation on the CWRU Bearing Dataset

To verify the effectiveness of the AIHO-SVM model, the Case Western Reserve University (CWRU) rolling bearing dataset [31] is adopted for validation. Vibration signals are collected from the accelerometer mounted on the drive end, with a sampling frequency of 12 kHz and a motor speed of 1797 r/min. The bearing type used in the experiment is SKF6205, and single-point faults are introduced by means of electric discharge machining, with fault sizes of 0.1778 mm, 0.3556 mm, and 0.5334 mm. The dataset includes a normal condition and three fault types, namely inner race fault, outer race fault, and rolling element fault, each corresponding to three damage sizes, resulting in a total of 10 health states (1 normal category and 9 fault categories).

The duration of each data acquisition ranges from 10 to 20 s, and 2048 data points are extracted to form one sample. A total of 120 samples are generated for each fault category,

of which 70% are used as the training set and 30% as the test set. The entire dataset comprises a total of 1200 samples, with 840 samples in the training set and 360 samples in the test set, as detailed in Table 7.

Table 7. Fault type configuration of the CWRU bearing dataset.

Bearing Condition	Fault Size/mm	Label	Number of Samples
Normal	None	1	120
	0.1778	2	120
	0.3556	3	120
	0.5334	4	120
Rolling element fault	0.1778	5	120
	0.3556	6	120
	0.5334	7	120
Outer race fault	0.1778	8	120
	0.3556	9	120
	0.5334	10	120

For the partitioned training set and test set, ICEEMDAN is applied to decompose the original vibration signals, and the decomposition results are shown in Figure 5. In the figure, the first subplot displays the original noisy vibration signal, and the remaining subplots sequentially show the decomposed IMF1 to IMF11. It can be observed that the lower-order IMF components (IMF1–IMF3) mainly contain high-frequency noise components, while the higher-order IMF components (IMF8–IMF11) reflect the low-frequency trends of the signal. The fault-related impulse features are primarily concentrated in the middle-order IMF components (IMF4–IMF7). Subsequently, the mutual information between each IMF component and the original signal is calculated according to the mutual information criterion, and the components with the strongest correlation are selected for subsequent analysis.

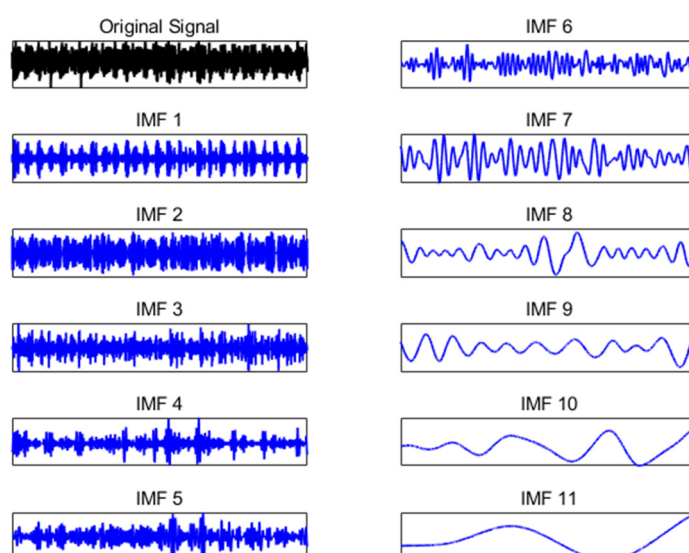


Figure 5. Schematic diagram of ICEEMDAN signal decomposition.

From the optimal IMF components selected by the mutual information criterion for each sample, a total of nine time-domain statistical features are extracted, including mean, variance, peak-to-peak value, kurtosis, root mean square value, crest factor, impulse factor, shape factor, and margin factor. These features are concatenated into a 9-dimensional

fault feature vector to reflect the energy, fluctuation, impact characteristics, and distribution morphology of the signal. An AIHO-SVM classification model is constructed using the SVM parameters obtained through AIHO iterative optimization, and the extracted features are classified by this model. The classification results are shown in Figure 6.

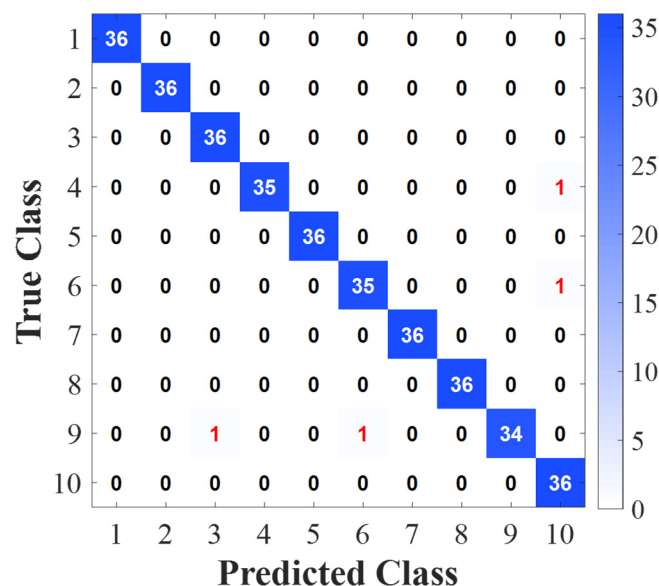


Figure 6. Confusion matrix of the proposed method.

Figure 6 presents the confusion matrix of the proposed method in the 10-class fault diagnosis task, where the horizontal axis denotes the predicted classes and the vertical axis denotes the true classes. The diagonal elements represent the number of correctly classified samples, while the off-diagonal elements represent the number of misclassified samples, with the color depth reflecting the number of samples. The proposed method achieves an accuracy of 98.89%, with only four samples misclassified into other categories. Among them, Class 9 is the most frequently confused category, with two samples misclassified as Class 3 and Class 6, respectively. The misclassified samples are scattered across different operating conditions, indicating that the confusion stems from inherent physical similarities between certain fault types rather than systematic classification failures. ICEEMDAN effectively highlights the signal features, while the SVM classifier optimized by AIHO constructs a clear decision boundary in the feature space, thereby achieving high accuracy and strong robustness.

Since the optimization results of swarm intelligence algorithms vary considerably with each iteration, the classification accuracy of each test may deviate accordingly. Therefore, under the same experimental conditions, each of the six methods is independently run five times, and the average classification accuracy is calculated for comparison, so as to ensure the feasibility of the model and the reliability of the comparison. The accuracy comparison among the six models is shown in Figure 7.

Figure 7 presents a bar chart comparing the classification accuracies of six different models in the fault diagnosis task, where the horizontal axis denotes the six model combinations and the vertical axis denotes the classification accuracy, with error bars representing the standard deviation over five repeated experiments. Comparing the accuracies of the first three optimization-based models, the proposed method achieves an average accuracy of 99.07%, demonstrating the best performance and relatively high stability; ICEEMDAN-HO-SVM ranks second; and ICEEMDAN-PSO-SVM has the lowest average accuracy among the three, though its error bars are relatively small. The comparison results of these three methods confirm the optimization advantages of AIHO, which

achieves the highest accuracy while maintaining good stability. The three optimized SVM models significantly outperform the unoptimized ICEEMDAN–SVM, indicating that the introduction of intelligent optimization algorithms plays a critical role in improving SVM classification performance. This conclusion is further supported by the comparison between EMD–AIHO–SVM and EMD–SVM, both using EMD as input. Moreover, under the same optimization algorithm and classifier conditions, the proposed method achieves higher classification accuracy than EMD–AIHO–SVM, demonstrating that signals decomposed by ICEEMDAN are superior to those decomposed by EMD in terms of highlighting fault features.

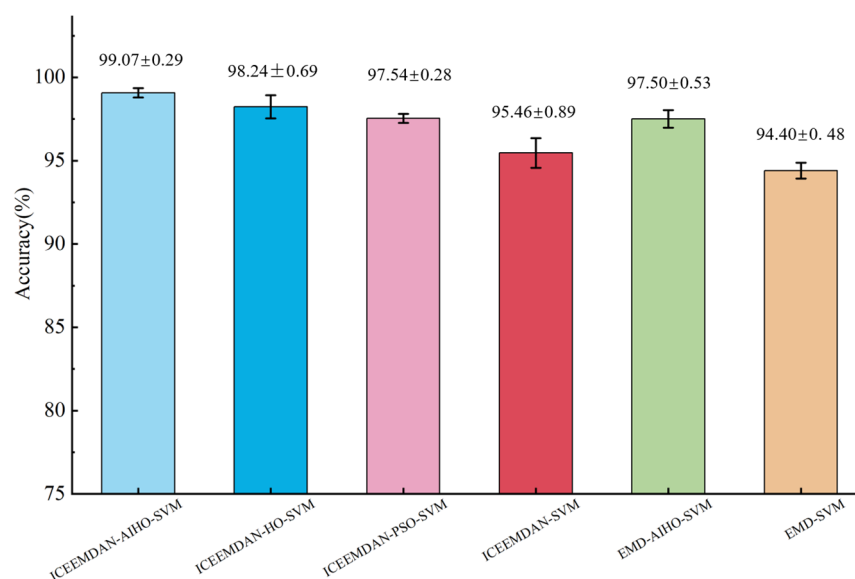


Figure 7. Comparison of average classification accuracy among six models.

Table 8 presents the specific classification accuracy data of different models. The proposed method achieves an average accuracy of 99.07%, demonstrating the best performance and relatively high stability. Among the three optimization algorithms, AIHO shows the best performance, improving accuracy by 0.83 and 1.53 percentage points over HO and PSO, respectively, with a relatively small standard deviation. Further comparison between models with and without optimization algorithms shows that after introducing AIHO, HO, and PSO into ICEEMDAN–SVM, the accuracies increase by 3.61, 2.78, and 1.08 percentage points, respectively; after introducing AIHO into EMD–SVM, the accuracy increases by 3.10 percentage points, indicating that regardless of the decomposition method adopted, optimization algorithms can effectively enhance feature separability. Meanwhile, ICEEMDAN–AIHO–SVM outperforms EMD–AIHO–SVM by 1.57 percentage points, and ICEEMDAN–SVM outperforms EMD–SVM by 1.06 percentage points, demonstrating that ICEEMDAN can more effectively suppress mode mixing compared with traditional EMD, thereby highlighting fault features more effectively.

Table 8. Average classification accuracy and standard deviation of six models.

Model	ICEEMDAN-AI HO-SVM	ICEEMDAN-HO- SVM	ICEEMDAN-PSO- SVM	ICEEMDAN-S VM	EMD-AIHO- SVM	EMD-SVM
Accuracy	99.07 ± 0.29	98.24 ± 0.69	97.54 ± 0.28	95.46 ± 0.89	97.50 ± 0.53	94.40 ± 0.48

In summary, the proposed method, through the synergy between ICEEMDAN and AIHO, enables the model to achieve high accuracy, verifying that the joint strategy of

adaptive decomposition and intelligent feature optimization delivers optimal performance in fault diagnosis.

4.3. Validation on the HUST Bearing Dataset

To verify the generalization performance of the proposed method on different datasets, the bearing dataset released by Huazhong University of Science and Technology (HUST) [32] is selected for further validation. This dataset was collected by Professor Shen Weiming's team using the Spectra-Quest mechanical fault simulator. The experimental platform includes a speed controller, motor, shaft, accelerometer, bearings, and a data acquisition card. The sampling frequency is 25.6 kHz, and vibration signals under nine health states (normal, moderate/severe inner race fault, moderate/severe outer race fault, moderate/severe rolling element fault, and moderate/severe combined inner and outer race fault) are collected.

The sampling frequency of the dataset is 25.6 kHz, and the original signals under each operating condition contain acceleration vibration signals in three directions, namely X, Y, and Z. In the data preprocessing stage, 5120 data points are extracted from the signal in each direction as one data segment, with the sliding step set to 1280. A total of 190 samples are generated for each fault condition, covering 9 different operating conditions. The complete dataset comprises a total of 1710 samples, of which 70% are used as the training set and 30% as the test set, as detailed in Table 9.

Table 9. Dataset partition.

Fault Category	Label	Samples
Moderate Ball Fault	1	190
Moderate Compound Fault	2	190
Moderate Inner Race Fault	3	190
Moderate Outer Race Fault	4	190
Severe Ball Fault	5	190
Severe Compound Fault	6	190
Normal	7	190
Severe Inner Race Fault	8	190
Severe Outer Race Fault	9	190

Consistent with the above processing steps, for the partitioned training set and test set, ICEEMDAN is applied to denoise and decompose the vibration signals, and the IMF components with the strongest correlation to the original signal are selected according to the mutual information criterion. From the optimal IMF components selected by the mutual information criterion for each sample, nine time-domain statistical features are extracted and concatenated into a 9-dimensional fault feature vector. The extracted features are then fed into the constructed AIHO-SVM classification model, and the classification results are shown in Figure 8.

Figure 8 presents the confusion matrix of the proposed method in this fault diagnosis task. The results show that the proposed method exhibits excellent fault identification capability under multi-class complex operating conditions, achieving a classification accuracy of 99.22%. Only four samples are misclassified into other categories, and the misclassified samples are extremely few and scattered across different classes, with no systematic class confusion observed. Specifically, one sample from Class 5 (Severe Ball Fault) and one sample from Class 9 (Severe Outer Race Fault) are mutually misclassified, forming a bidirectional confusion, which is mainly attributed to the physical similarity of impact characteristics between ball faults and outer race faults at certain rotational speeds; the remaining two misclassified samples involve Class 7 (Normal) and Class 8 (Severe Inner Race

Fault), respectively, and are distributed under different operating conditions, further corroborating the randomness and non-systematic nature of the misclassifications, rather than any structural identification deficiency of the classifier itself.

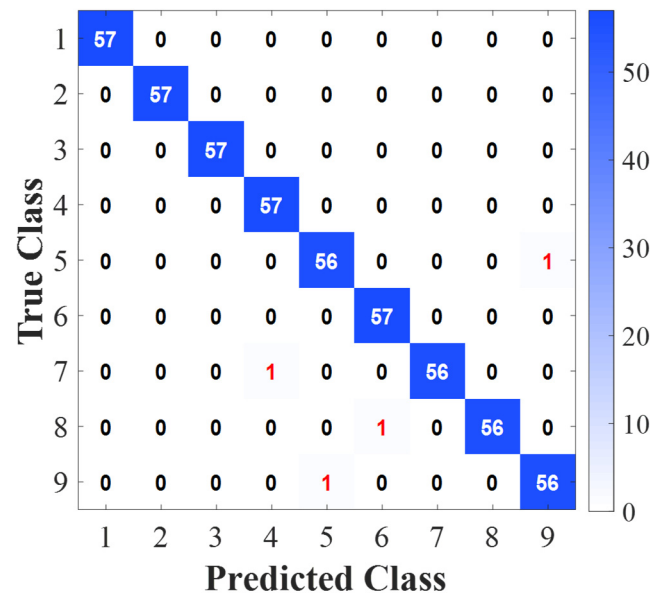


Figure 8. Confusion matrix of the proposed method on the HUST dataset.

Similarly, five other classification models are compared on this dataset, and five independent trials are conducted for each model, with the average classification accuracy and error bars shown in Figure 9. As can be seen from the figure, the proposed method achieves the best performance and the highest stability, while the models optimized by HO and PSO have lower classification accuracy and larger error bars compared with the proposed method, thus demonstrating the advantage of AIHO in optimizing SVM performance. In addition, compared with the ICEEMDAN-SVM model without optimization, the classification accuracy is improved after introducing intelligent optimization algorithms, indicating that optimizing SVM parameters can effectively enhance its classification performance. Furthermore, compared with models using EMD for feature extraction, under the same classifier conditions, adopting ICEEMDAN decomposition can better suppress mode mixing and remove noise interference, thereby achieving superior classification performance over EMD.

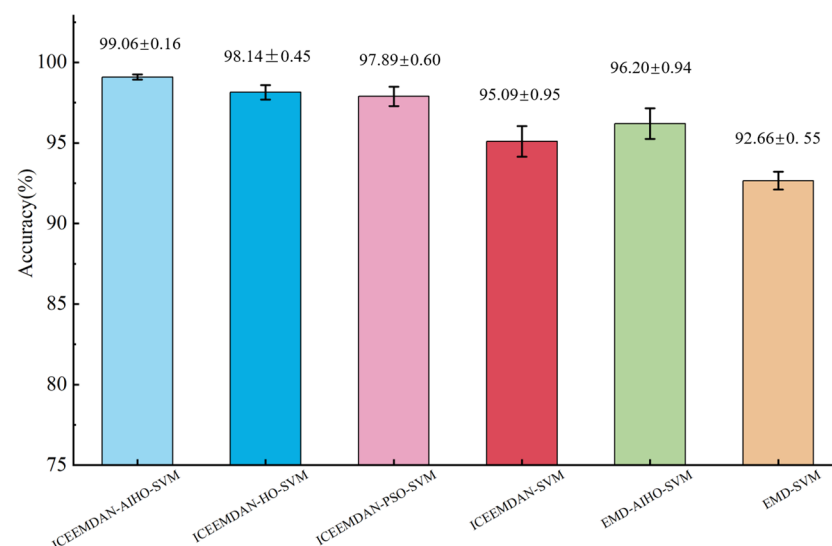


Figure 9. Comparison of average classification accuracy among six models on the HUST dataset.

Table 10 presents the specific classification accuracy data of different models. The proposed method achieves an average accuracy of 99.09%, demonstrating the best performance and the smallest variation in error. Comparing the results of the first three different optimization processes, the AIHO-based optimization yields the best classification performance, improving the accuracy by 0.95 and 1.20 percentage points over HO and PSO, respectively, with the smallest standard deviation. The comparison results of these three methods confirm the optimization advantage of AIHO, which achieves the highest accuracy while providing the best stability. All three optimized SVM models outperform the unoptimized ICEEMDAN-SVM method, demonstrating that the introduction of intelligent optimization algorithms plays a key role in enhancing SVM classification performance, which is also corroborated by the comparison between EMD-AIHO-SVM and EMD-SVM using EMD as input. Moreover, under the same optimization algorithm and classifier conditions, the proposed method achieves a classification accuracy 2.89% higher than that of the EMD-AIHO-SVM model, indicating that signals decomposed by ICEEMDAN are superior to those by EMD in highlighting fault features and suppressing mode mixing. In summary, through the synergy of ICEEMDAN and AIHO, the proposed method can accomplish bearing fault classification tasks with high accuracy.

Table 10. Average classification accuracy and standard deviation of six models on the HUST dataset.

Model	ICEEMDAN-AI HO-SVM	ICEEMDAN-HO- SVM	ICEEMDAN-PSO- SVM	ICEEMDAN-S VM	EMD-AIHO- SVM	EMD-SVM
Accuracy	99.09 ± 0.16	98.14 ± 0.45	97.89 ± 0.60	95.09 ± 0.95	96.20 ± 0.94	92.66 ± 0.55

5. Conclusions

To address the difficulty of fault feature extraction from rolling bearings under strong background noise, this paper proposes a fault diagnosis method based on the combination of ICEEMDAN and AIHO-SVM, and the following conclusions are drawn:

(1) To overcome the shortcomings of the traditional hippopotamus optimization algorithm, such as insufficient population diversity, slow convergence, and susceptibility to local optima, this paper introduces four improvement mechanisms, namely Chebyshev chaotic mapping combined with refraction opposite learning, dynamic weight strategy, adaptive step size strategy, and guided learning strategy, and proposes AIHO. Performance tests on 12 benchmark functions and the CEC2019 test suite demonstrate that AIHO outperforms the compared algorithms in terms of solution accuracy, convergence speed, and stability.

(2) For rolling bearing vibration signals, the ICEEMDAN algorithm is employed for noise reduction and decomposition, and the feature vector is constructed accordingly. On this basis, the AIHO algorithm is used to adaptively optimize the penalty factor and kernel parameter of SVM, establishing the AIHO-SVM fault diagnosis model.

(3) The proposed model is validated on two bearing datasets. The results show that the proposed method achieves average diagnostic accuracies of 99.07% and 99.09% on the two datasets, respectively, which are superior to those of other comparison models. It can effectively identify fault categories, fully verifying the superiority and stability of the proposed algorithm.

Although the proposed method achieves high diagnostic accuracy on the datasets, several limitations still remain. The experimental data are collected from test rigs under simulated operating conditions, and thus complex factors in real industrial environments, such as load fluctuations, speed variations, strong noise, and gradual fault evolution, have

not been fully considered. Moreover, both ICEEMDAN decomposition and AIHO parameter optimization involve substantial computational costs, and their combination significantly increases the overall computational burden, making the method more suitable for offline diagnosis and periodic health assessment at the current stage. Future work will focus on validation under real industrial scenarios, as well as on developing lightweight decomposition strategies for ICEEMDAN and fast optimization mechanisms for AIHO, so as to reduce resource consumption and improve real-time performance.

Author Contributions: L.W.: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Visualisation, Writing—original draft, Writing—review & editing. Y.Z.: Data curation, Formal analysis, Software, Visualisation, Writing—review & editing. G.X.: Data curation, Investigation, Resources, Writing—review & editing. Y.C.: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing—original draft, Writing—review & editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Tackling Project of Henan Province, grant number 252102211052.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVM	Support vector machine
IMF	Intrinsic mode function
AIHO	Adaptive Improved Hippopotamus Optimization Algorithm
EMD	Empirical Mode Decomposition
EEMD	Ensemble Empirical Mode Decomposition
CEEMDAN	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
ICEEMDAN	Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
PSO	Particle swarm optimization
SOA	Seagull optimization algorithm
HO	Hippopotamus Optimization Algorithm
WOA	Whale optimization algorithm
HHO	Harris hawks optimization
DBO	Dung beetle optimizer
CWRU	Case Western Reserve University
HUST	Huazhong University of Science and Technology

References

1. Zhang, Y.; Lv, Y.; Ge, M. Time–frequency analysis via complementary ensemble adaptive local iterative filtering and enhanced maximum correlation kurtosis deconvolution for wind turbine fault diagnosis. *Energy Rep.* **2021**, *7*, 2418–2435.
2. Martin, G.; Becker, F.M.; Kirchner, E. A novel method for diagnosing rolling bearing surface damage by electric impedance analysis. *Tribol. Int.* **2022**, *170*, 107503.
3. Rai, A.; Upadhyay, S.H. A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings. *Tribol. Int.* **2016**, *96*, 289–306.
4. Song, Q.; Jiang, X.; Du, G.; Liu, J.; Zhu, Z. Smart multichannel mode extraction for enhanced bearing fault diagnosis. *Mech. Syst. Signal Process.* **2023**, *189*, 110107.

5. Song, S.; Zhang, S.; Dong, W.; Zhang, X.; Ma, W. A new hybrid method for bearing fault diagnosis based on CEEMDAN and ACPSO-BP neural network. *J. Mech. Sci. Technol.* **2023**, *37*, 5597–5606.
6. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41.
7. Torres, M.E.; Colominas, M.A.; Schlotthauer, G.; Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech And Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2011; pp. 4144–4147.
8. Colominas, M.A.; Schlotthauer, G.; Torres, M.E. Improved complete ensemble EMD: A suitable tool for biomedical signal processing. *Biomed. Signal Process. Control* **2014**, *14*, 19–29.
9. Li, T.; Qian, Z.; He, T. Short-Term Load Forecasting With Improved CEEMDAN and GWO-Based Multiple Kernel ELM. *Complexity* **2020**, *2020*, 1209547.
10. Gao, L.; Gu, Y.; Chen, C.; Zhang, P.; Zhang, Z. Wind turbine gearbox bearing fault diagnosis method based on ICEEMDAN and flexible wavelet threshold. *J. Fail. Anal. Prev.* **2024**, *24*, 1181–1198.
11. Chen, Z.; Zhu, L.; Lu, H.; Chen, S.; Zhu, F.; Liu, S.; Han, Y.; Xiong, G. Research on bearing fault diagnosis based on improved genetic algorithm and BP neural network. *Sci. Rep.* **2024**, *14*, 15527.
12. Liu, A.; Yang, Z.; Li, H.; Wang, C.; Liu, X. Intelligent diagnosis of rolling element bearing based on refined composite multiscale reverse dispersion entropy and random forest. *Sensors* **2022**, *22*, 2046.
13. Li, X.; An, S.; Shi, Y.; Huang, Y. Remaining useful life estimation of rolling bearing based on SOA-SVM algorithm. *Machines* **2022**, *10*, 729.
14. Cui, M.; Wang, Y.; Lin, X.; Zhong, M. Fault diagnosis of rolling bearings based on an improved stack autoencoder and support vector machine. *IEEE Sens. J.* **2020**, *21*, 4927–4937.
15. Liu, M.; Tong, W.; Li, Y. Rolling Bearing Fault Diagnosis Based on SABO-VMD-IBES-SVM. *Shock Vib.* **2026**, *2026*, 9958568.
16. Yan, X.; Jia, M. A novel optimized SVM classification algorithm with multi-domain feature and its application to fault diagnosis of rolling bearing. *Neurocomputing* **2018**, *313*, 47–64.
17. Fan, Q.; Yu, F.; Xuan, M. Transformer fault diagnosis method based on improved whale optimization algorithm to optimize support vector machine. *Energy Rep.* **2021**, *7*, 856–866.
18. Zhang, X.; Wang, H.; Ren, M.; He, M.; Jin, L. Rolling bearing fault diagnosis based on multiscale permutation entropy and SOA-SVM. *Machines* **2022**, *10*, 485.
19. Ding, W.; Zheng, J.; Li, J.; Pan, H.; Cheng, J.; Tong, J. Fault Diagnosis of Rolling Bearing Based on Two-dimensional Composite Multi-scale Ensemble Gramian Dispersion Entropy. *Chin. J. Mech. Eng.* **2025**, *39*, 100065.
20. Amiri, M.H.; Hashjin, N.M.; Montazeri, M.; Mirjalili, S.; Khodadadi, N. Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm. *Sci. Rep.* **2024**, *14*, 5032.
21. Han, T.; Wang, H.; Li, T.; Liu, Q.; Huang, Y. MHO: A modified hippopotamus optimization algorithm for global optimization and engineering design problems. *Biomimetics* **2025**, *10*, 90.
22. Minocha, S.; Sharma, S.R.; Singh, B.; Gandomi, A.H. Adaptive image encryption approach using an enhanced swarm intelligence algorithm. *Sci. Rep.* **2025**, *15*, 9476.
23. Cervantes, J.; Garcia-Lamont, F.; Rodríguez-Mazahua, L.; Lopez, A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing* **2020**, *408*, 189–215.
24. Tavazoei, M.S.; Haeri, M. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. *Appl. Math. Comput.* **2007**, *187*, 1076–1085.
25. Jia, H.; Lu, C. Guided learning strategy: A novel update mechanism for metaheuristic algorithms design and improvement. *Knowl.-Based Syst.* **2024**, *286*, 111402.
26. Liang, J.J.; Suganthan, P.N.; Deb, K. Novel composition test functions for numerical global optimization. In *Proceedings of the 2005 IEEE Swarm Intelligence Symposium; SIS 2005*; IEEE: Piscataway, NJ, USA, 2005; Volume 2005, pp. 68–75.
27. Price, K.V.; Awad, N.H.; Ali, M.Z.; Suganthan, P.N. *Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization*; Technical report; Nanyang Technological University: Singapore, 2018.
28. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67.
29. Heidari, A.A.; Mirjalili, S.; Faris, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872.

30. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *Supercomputing* **2023**, *79*, 7305–7336.
31. Smith, W.A.; Randall, R.B. Rolling element bearing diagnostics using the Case Western Reserve University data: A benchmark study. *Mech. Syst. Signal Process.* **2015**, *64*, 100–131.
32. Zhao, C.; Zio, E.; Shen, W. Domain generalization for cross-domain fault diagnosis: An application-oriented perspective and a benchmark study. *Reliab. Eng. Syst. Saf.* **2024**, *245*, 109964.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.