Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

IBKA-Transformer: An Improved Black Kite Algorithm Optimized Transformer for Industrial Robot Bearing Fault Diagnosis

Jiangnan Su
Hunan Railway Professional Technology College
Jun Huang
757969703@qq.com

Hunan Railway Professional Technology College

Article

Keywords: Industrial robot, Bearing, Black kite optimization algorithm, Transformer model, Feature extraction, Intelligent diagnosis

Posted Date: July 27th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10306908/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

IBKA-Transformer: An Improved Black Kite Algorithm Optimized Transformer for Industrial Robot Bearing Fault Diagnosis

Jiangnan Su, Jun Huang

Hunan Railway Professional Technology College, Zhuzhou 412001, China;
Email:155308147@qq.com
Corresponding author: " Jun Huang "
Email: 757969703@qq.com

Abstract: To address the insufficient accuracy of traditional methods in industrial robot bearing fault diagnosis, this study proposes a novel artificial intelligence–based diagnostic method that optimizes a Transformer model using an improved Black Kite Algorithm (IBKA). First, to overcome the drawbacks of the original Black Kite Algorithm—namely unstable convergence, susceptibility to local optima, and low efficiency in handling mixed parameter spaces—the algorithm is improved from three aspects: an adaptive nonlinear convergence factor is introduced to balance exploration and exploitation; a dynamic step-size adjustment mechanism is designed to enhance convergence stability; and a hybrid encoding strategy is proposed to efficiently handle mixed continuous–discrete hyperparameter spaces. Next, the IBKA is integrated with the Transformer model to construct an end-to-end fault diagnosis framework. Multi-dimensional feature engineering is employed to extract vibration signal features, while IBKA is used to simultaneously optimize Transformer hyperparameters and perform feature selection. The optimized Transformer then completes the fault classification task. Finally, experimental validation is conducted on a self-constructed industrial robot bearing fault dataset covering multiple fault types and operating conditions. The experimental results demonstrate that the proposed method achieves a diagnostic accuracy of 99.86%, significantly outperforming comparative methods such as ACGAN, SVM, and CNN. Ablation experiments further confirm the effectiveness of the proposed improvement strategies.

Keywords: Industrial robot; Bearing; Black kite optimization algorithm; Transformer model; Feature extraction; Intelligent diagnosis

1. Introduction

With the widespread adoption of industrial robots in high-end manufacturing, operational reliability and intelligent maintenance have become key factors affecting equipment safety and production efficiency. Rolling bearings are critical yet vulnerable components in the joint transmission system. They are subjected to frequent starts and stops, variable-speed operation, and complex coupled loads. Once local damage occurs, the fault response is rapidly amplified through vibration and propagates across the entire machine [1-2]. Compared with constant-speed rotating machinery, the vibration signals of industrial robot bearings exhibit stronger non-stationarity and nonlinearity. Fault impulses and modulation patterns undergo stretching, compression, and drift with changing operating conditions. Environmental noise and multi-source interference further complicate the problem. These factors significantly increase the difficulty of extracting stable and informative diagnostic features [3-4]. Therefore, reliable extraction and intelligent identification of bearing fault information under complex and noisy conditions has become a core challenge in industrial robot condition monitoring.

Early studies on rolling-bearing fault diagnosis mainly relied on signal processing and

handcrafted feature engineering. Time-domain statistics, spectral features, and envelope analysis indicators were extracted and then combined with models such as support vector machines for fault recognition [5-6]. However, these methods depend heavily on expert experience and parameter settings. When operating conditions or noise levels change, feature distributions drift easily and diagnostic performance degrades markedly [7]. To reduce such dependence, Zhang et al. introduced convolutional neural networks (CNN) for bearing diagnosis. Their results showed that CNNs can automatically learn local impact and modulation patterns and improve multi-class recognition performance. Multi-scale convolutional structures further enhance feature extraction capability [8]. Nevertheless, convolution mainly models local receptive fields. Under variable-speed and variable-load conditions in industrial robots, cross-scale and non-periodic behaviors become prominent. CNNs therefore struggle to characterize these patterns, which limits model stability.

To improve robustness under complex operating conditions, recent research has explored two directions: data augmentation and sequence modeling [9]. On one hand, Zhao et al. employed an auxiliary-classifier generative adversarial network to expand small training sets. This approach improves training sufficiency and recognition accuracy to some extent. However, convergence instability and limited controllability of generated samples remain, so such methods are better suited as auxiliary tools [10]. On the other hand, Transformer architectures have been introduced for non-stationary vibration analysis because self-attention is well suited for modeling global dependencies. These models have shown strong potential for capturing long-range correlations [11-12]. Even so, Transformer performance is highly sensitive to hyperparameter settings and input organization. If tuning still relies on manual experience, it is difficult to obtain stable and repeatable optimal configurations in high-dimensional search spaces. Meta-heuristic optimization algorithms have therefore been applied for automated hyperparameter tuning. Among them, the Black-Winged Kite Algorithm (BKA) has shown advantages in complex non-convex optimization. Yet, in high-dimensional and mixed search spaces, it may still suffer from unstable convergence and premature stagnation [13-14]. Moreover, most existing studies focus on optimizing a single category of parameters. Few jointly consider feature selection and model hyperparameters. This limitation may lead to redundant features and weak generalization, highlighting the need for a collaborative optimization framework [15-16].

Motivated by these challenges, this study develops a Transformer-based bearing fault diagnosis framework driven by an improved BKA, targeting complex industrial-robot operating conditions. First, an adaptive nonlinear regulation mechanism and a dynamic update strategy are introduced to strengthen global search capability. Second, a multidimensional feature-sequence input structure is constructed according to the characteristics of robot bearing vibration signals, enabling joint representation of fault information. Third, a hybrid encoding strategy is employed to achieve coordinated optimization of key Transformer hyperparameters and feature subsets. Finally, comparative experiments and ablation studies are conducted on industrial-robot bearing vibration datasets to systematically evaluate the proposed framework.

2. Methods

2.1 Black kite optimization algorithm

The Black Kite Optimization Algorithm (BKA) is a metaheuristic optimization method inspired by the predatory behavior of black kites in nature. It simulates intelligent behaviors such as hovering observation, diving attacks, and adaptive strategy adjustment, and searches for the optimal solution through an alternating mechanism between global exploration and local

exploitation [17]. First, the population is initialized by uniformly distributing the positions of black kites, where the position of each black kite is expressed in Eq. (1).

$$X_i = BK_{lb} + rand\left(BK_{ub} - BK_{lb}\right) \tag{1}$$

where X_i denotes the new position of the *i*-th individual; Bk_{lb} represents the current position of the current individual (i.e., the *i*-th black kite); BK_{ub} and BK_{lb} denote the positions of two other randomly selected black kites; and $rand$ denotes a random number uniformly distributed in [0,1].

During the initialization process, BKA selects the individual with the best fitness value as the initial leader X_L, which is regarded as the optimal position of the black kite. The initial leader position is expressed in Eq. (2).

$$\begin{aligned} f_{best} &= \min\left(f\left(X_i\right)\right) \\ X_L &= X\left(find\left(f_{best} == f\left(X_i\right)\right)\right) \end{aligned} \tag{2}$$

where f_{best} denotes the optimal individual.

Black kites are predators of small mammals and insects in grassland ecosystems. During flight, they adjust the angles of their wings and tail according to wind speed, hover steadily to observe prey, and then rapidly dive to attack. This strategy involves different attack behaviors that facilitate global exploration and search. The mathematical model of the black kite attack behavior is expressed in Eq. (3).

$$y_{t+1}^{i,j} = \begin{cases} y_t^{i,j} + n\left(1+\sin\left(r\right)\right)\times y_t^{i,j} & p < r \\ y_t^{i,j} + n\times\left(2r-1\right)\times y_t^{i,j} & else \end{cases} \tag{3}$$

$$n = 0.05\times e^{-2\times\left(\frac{r}{T}\right)^2} \tag{4}$$

where $y_t^{i,j}$ denotes the position of individual i in the *j*-th dimension at iteration t; $y_{t+1}^{i,j}$ represents the updated position; r is a random number uniformly distributed in [0,1]; t denotes the current iteration number (time step); p is another random number uniformly distributed in [0,1] used for strategy selection; and T denotes the total number of iterations.

Bird migration is a complex behavior influenced by environmental factors such as climate conditions and food availability. To adapt to seasonal changes, many bird species migrate southward during winter to obtain more favorable living conditions and resources. Migration is typically guided by leaders, whose navigation ability is crucial to the success of the group. If the fitness value of the current population is lower than that of a randomly generated population, the leader relinquishes its leadership role and joins the migrating population, indicating that it is no longer suitable to guide the group forward. Conversely, if the fitness value of the current population is higher than that of the random population, the leader continues to guide the population toward the destination [18-19]. This strategy enables dynamic selection of superior leaders, thereby ensuring successful migration.

The mathematical model of the black kite migration behavior is expressed in Eq. (5).

$$y_{t+1}^{i,j} = \begin{cases} y_t^{i,j} + C(0,1) \times \left(y_t^{i,j} - L_t^j \right) & F_i < F_{ri} \\ y_t^{i,j} + C(0,1) \times \left(L_t^j - m \times y_t^{i,j} \right) & else \end{cases} \tag{5}$$

$$m = 2 \times \sin\left(r + \frac{\pi}{2} \right) \tag{6}$$

where L_t^j denotes the position of the leader or attractor in the *j*-th dimension; $y_t^{i,j}$ represents the position of the *i*-th individual in the *j*-th dimension at the *t*-th iteration; F_i denotes the fitness value of the current *i*-th individual; F_{ri} represents the reference fitness value used for comparison; m denotes a dynamic adjustment factor; and $C(0,1)$ represents the Cauchy mutation.

The one-dimensional Cauchy distribution is a continuous probability distribution with two parameters. The probability density function of the one-dimensional Cauchy distribution is given in Eq. (7).

$$f(x, \delta, \mu) = \frac{1}{\pi} \frac{\delta}{\delta^2 + (x - \mu)^2}, -\infty < x < \infty \tag{7}$$

When $\delta = 1$ and $\mu = 0$, the probability density function reduces to the standard form. The probability density function is expressed in Eq. (8).

$$f(x, \delta, \mu) = \frac{1}{\pi} \frac{\delta}{x^2 + 1}, -\infty < x < \infty \tag{8}$$

where x denotes the random variable; δ represents the scale parameter; μ represents the location parameter.

The computational workflow of the Black Kite Algorithm is illustrated in Figure 1.

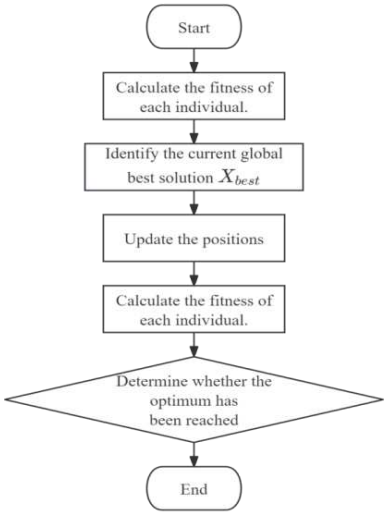


Figure 1. Flowchart of the Black Kite Algorithm

2.2 Problems of the Black Kite Optimization Algorithm

Applying the Black Kite Optimization Algorithm (BKA) to Transformer models can significantly improve model training and structural design. The algorithm can automatically search for and determine the optimal combination of Transformer hyperparameters, such as learning rate, hidden-layer dimension, and the number of attention heads. It therefore replaces time-consuming manual tuning. However, BKA still suffers from inherent drawbacks in convergence efficiency and stability, as well as a mechanism mismatch when dealing with complex optimization tasks [20-21].

Problem 1: Inherent deficiencies in convergence efficiency and stability.

The updating mechanism of BKA relies heavily on random selection and random switching. As a result, the convergence path becomes unstable and poorly repeatable in complex, non-convex search spaces. In addition, its linear exploration–exploitation balancing strategy cannot adaptively respond to different stages of the optimization process. This often leads to premature convergence or ineffective exploration. Both phenomena degrade the reliability and efficiency of the final optimization results [22].

Problem 2: Mechanism mismatch in high-dimensional complex optimization tasks.

BKA was originally designed for continuous optimization problems. Its position-updating rules, mutation operators, and leader-updating strategies are not well suited to the characteristics of Transformer hyperparameter search. These include high dimensionality, mixed discrete–continuous variables, and extremely high evaluation costs [23]. Consequently, the algorithm incurs heavy computational overhead and exhibits poor scalability. Its core operators also lack adaptability to complex structural search tasks, which becomes a key bottleneck in practical applications.

2.3 Improved Black Kite Optimization Algorithm

The standard BKA shows several inherent limitations when solving complex optimization problems. Its convergence path is unstable. It is prone to local optima. Its efficiency also degrades in high-dimensional mixed parameter spaces. To enhance its overall performance in complex scenarios such as hyperparameter optimization, this study introduces a systematic improvement to BKA from three perspectives: convergence strategy, step-size regulation, and encoding mechanism. On this basis, an enhanced BKA is proposed. The overall improvement procedure is illustrated in Figure 2.

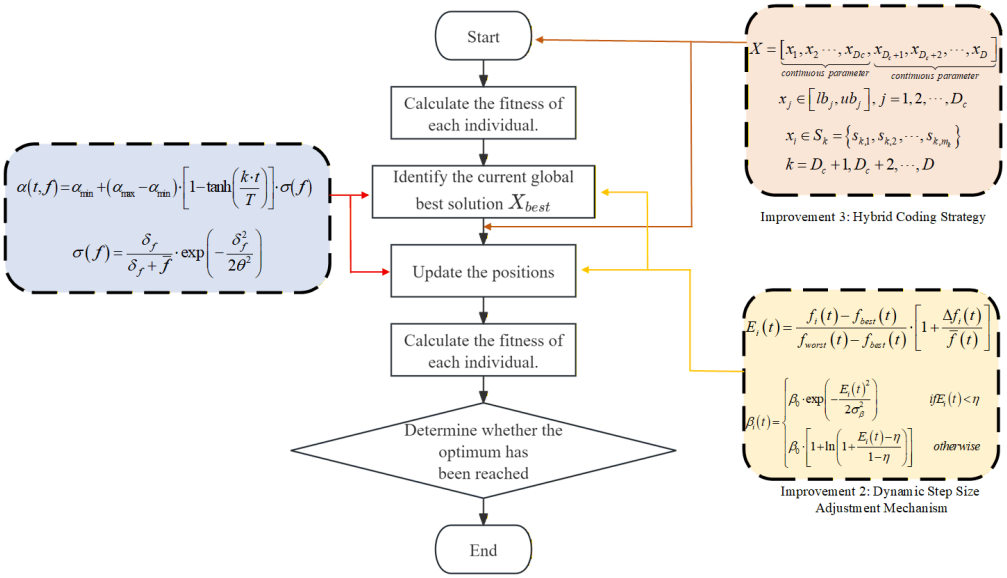


Figure 2. Computational flowchart of the improved BKA

(1) Improvement 1 of BKA: Adaptive nonlinear convergence factor

To mitigate the unstable convergence paths and premature convergence caused by random switching in BKA, an adaptive convergence factor is introduced. It is constructed based on iteration count and fitness diversity. In addition, a dynamic step-size adjustment mechanism is designed. These two components enable the algorithm to perform sufficient global exploration in

the early stage and fine local exploitation in the later stage, thereby improving both convergence efficiency and stability.

To address the limitations of the original linear convergence strategy in BKA, this study designs an adaptive nonlinear convergence factor $\alpha(t,f)$ based on the hyperbolic tangent function. This factor is related not only to the iteration count but also to population diversity. As a result, the algorithm exhibits stronger global exploration ability in the early phase of optimization, while placing greater emphasis on local exploitation in the later phase.

The adaptive nonlinear convergence factor is defined in Eq. (9):

$$\alpha(t,f)=\alpha_{\min}+(\alpha_{\max}-\alpha_{\min})\cdot\left[1-\tanh\left(\frac{k\cdot t}{T}\right)\right]\cdot\sigma(f)\qquad t=1,2,\cdots,T \tag{9}$$

where $\alpha_{\min}$ and $\alpha_{\max}$ denote the lower and upper bounds of the convergence factor, respectively, and are typically set to $\alpha_{\min}=0.1$ and $\alpha_{\max}=1.5$; t denotes the current iteration number; T denotes the maximum number of iterations; k is the shape-adjustment parameter that controls the convergence rate and is commonly set to $k\in[2,5]$; $\sigma(f)$ denotes the fitness-diversity adjustment coefficient of the population.

The fitness-diversity adjustment coefficient of the population is defined in Eq. (10):

$$\sigma(f)=\frac{\delta_f}{\delta_f+\bar{f}}\cdot\exp\left(-\frac{\delta_f^2}{2\theta^2}\right) \tag{10}$$

where $\delta_f=f_{\max}-f_{\min}$ denotes the range of the current population fitness values; $\bar{f}=\frac{1}{N}\sum_{i=1}^{N}f_i$ denotes the mean fitness of the population; θ is the adjustment parameter that controls the sensitivity to population diversity.

When the population diversity is high (δ_f is large), $\sigma_f\to 1$ takes a relatively large value so that the algorithm can maintain strong global exploration ability. When the population tends to converge (δ_f is small), $\sigma(f)\to 0$ becomes smaller, which strengthens the local exploitation ability.

(2) Improvement 2 of the BKA: Dynamic step-size adjustment mechanism

To overcome the instability caused by random step sizes in the original algorithm, a dynamic step-size adjustment mechanism based on the evaluation of the individual evolutionary state is proposed. This mechanism consists of three core components.

First, the evolutionary state evaluation function of each individual is defined, as shown in Eq. (11):

$$E_i(t)=\frac{f_i(t)-f_{best}(t)}{f_{worst}(t)-f_{best}(t)}\cdot\left[1+\frac{\Delta f_i(t)}{\bar{f}(t)}\right] \tag{11}$$

where $E_i(t)$ is the evolutionary-state evaluation value of the *i*-th individual; $f_i(t)$ refers to the fitness of the *i*-th individual at iteration *t*; $f_{best}(t)$ and $f_{worst}(t)$ represent the current best and worst fitness values, respectively; $\Delta f_i(t)=\left|f_i(t)-f_i(t-1)\right|$ indicates the fitness variation of the individual; $\bar{f}(t)$ is the current mean fitness of the population.

Then, a dynamic step-size adjustment factor is introduced. This factor is constructed based on the evolutionary-state evaluation value of each individual and is defined in Eq. (12).

$$\beta_i(t)=\begin{cases}\beta_0\cdot\exp\left(-\dfrac{E_i(t)^2}{2\sigma_\beta^2}\right) & if\ E_i(t)<\eta\\ \beta_0\cdot\left[1+\ln\left(1+\dfrac{E_i(t)-\eta}{1-\eta}\right)\right] & otherwise\end{cases} \tag{12}$$

where β_0 acts as the baseline step-size factor; σ_β serves as the scale parameter of the Gaussian distribution and regulates the exploitation intensity of elite individuals; η functions as a threshold parameter that separates elite individuals from non-elite ones.

Finally, an improved attack-behavior updating rule is developed in this study. The adaptive nonlinear convergence factor and the dynamic step-size adjustment mechanism are incorporated into the original attack-behavior model. In this way, an enhanced position-updating formulation is obtained. The improved updating rules are expressed in Eqs. (13)–(14).

When $r_2<0.5$, the algorithm operates in the global-exploration mode:

$$x_{i,d}^{new}(t+1)=x_{i,d}(t)+\alpha(t,f)\cdot\beta_i(t)\cdot r_1\cdot\left[x_{r1,d}(t)-x_{r2,d}(t)\right]\cdot\left(1+\gamma\cdot C\right) \tag{13}$$

When $r_2\geq 0.5$, the algorithm operates in the local-exploitation mode:

$$x_{i,d}^{new}(t+1)=x_{i,d}(t)+\alpha(t,f)\cdot\beta_i(t)\cdot r_1\cdot\left[L_d(t)-x_{i,d}(t)\right]\cdot\left(1-0.5\cdot\gamma\cdot C\right) \tag{14}$$

where $C\sim Cauchy(0,1)$ denotes a random variable that follows the standard Cauchy distribution; γ represents the disturbance-intensity coefficient.

(3) Improvement 3 of the BKA: Hybrid encoding strategy

To address the challenges of hyperparameter optimization, such as high dimensionality, mixed discrete–continuous variables, and high evaluation cost, a hybrid encoding strategy is designed in this study. Continuous and discrete parameters are encoded separately. In addition, a dimension-aware Cauchy–Gaussian hybrid mutation mechanism is introduced. These designs enhance the search capability and adaptability of the algorithm in high-dimensional complex spaces.

Let the total number of parameters to be optimized be D. The dimensionality of continuous parameters is D_c, and that of discrete parameters is D_d, satisfying $D=D_c+D_d$. A piecewise hybrid encoding strategy is adopted. The solution vector is represented in Eq. (15).

$$X=\underbrace{\left[x_1,x_2\cdots,x_{Dc}\right.}_{continuous\ parameter},\underbrace{\left.x_{D_c+1},x_{D_c+2},\cdots,x_D\right]}_{continuous\ parameter} \tag{15}$$

For the continuous-parameter part, standard floating-point encoding is applied.The boundary of each dimension is expressed in Eq. (16).

$$x_j\in\left[lb_j,ub_j\right],j=1,2,\cdots,D_c \tag{16}$$

For the discrete-parameter part, integer encoding is adopted. The value of each dimension is selected from a predefined discrete set. This discrete set is expressed in Eq. (17).

$$\begin{aligned}&x_i\in S_k=\left\{s_{k,1},s_{k,2},\cdots,s_{k,m_k}\right\}\\&k=D_c+1,D_c+2,\cdots,D\end{aligned} \tag{17}$$

where X denotes the complete solution vector; x_j refers to the value of the *j*-th continuous parameter; lb_j and ub_j are the lower and upper bounds of the *j*-th continuous parameter, respectively; x_k indicates the value of the *k*-th discrete parameter; S_k represents the candidate

set of the *k*-th discrete parameter; $s_{k,i}$ corresponds to the *i*-th candidate value in the candidate set of the *k*-th discrete parameter; m_k denotes the number of candidate values for the *k*-th discrete parameter.

To improve the search efficiency of the algorithm in high-dimensional spaces, a dimension-aware Cauchy–Gaussian hybrid mutation mechanism is proposed. The mechanism automatically adjusts the mutation intensity according to the parameter dimensionality. In addition, probabilistic perturbations are introduced when updating discrete parameters to enhance global exploration ability. For the continuous-parameter part, the mutation rule is given in Eq. (18).

$$x_j' = x_j + \omega_j \cdot \left(Cauchy(0,1) + \lambda \cdot N\left(0, \sigma_j^2\right) \right) \tag{18}$$

where the dimension-aware weight ω_j is defined as shown in Eq. (19).

$$\omega_j = \frac{1}{1 + \alpha \cdot (j - 1)} \tag{19}$$

The standard deviation varies with the dimensionality σ_j, and is expressed in Eq. (20).

$$\sigma_j = \beta \cdot \exp\left(-\frac{j-1}{D_c} \right) \tag{20}$$

For the discrete-parameter part, a probabilistic perturbation strategy is introduced, which is formulated in Eq. (21).

$$P_{mutation}\left(x_k = s_{k,i}\right) = \frac{\exp\left(-\gamma \cdot |\, i - I_{current}\,|\right)}{\sum_{q=1}^{m_k} \exp\left(-\gamma \cdot |\, q - I_{current}\,|\right)} \tag{21}$$

where x_j' is the value of the *j*-th continuous parameter after mutation; ω_j denotes the mutation weight of the *j*-th dimension; α is the weight-decay coefficient; σ_j refers to the Gaussian-mutation standard deviation of the *j*-th dimension; β acts as the initial standard-deviation coefficient; λ represents the mixing ratio between the Cauchy and Gaussian mutations; γ is the probability-decay coefficient that controls how fast the probability decreases with the distance between a candidate value and the current value; and $I_{current}$ indicates the index of the current discrete-parameter value in the candidate set.

2.4 Transformer model

The Transformer model is a powerful sequence feature extractor. It completely removes recurrence and convolution. Instead, it relies solely on the self-attention mechanism and feed-forward neural networks. The model processes the entire input sequence in parallel and efficiently captures long-range dependencies. In essence, it adopts an encoder-only architecture. The decoder is omitted, because the task concerns sequence classification rather than sequence generation. Through multi-head self-attention, the model directly mines fault-related local impulses, periodic modulation patterns, and long-term dependencies in vibration signals. Global pooling and a classifier are then applied to produce the final diagnostic results. This makes the Transformer particularly suitable for fault identification in rolling bearings and other rotating machinery. The operating principle of the Transformer model is illustrated as follows.

Step 1: The raw signal is segmented, and feature extraction is performed on each segment. The procedure is expressed in Eq. (22).

$$X = \left[FFT\left(F_{time}\left(S_{segment}\right)\right); F_{stat}\left(S_{segment}\right) \right] \tag{22}$$

where S is the original vibration-acceleration signal sequence; $S_{segment}$ refers to the signal segments obtained through sliding-window partitioning; $F_{time}(\cdot)$ denotes the time-domain

feature-extraction function; $F_{stat}(\cdot)$ represents the statistical feature-extraction function; $FFT(\cdot)$ stands for the fast Fourier transform; $\Phi(\cdot)$ serves as an automated feature extractor within feature engineering; X indicates the input feature-sequence matrix; T is the number of time steps; and $d_{feature}$ is the feature dimensionality.

Step 2: Positional encoding is applied. The learnable positional encoding of the sequence is expressed in Eq. (23).

$$X_{in} = X + P_{learnable} \tag{23}$$

where X_{in} refers to the initial input of the encoder; $P_{learnable} \in R^{T \times d_{feature}}$ corresponds to the learnable positional-encoding matrix.

The multi-head attention mechanism of the Transformer model is illustrated in Figure 3.

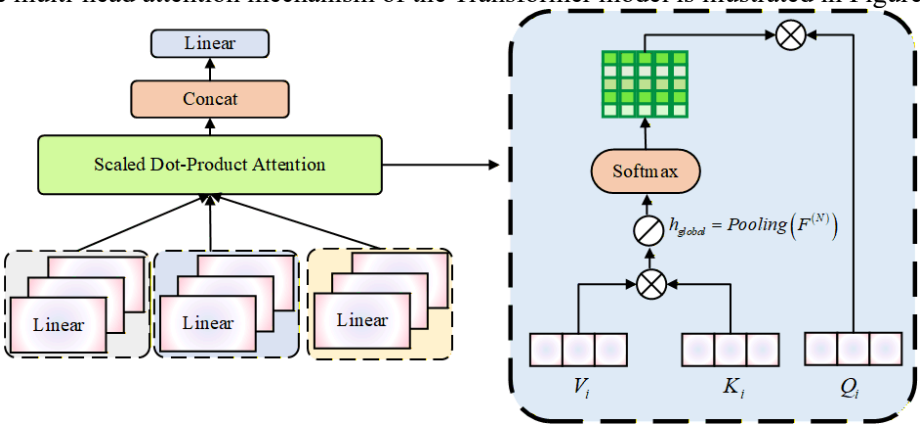


Figure 3. Multi-head attention mechanism of the Transformer model

The specific computational formulation of the multi-head attention mechanism in the Transformer model, as shown in Figure 3, is expressed in Eq. (24).

$$\begin{aligned} MultiHead(X_{in}) &= Concat(head_1, \cdots, head_h) W^O \\ head_i &= Attention(X_{in} W_i^Q, X_{in} W_i^K, X_{in} W_i^V) \end{aligned} \tag{24}$$

where h represents the number of attention heads; W_i^Q and $W_i^K \in R^{d_{feature} \times d_k}$ serve as the projection matrices of the *i*-th head for the "query" and "key" of the fault features, respectively; $W_i^V \in R^{d_{feature} \times d_v}$ acts as the projection matrix of the *i*-th head for the "value" of the fault features; d_k and d_v specify the projection dimensionality of each attention head.

Step 3: Residual connection and layer normalization are applied, as expressed in Eq. (25).

$$\begin{aligned} Z' &= LayerNorm(X_{in} + Dropout(MultiHead(X_{in}))) \\ F &= LayerNorm(Z' + Dropout(FFN(Z'))) \end{aligned} \tag{25}$$

where Z' denotes the intermediate feature obtained after aggregating contextual information through the attention mechanism; F corresponds to the output feature of a single encoder layer; $FFN(\cdot)$ represents the feed-forward neural network;

Step 4: A diagnostic vector is generated, completing the global feature aggregation process. The formulation is expressed in Eq. (26).

$$h_{global} = Pooling(F^{(N)}) \tag{26}$$

where $F^{(N)}$ refers to the output sequence of the *N*-th encoder layer with dimensionality

$T \times d_{feature}$; $Pooling(\cdot)$ denotes the global max-pooling operation; $F_0^{(N)}$ indicates that the feature vector is used as the global representation; and $h_{global\in}R^{d_{feature}}$ represents the global diagnostic feature vector that aggregates information from the entire sequence.

Step 5: Fault classification is performed, as expressed in Eq. (27).

$$\hat{y} = soft\max\left(W_{cls}h_{global} + b_{cls}\right) \tag{27}$$

where $W_{cls} \in R^{C\times d_{feature}}$ denotes the weight matrix of the classification layer; $b_{cls} \in R^{C}$ refers to the bias vector of the classification layer; C specifies the total number of fault categories; and $\hat{y} \in R^{C}$ represents the probability distribution of the model predictions over all categories.

The supervised training loss is expressed in Eq. (28).

$$L = -\frac{1}{M}\sum_{i=1}^{M}\sum_{c=1}^{C} y_{i,c}\log\left(\hat{y}_{i,c}\right) + \lambda\left\|\Theta\right\|^{2} \tag{28}$$

where M denotes the batch size; $y_{i,c} \in \{0,1\}$ is the true fault-class label of sample i; $\hat{y}_{i,c}$ gives the predicted probability that sample i belongs to class c; λ serves as the L2 regularization coefficient; and Θ represents the set of all model weight parameters.

2.5 Computational workflow of the industrial robot bearing fault diagnosis method based on an improved BKA-optimized Transformer

An end-to-end fault diagnosis framework for industrial robot bearings is developed in this study. The framework integrates intelligent hyperparameter optimization with deep sequence modeling by combining the improved BKA with the Transformer model. First, multidimensional feature engineering is applied to the raw vibration signals to construct the initial feature sequences. Then, the improved IBKA is employed to perform automated global optimization of the key Transformer hyperparameters. At the same time, feature-importance screening is conducted in the high-dimensional feature space to improve model efficiency and generalization. Finally, the optimized features are fed into the Transformer network, whose structure and parameters have both been optimized, to model contextual dependencies and accurately identify fault states. The computational workflow of the proposed algorithm is illustrated as follows.

Step 1: Vibration signal acquisition and preprocessing

Raw one-dimensional vibration acceleration signals are collected from the rolling bearings in the joint drive unit of an industrial robot under different health conditions. These conditions include normal operation, inner-race faults, outer-race faults, rolling-element faults, and multiple fault-severity levels. The continuous signals are sampled at a sufficiently high frequency to satisfy the Nyquist sampling theorem, yielding discrete-time sequences. Based on the characteristic fault frequencies and rotational speed of the bearing, each sequence is divided into samples of fixed length to ensure that every sample contains a complete fault-impact period. Each sample is then standardized to zero mean and unit variance. This removes scale differences and accelerates the convergence of the subsequent deep model training process.

Step 2: Construction of multidimensional feature sequences and collaborative optimization with IBKA

For each preprocessed sample, multidimensional feature extraction is performed to construct the initial input sequence. This process integrates time-domain statistical features (such as peak value, kurtosis, and root-mean-square value), frequency-domain features (including spectral amplitude and spectral centroid obtained via fast Fourier transform), and deep primitive features automatically learned by a one-dimensional convolutional network. These components jointly form a high-dimensional hybrid feature vector, which is then reconstructed into a sequential format. Subsequently, the IBKA is employed to iteratively optimize the Transformer hyperparameters. At the same time, the constructed high-dimensional hybrid features are evaluated and filtered according to their importance. Through iterative search with IBKA, two optimal outputs are finally obtained: one is the best hyperparameter combination for the Transformer model; the other is a selected feature subset that is most sensitive to faults. In this way, automated and coordinated optimization of feature dimensionality reduction and model

parameter configuration is achieved.

Step 3: Model training and fault classification

The optimized feature vectors, obtained after feature selection and sequence reconstruction, are used as the inputs to construct a Transformer encoder network. The model employs learnable positional encoding to adapt to non-uniformly sampled signals. Through the multi-head self-attention mechanism, it parallelly captures long-range dependency relationships among fault features within the sequence. Nonlinear transformation and feature fusion are then performed using a feed-forward neural network together with residual connections. The network structure is initialized with the hyperparameter configuration optimized by IBKA. The loss function is minimized via backpropagation and adaptive optimization algorithms, enabling the model to learn an accurate mapping from complex vibration sequences to fault categories.

After training, the model processes the input feature sequence in an end-to-end manner. A global pooling layer is used to aggregate the sequential information, and a Softmax classifier outputs the probability distribution over the bearing fault states. In this way, high-accuracy intelligent fault diagnosis is achieved. The computational workflow of the proposed industrial robot bearing fault diagnosis method, in which the Transformer is optimized by the improved BKA, is shown in Figure 4.

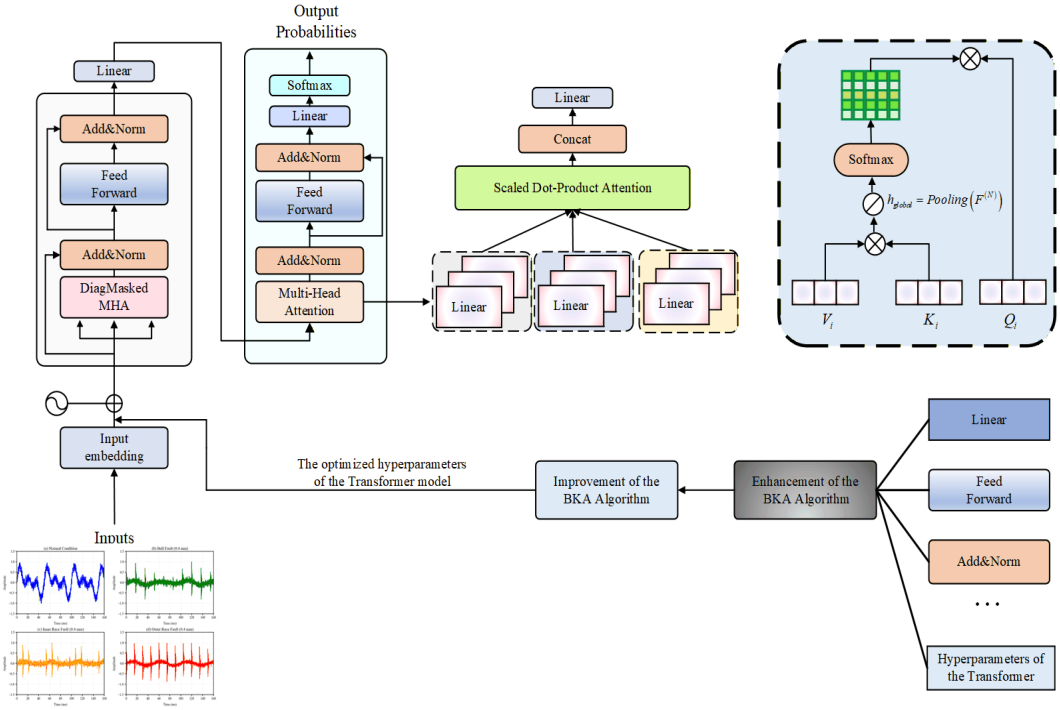


Figure 4. Improved BKA-Transformer–based bearing fault diagnosis workflow

3. Results and discussion

3.1 Effectiveness analysis of the improvements in the BKA optimization

To systematically evaluate the performance of the improved BKA proposed in this study and to verify the effectiveness of the improvement strategies, eight representative benchmark test functions with different characteristics (F1–F8) are selected for performance validation [24–25].

All comparative experiments are conducted on the same computing platform. The operating system is Windows 11. The hardware configuration consists of an AMD Ryzen 7-5800H processor and 32 GB of memory.

To ensure fairness and reproducibility, the population size of all algorithms is uniformly set to 60, and the maximum number of iterations is fixed at 100. Each algorithm is independently executed 20 times. For every run, the best solution, the worst solution, and the average running time are recorded. These indicators are then used to comprehensively evaluate the convergence accuracy, robustness, and computational efficiency of the algorithms.

The specific benchmark test functions are listed in Table 1.

Table 1. Benchmark test functions

Function ID	Function name	Domain
F1	Schwefel Function	[-500,500]
F2	Rastrigin Function	[-5.12,5.12]
F3	Ackley Function	[-32.768,32.768]
F4	Griewank Function	[-600,600]
F5	Levy Function	[-10,10]
F6	Levy Function 13	[-10,10]
F7	Shekel's Foxholes Function	[-65.536,65.536]
F8	Kowalik Function	[-5,5]

To verify the effectiveness of the optimization improvements made to the BKA in this study, comparative experiments are conducted using the BKA, the IBKA, and the AOA algorithm. The eight benchmark functions described above are used as the test set. Each algorithm is executed 20 independent runs. The convergence curves of functions F1-F8 are visualized for comparison. The simulation results are shown in Figure 5.

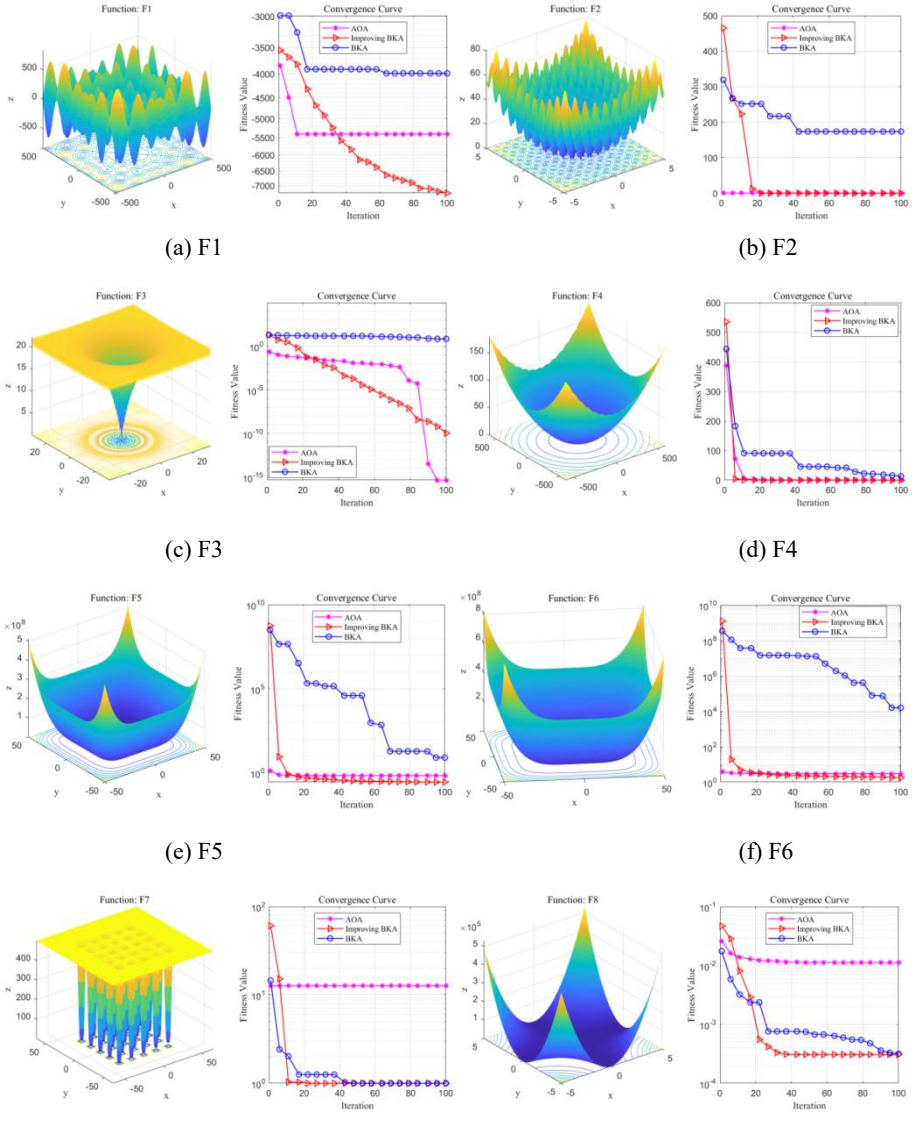


Figure 5. Simulation comparison results for benchmark functions F1–F8

In Figure 5, the left column shows the benchmark test functions, while the right column

presents the convergence curves of the three algorithms: AOA, the improved BKA, and the standard BKA. From the convergence curves, it can be observed that the improved BKA exhibits superior optimization performance. For all benchmark functions, its fitness value decreases more rapidly, and the final optimization result is also better. The optimization results of the proposed method are summarized in Table 2.

Table 2. Optimization results of the proposed method

Test function	Best solution	Worst solution	Average time / s
F1	-7314.3124	-3500.2417	0.72
F2	−10.4029	320	0.31
F3	0	1.0921	0.23
F4	0	532.1286	0.14
F5	0	77.2373	0.17
F6	12.3124	11×109	0.78
F7	0.3979	5.8214	0.13
F8	0	81.7426	0.19

By analyzing the optimization results of the proposed method presented in Table 2, it can be observed that the improved BKA exhibits noticeable performance variation across different types of benchmark functions. The evaluation is conducted from three perspectives: convergence accuracy, robustness, and computational efficiency. .

First, in terms of convergence accuracy, the improved BKA is able to obtain the theoretical optimal solution or a near-optimal solution for several benchmark functions. For example, the algorithm achieves the theoretical optimum of 0 on functions F3, F4, F5, and F8. This indicates strong local convergence capability for relatively regular or low-dimensional problems. In addition, on F7 (Branin), the best solution of 0.3979 is also very close to the known global optimum, suggesting that the algorithm is effective for such classical test functions. However, for F1 and F2, the best solutions are -7314.3124 and -10.4029, respectively. Although these results demonstrate a certain level of optimization capability, the absence of known theoretical optima makes it difficult to directly assess their convergence accuracy. It is also worth noting that for F6, the best solution is 12.3124, while the worst solution reaches 1.1×10^9. This suggests that the function may have a highly complex search landscape, and that the algorithm may experience severe degradation in convergence performance in some runs.

Second, in terms of robustness, the gap between the best and worst solutions reflects the stability of the algorithm across multiple runs. For F3, F5, and F7, the differences between the best and worst results are relatively small. This indicates that the algorithm maintains good repeatability and stable convergence on these functions. However, for F2, F4, F6, and F8, the gaps are extremely large. For example, the worst solution for F2 is 320, which differs greatly in both sign and magnitude from the best solution of -10.4029. For F6, the worst solution reaches 11×10^9. These results reveal that the algorithm behaves very inconsistently on these functions and is easily influenced by initial conditions or random factors, leading to poor local solutions or divergence. Such large fluctuations imply that the balance between exploration and exploitation may still be insufficient, particularly for multimodal, high-dimensional, or non-convex functions.

Finally, regarding computational efficiency, the improved BKA shows consistently low average running time across all benchmark functions, ranging from 0.13 s to 0.78 s. This indicates good time performance. Even for F6, where the results fluctuate significantly, the average running time is only 0.78 s. This demonstrates that the algorithm executes rapidly and has relatively low computational complexity, making it suitable for applications with certain real-time requirements. For more complex functions such as F1, the average time of 0.72 s is still acceptable.

In summary, the proposed method performs well in terms of computational efficiency and achieves the theoretical global optimum for several benchmark functions, demonstrating strong local optimization capability. However, the lack of robustness and the large variations in optimization results observed for multiple functions indicate that its global convergence stability remains insufficient when dealing with complex, multimodal, or high-dimensional optimization problems. Future work may therefore focus on further enhancing the global exploration mechanism to avoid premature convergence and large performance fluctuations.

3.2 Data source

The experimental data used in this study are obtained from a self-constructed vibration dataset of industrial robot joint-bearing faults. The dataset is collected on a high-precision bearing fault test rig, which can accurately reproduce the dynamic characteristics of robot bearings under various operating modes and load conditions. The integrated experimental platform employed for data acquisition is shown in Figure 6.

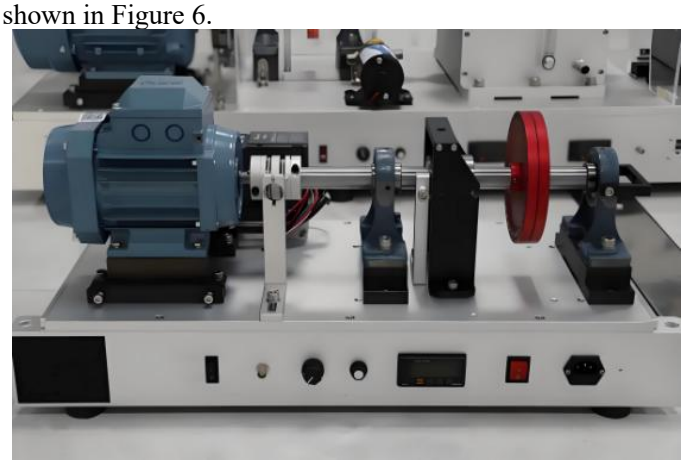


Figure 6. Bearing fault test rig for industrial robots

The constructed dataset is designed to support precise fault-classification research. Therefore, both the fault categories and the dataset scale are deliberately configured. A total of 50,000 high-quality vibration time-series samples are included. Each sample has a fixed length of 2,048 data points. This length is chosen to effectively capture characteristic fault frequencies and impact responses, while maintaining computational efficiency during model training.

The dataset covers four core operating states: normal condition, rolling-element fault, inner-race fault, and outer-race fault [26]. Multiple damage-severity levels are assigned to each fault type in order to simulate the progression of faults from initiation to development. To better approximate real industrial environments, data collection is performed under combined variable-speed and variable-load conditions. This significantly increases the complexity and engineering relevance of the dataset. The dataset therefore provides a reliable basis for evaluating the accuracy, generalization ability, and noise robustness of the proposed diagnosis model in identifying key bearing fault types in industrial robots. The detailed configuration parameters of the dataset are summarized in Table 3.

Table 3. Detailed parameters of the industrial robot joint-bearing vibration dataset

Category	Fault diameter	Rotational speed (rpm)	Load (N·m)	Number of samples	Data length
Normal condition	N/A	600, 900, 1200, 1500	0, 5, 10	12,500	2048
Rolling-element fault	0.2 mm	600, 900, 1200, 1500	0, 5, 10	4,167	2048
	0.4 mm	600, 900, 1200, 1500	0, 5, 10	4,167	2048
	0.6 mm	600, 900, 1200, 1500	0, 5, 10	4,166	2048
Inner-race fault	0.2 mm	600, 900, 1200, 1500	0, 5, 10	4,167	2048
	0.4 mm	600, 900, 1200, 1500	0, 5, 10	4,167	2048
	0.6 mm	600, 900, 1200, 1500	0, 5, 10	4,166	2048

	0.2 mm	600, 900, 1200, 1500	0, 5, 10	4,167	2048
Outer-race fault	0.4 mm	600, 900, 1200, 1500	0, 5, 10	4,167	2048
	0.6 mm	600, 900, 1200, 1500	0, 5, 10	4,166	2048

3.3 Time–frequency analysis

Time–frequency analysis is a key approach for revealing the time-varying characteristics of nonstationary vibration signals in mechanical fault diagnosis. To visually verify that the constructed dataset contains distinguishable fault-related features, and to investigate how different fault states manifest in the joint time–frequency domain, representative samples from the dataset are analyzed in this section. The objective is to confirm, from a physical-mechanism perspective, the presence and differences of fault characteristics such as periodic impacts and resonance-frequency modulation. This provides a theoretical basis for subsequent feature learning using the Transformer model.

In this study, the short-time Fourier transform (STFT) is employed as the time–frequency analysis tool. To comprehensively illustrate the fault characteristics, four representative samples are selected from the dataset described in Table 3 under the same operating condition of 1200 rpm and 5 N·m. These samples correspond to normal condition, rolling-element fault (0.4 mm), inner-race fault (0.4 mm), and outer-race fault (0.4 mm), respectively. Each sample contains 2,048 data points, and the sampling frequency is 12.8 kHz. The experimental results are shown in Figure 7.

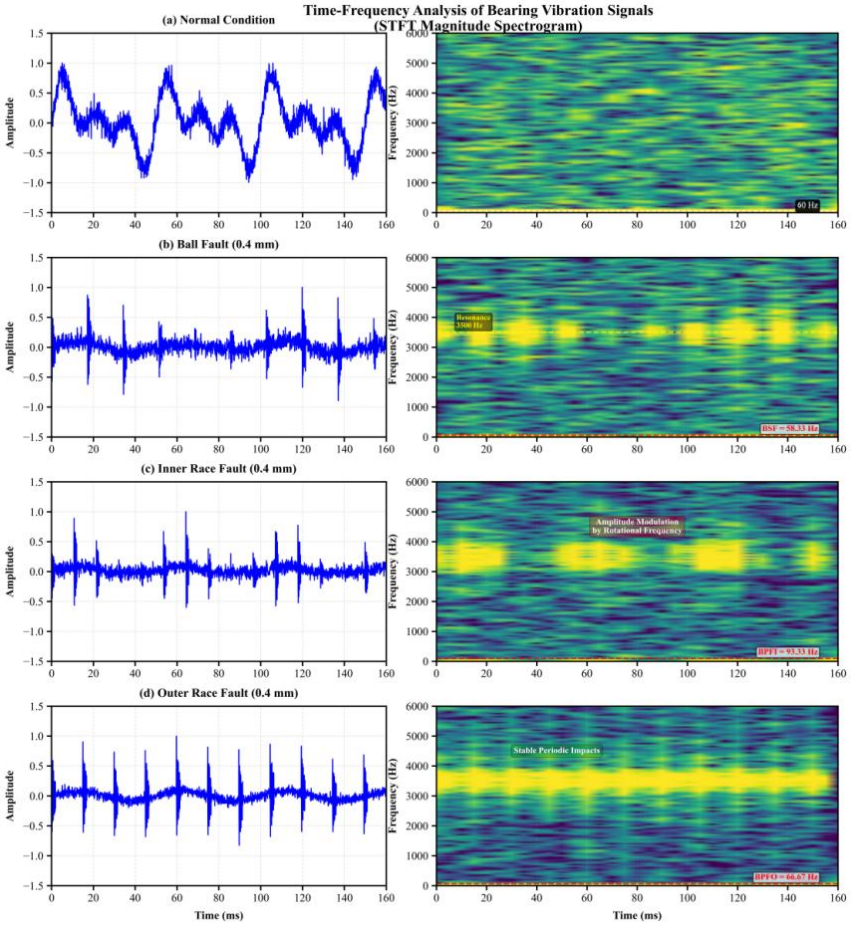


Figure 7. Time–frequency analysis results of industrial robot bearing fault vibration signals

From Figure 7, it can be observed that for the bearing vibration signal under normal operating conditions (Figure 7(a)), the time-domain waveform exhibits small-amplitude and stationary

random fluctuations, without any obvious impulsive disturbances. In the corresponding short-time Fourier transform (STFT) time–frequency map, the frequency components are distributed relatively uniformly. The energy does not show clear time-varying aggregation or periodic high-energy patches across different frequency bands. This indicates that the time-varying characteristics of the vibration signal are weak in the normal state, and no modulation or impact effects associated with faults are present. These observations are consistent with the characteristics of a healthy bearing.

For the rolling-element fault with a defect size of 0.4 mm (Figure 7(b)), the time-domain waveform shows pronounced impulsive fluctuations. The impulse intervals are consistent with the characteristic fault frequency of the rolling element. In the STFT time–frequency map, periodically distributed high-energy patches appear in the mid-to-high-frequency region. The frequency range of these energy clusters corresponds to the resonance band of the bearing system. This phenomenon originates from the periodic impact excitation generated when the defective rolling element contacts the inner and outer races. Each time the defect passes through the load zone, structural resonance is excited, resulting in periodic energy aggregation in the time–frequency domain. This reflects the typical "impact–resonance" vibration mechanism associated with rolling-element faults.

For the inner-race fault with a defect size of 0.4 mm (Figure 7(c)), the impulses in the time-domain waveform occur more frequently, and the impulse amplitude shows a modulation pattern related to the shaft-rotation frequency. In the corresponding STFT map, the magnitude of the high-energy regions varies periodically with the rotational period. The underlying physical mechanism is that the inner race rotates synchronously with the shaft, so the defect-pass frequency is determined by the shaft-rotation frequency. As a result, the fault-induced impact signal is modulated by the rotational motion, leading to periodic fluctuations in the amplitude of the high-energy patches in the time–frequency domain. This is a typical time-varying characteristic that distinguishes inner-race faults from other fault types.

For the outer-race fault with a defect size of 0.4 mm (Figure 7(d)), the impulses in the time-domain waveform exhibit a stable periodic pattern, and the variation in impulse amplitude is relatively small. This is because the outer race is generally stationary, and the spatial position of the defect remains unchanged. Each time a rolling element passes over the defect, an impact is generated. The impact period is therefore determined by the orbital frequency of the rolling elements. As a consequence, the high-energy regions in the time–frequency map form continuous and stable band-like distributions, reflecting the fixed impact location and stable periodicity characteristic of outer-race faults.

3.4 Ablation study comparison

To systematically evaluate the specific contributions of the three improvement strategies proposed in this study—namely, the adaptive nonlinear convergence factor, the dynamic step-size adjustment mechanism, and the hybrid encoding strategy—to the performance enhancement of the Black Kite Algorithm (BKA), an ablation study is conducted in this section. By gradually removing or replacing each improvement module, the performance differences among several algorithm variants are compared on both benchmark-function optimization tasks and Transformer hyperparameter optimization tasks. In this way, the effectiveness and necessity of each improvement component are verified.

Five algorithm variants are designed for comparison:

Original BKA: the standard Black Kite Algorithm, used as the performance baseline.

BKA-V1: the original BKA with only the adaptive nonlinear convergence factor introduced (corresponding to the improvement in Section 2.3.1).

BKA-V2: the original BKA with only the dynamic step-size adjustment mechanism introduced (corresponding to the improvement in Section 2.3.2, including evolutionary-state evaluation and the improved attack-behavior update).

BKA-V3: the original BKA with only the hybrid encoding strategy introduced (corresponding to the improvement in Section 2.3.3, targeting mixed continuous–discrete parameter spaces).

IBKA (Proposed): the fully improved BKA integrating all three enhancement strategies.

The results of the ablation study are shown in Figure 8.

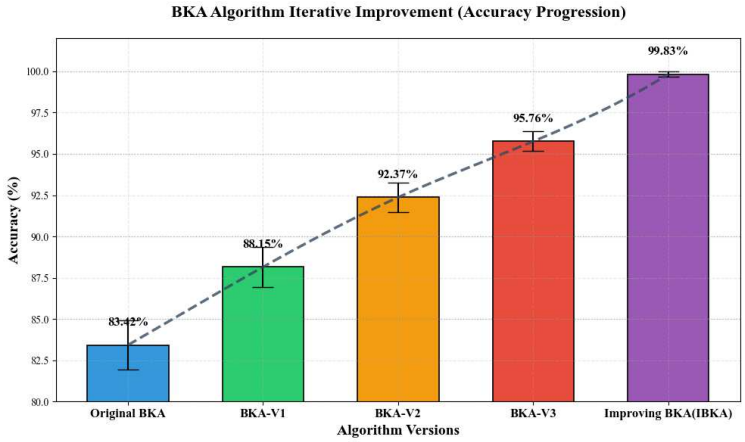


Figure 8. Ablation study comparison results

From Figure 8, it can be seen that the figure presents a bar chart illustrating the progressive improvement in accuracy achieved through the iterative enhancement of the BKA algorithm. The horizontal axis corresponds to the ablation-study variants, while the vertical axis represents the classification accuracy. The numerical labels above the bars indicate the specific accuracy achieved by each variant. The error bars at the top reflect the statistical fluctuation of the accuracy results. The dashed trend line intuitively shows the change in accuracy throughout the iterative improvement process. Using the Original BKA (accuracy: 83.42%) as the baseline, the accuracy demonstrates a consistent upward trend across the successive variants. The accuracy of BKA-V1 increases to 88.15% (an improvement of +4.73 percentage points over the baseline). BKA-V2 further improves the accuracy to 92.37% (an increase of +4.22 percentage points over BKA-V1). BKA-V3 reaches 95.76% (an additional +3.39 percentage points relative to BKA-V2). Finally, the fully improved version, IBKA, achieves an accuracy as high as 99.83%, representing an increase of +4.07 percentage points over BKA-V3.

From the overall trend, the dashed line exhibits a strictly monotonic increase, which verifies the progressive effectiveness of the iterative improvement strategy applied to the BKA. Moreover, the error bars do not enlarge as the accuracy improves (and are even more compact for IBKA), indicating that algorithmic robustness and result stability are enhanced simultaneously during the improvement process. At the same time, although the accuracy gains between successive variants fluctuate slightly, the overall optimization rate remains high. In particular, the renewed increase observed for IBKA suggests that the later improvements successfully overcame the performance bottleneck that limited earlier variants. In summary, the figure provides systematic validation of the multi-generation iterative improvement framework applied to the BKA. The accuracy increases from 83.42% to 99.83%, corresponding to a cumulative improvement of approximately 16.41 percentage points.

3.5 Experimental comparison results of different fault diagnosis methods for industrial robot bearings

To comprehensively evaluate the overall performance of the proposed method, several mainstream fault diagnosis approaches, including ACGAN, SVM, and CNN, are selected for comparison. All methods are tested on the self-constructed industrial robot joint-bearing fault vibration dataset. The experimental results are presented in Figure 9.

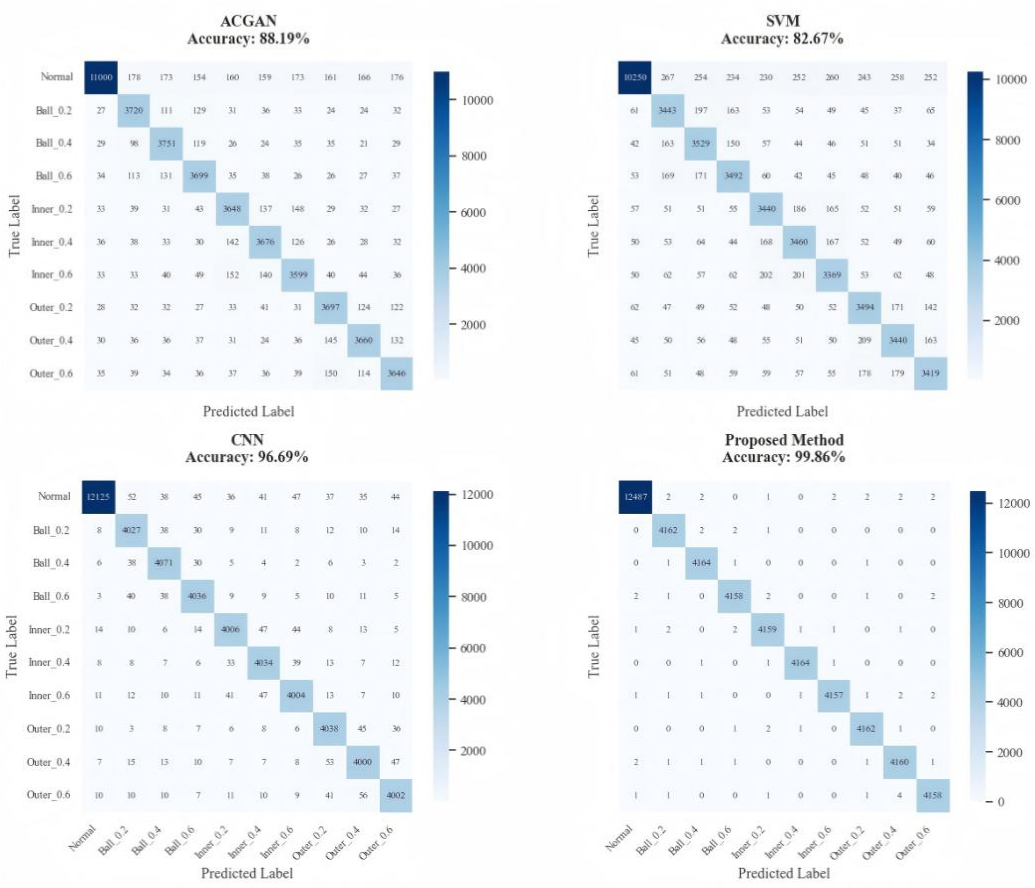


Figure 9. Comparison of confusion matrices for industrial robot bearing fault diagnosis using different algorithms

From Figure 9, it can be seen that the confusion matrices of ACGAN, SVM, CNN, and the proposed method for the target classification task are presented. The rows and columns of each matrix correspond to the true and predicted class labels, respectively. The value of each element represents the number of samples in the corresponding class pair. The color intensity is positively correlated with the sample count. The classification accuracy displayed in the upper-right corner serves as the main evaluation indicator. Based on these matrices, the performance of the different methods can be analyzed as follows. In terms of accuracy, SVM performs the worst, with an accuracy of 82.67%. ACGAN achieves 88.19%, outperforming SVM but still remaining at a relatively low level. CNN shows strong classification capability, with an accuracy of 96.69%. The proposed method achieves the highest accuracy of 99.86%. From the distribution of the elements in the confusion matrices, the non-diagonal elements of SVM and ACGAN are relatively large. This indicates obvious misclassification phenomena and weak discrimination ability among classes. For CNN, the proportion of diagonal elements increases significantly, while most non-diagonal elements are in the single- or double-digit range. Only a small number of samples are misclassified, indicating much stronger class separability. For the proposed method, the diagonal elements dominate almost completely. Most non-diagonal elements are either zero or extremely small, with only a very small number of misclassified samples. This means that the method almost achieves "confusion-free" classification, and the class-discrimination capability is close to perfect. In summary, for this classification task, the performance of the proposed method is significantly superior to that of ACGAN, SVM, and CNN. It not only achieves a classification accuracy of 99.86%, but also greatly reduces inter-class confusion, making it highly suitable for high-precision fault diagnosis applications.

4. Conclusion

In this study, a novel artificial intelligence–based diagnostic method for industrial robot bearing faults is proposed and validated. The method is built upon a Transformer model optimized using an IBKA. Based on experimental analysis and comparative evaluation using a self-constructed bearing vibration dataset, two clear and quantitative conclusions are obtained:

(1) The proposed IBKA–Transformer diagnostic model achieves almost perfect classification performance. On the industrial robot bearing vibration dataset covering multiple fault types and operating conditions, the method attains a fault-diagnosis accuracy of 99.86%, which significantly exceeds that of mainstream approaches such as ACGAN (88.19%), SVM (82.67%), and CNN (96.69%). Analysis of the confusion matrices further shows that the number of misclassified samples is reduced to an extremely low level. Most non-diagonal elements are nearly zero, indicating that the method almost achieves "confusion-free" classification. This fully demonstrates its excellent performance and reliability in high-precision fault diagnosis scenarios.

(2) The systematic improvements introduced to the BKA are the key to enhancing diagnostic performance. The adaptive nonlinear convergence factor, dynamic step-size adjustment mechanism, and hybrid encoding strategy—designed to address the limitations of the original algorithm—are validated through standard benchmark testing and ablation studies. These improvements effectively balance exploration and exploitation, enhance convergence stability, and improve optimization efficiency in high-dimensional mixed-parameter spaces. As a result, they ensure the optimal joint optimization of Transformer hyperparameters and feature selection, thereby providing a solid algorithmic foundation for achieving the final diagnostic accuracy of 99.86%.

5. List of abbreviations

Full name	Abbreviations
Improved Black Kite Algorithm	IBKA
Black Kite Algorithm	BKA
Convolutional Neural Networks	CNN
Short-Time Fourier Transform	STFT

6. Declarations

Availability of data and material: All data included in this study are available upon request by contact with the corresponding author.

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

Funding: This study was supported by the Scientific Research Project of Hunan Provincial Department of Education (Grant No. 24B1045).

Authors' contributions: Jiangnan Su and Jun Huang jointly contributed to the conception, design, and analysis of the study. Jiangnan Su drafted the manuscript. Jun Huang provided critical revisions and assisted with data analysis and interpretation. Both authors read and approved the final manuscript. Jun Huang is the corresponding author.

Acknowledgements: The authors gratefully acknowledge the support provided by the Scientific Research Project of Hunan Provincial Department of Education (Grant No. 24B1045).

References

1. Peng, B., Bi, Y., Xue, B., et al. (2022). A survey on fault diagnosis of rolling bearings. Algorithms, 15(10), 347. https://doi.org/10.3390/a15100347.

2. Urrea, C., & Domínguez, C. (2024). Fault Diagnosis in a Four-Arm Delta Robot Based on Wavelet Scattering Networks and Artificial Intelligence Techniques. Technologies, 12(11), 225. https://doi.org/10.3390/technologies12110225.

3. Zhang, X., & Gu, G. (2024). Fault Diagnosis for Rolling Bearings Under Complex Working Conditions Based on Domain-Conditioned Adaptation. Machines, 12(11), 787. https://doi.org/10.3390/machines12110787.

4. Mao, W., Zhang, D., Tian, S., et al. (2020). Robust detection of bearing early fault based on deep transfer learning. Electronics, 9(2), 323. https://doi.org/10.3390/electronics9020323.

5. Yuan, L., Lian, D., Kang, X., et al. (2020). Rolling bearing fault diagnosis based on convolutional neural network and support vector machine. IEEE Access, 8, 137395-137406. https://doi.org/10.1109/ACCESS.2020.3012053.

6. Ouachtouk, I., El Hani, S., & Dahi, K. (2021). Intelligent bearing fault diagnosis method based on HNR envelope and classification using supervised machine learning algorithms. Adv. Electr. Electron. Eng, 19(4), 282-294. https://doi.org/10.15598/AEEE.V19I4.4183.

7. Chi, F., Yang, X., Shao, S., et al. (2022). Bearing fault diagnosis for time-varying system using vibration–speed fusion network based on self-attention and sparse feature extraction. Machines, 10(10), 948. https://doi.org/10.3390/machines10100948.

8. Zhang, K., Wang, J., Shi, H., et al. (2021). A fault diagnosis method based on improved convolutional neural network for bearings under variable working conditions. Measurement, 182, 109749. https://doi.org/10.1016/j.measurement.2021.109749.

9. Khan, A., Hwang, H., & Kim, H. S. (2021). Synthetic data augmentation and deep learning for the fault diagnosis of rotating machines. Mathematics, 9(18), 2336. https://doi.org/10.3390/math9182336.

10. Zhao, J., Li, J., Yuan, Z., et al. (2024). Sample Augmentation Using Enhanced Auxiliary Classifier Generative Adversarial Network by Transformer for Railway Freight Train Wheelset Bearing Fault Diagnosis. Entropy, 26(12), 1113. https://doi.org/10.3390/e26121113.

11. Ahmed, H. O., & Nandi, A. K. (2023). Convolutional-transformer model with long-range temporal dependencies for bearing fault diagnosis using vibration signals. Machines, 11(7), 746. https://doi.org/10.3390/machines11070746.

12. Hou, Y., Wang, J., Chen, Z., et al. (2023). Diagnosisformer: An efficient rolling bearing fault diagnosis method based on improved Transformer. Engineering Applications of Artificial Intelligence, 124, 106507. https://doi.org/10.1016/j.engappai.2023.106507.

13. Li, M., & Wang, Z. (2020). Deep learning for high-dimensional reliability analysis. Mechanical Systems and Signal Processing, 139, 106399. http://doi.org/10.1016/j.ymssp.2019.106399.

14. Layeb, A. (2024). Differential evolution algorithms with novel mutations, adaptive parameters, and Weibull flight operator. Soft Computing, 28(11), 7039-7091. https://doi.org/10.1007/s00500-023-09561-3.

15. Li, J., Wang, Y., Zi, Y., et al. (2021). Causal disentanglement: A generalized bearing fault diagnostic framework in continuous degradation mode. IEEE Transactions on Neural Networks and Learning Systems, 34(9), 6250-6262. https://doi.org/10.1109/TNNLS.2021.3135036.

16. An, Y., Zhang, K., Chai, Y., et al. (2022). Bearing fault diagnosis under variable working conditions base on contrastive domain adaptation method. IEEE Transactions on Instrumentation and Measurement, 71, 1-11. https://doi.org/10.1109/TIM.2022.3200106.

17. Wang, J., Wang, W. C., et al. (2024). Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems. Artificial Intelligence Review, 57(4), 98. https://doi.org/10.1007/s10462-024-10723-4.

18. Mohapatra, S., Kaliyaperumal, D., & Gharehchopogh, F. S. (2025). A revamped black winged kite algorithm with advanced strategies for engineering optimization. Scientific Reports, 15(1), 17681. https://doi.org/10.1038/s41598-025-93370-1.

19. Chen, H., Jia, T., Guo, J., et al. (2025). A multi-strategy enhanced black-winged kite algorithm for UAV path planning. The Journal of Supercomputing, 81(15), 1-32. https://doi.org/10.1007/s11227-025-07905-4.

20. Zhu, X., Zhang, J., Jia, C., et al. (2025). A Hybrid Black-Winged Kite Algorithm with PSO and Differential Mutation for Superior Global Optimization and Engineering Applications. Biomimetics, 10(4), 236. https://doi.org/10.3390/biomimetics10040236.

21. Li, C., Zhang, K., Zheng, B., et al. (2025). Path planning problem solved by an improved black-winged kite optimization algorithm based on multi-strategy fusion. International Journal of Machine Learning and Cybernetics, 1-37. https://doi.org/10.1007/s13042-025-02693-y.

22. Wang, S., Xu, B., Zheng, Y., et al. (2025). Path Optimization Strategy for Unmanned Aerial Vehicles Based on Improved Black Winged Kite Optimization Algorithm. Biomimetics, 10(5), 310. https://doi.org/10.3390/biomimetics10050310.
23. a Ilemobayo, J., Durodola, O., Alade, et al. (2024). Hyperparameter tuning in machine learning: A comprehensive review. Journal of Engineering Research and Reports, 26(6), 388-395. https://doi.org/10.9734/jerr/2024/v26i61188.
24. Wang, W., Li, K., Jalil, H., et al. (2022). An improved estimation of distribution algorithm for multi-objective optimization problems with mixed-variable. Neural Computing and Applications, 34(22), 19703-19721. https://doi.org/10.1007/s00521-022-07695-3.
25. Naser, M. Z., Al‐Bashiti, M. K., et al. (2025). A review of benchmark and test functions for global optimization algorithms and metaheuristics. Wiley Interdisciplinary Reviews: Computational Statistics, 17(2), e70028. https://doi.org/10.1002/wics.70028.
26. Lei, T., Pan, F., Hu, J., et al. (2025). A fault diagnosis method for rolling bearings in open-set domain adaptation with adversarial learning. Scientific Reports, 15(1), 10793. https://doi.org/10.1038/s41598-025-88353-1.