

Research Article

Bearing Fault Classification Using Improved Antlion Optimizer and Extreme Learning Machine

Zhuanzhe Zhao ,^{1,2} Yu Zhang,¹ Qiang Ma,¹ Yujian Rui ,¹ Guowen Ye,¹ Mengxian Wang,¹ Yongming Liu ,^{1,3} Zhen Zhang,^{1,2} Neng Wei,² and Zhijian Tu²

¹School of Mechanical Engineering, Anhui Polytechnic University, Wuhu, Anhui 241000, China

²Wuhu Ceprei Robotics Industry Technology Research Institute Co. Ltd., Wuhu, Anhui 241003, China

³Anhui New R&D Institutions of Human-machine Interaction and Collaboration, Wuhu, Anhui 241000, China

Correspondence should be addressed to Yongming Liu; liuyongming1015@163.com

Received 4 June 2021; Revised 12 April 2022; Accepted 21 April 2022; Published 18 May 2022

Academic Editor: Aniello Riccio

Copyright © 2022 Zhuanzhe Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Bearing is an important part of rotating machinery, and its early fault diagnosis and accurate classification have always been difficult in engineering application. At present, the models based on the fusion of various optimization algorithms and neural networks have become one of the emerging techniques for accurate fault identification. Firstly, an improved antlion optimizer (ALO) algorithm based on estimation of distribution algorithm (EDA) and variable-step Lévy flight strategy, abbreviated as ELALO, is proposed as a new bionic intelligence. During the initialization of population, the individuals with poor fitness are redistributed by the Gaussian probability model. In view of the stagnation of iteration, Lévy flight strategy is introduced and the adaptive change of disturbance step length is controlled. Experimental results on 4 benchmark functions show that the novel ELALO can effectively improve the solution accuracy and convergence speed, compared with the original ALO. Secondly, in order to solve the disadvantage that extreme learning machine (ELM) network is easy to fall into local optimization, this ELALO algorithm is used to initialize the weights and thresholds of its network and to form the new pattern recognition model, ELALO-ELM. Finally, the bearing data of 8 patterns from Western Reserve University are decomposed by local mean decomposition (LMD), and then the symbolic entropy (SE) of the first three product function (PF) components signals is extracted and used as the input eigenvectors. Compared with the standard ELM and ALO-ELM models, the ELALO-ELM model has better generalization and stronger robustness and it can effectively improve the efficiency of network training and the accuracy of early fault pattern classification in bearing fault diagnosis. The new ELALO-ELM model can also be used for other difficult classification problems.

1. Introduction

Bearing is one of the most widespread and important parts in mechanical equipment. The classification of early faults of bearings is a very important issue in practical applications, which has attracted the attention of many scholars and engineers. Due to the complex structure of most mechanical equipment, there is a nonlinear coupling relationship between components. And the weak non-stationary bearing fault signal is often submerged in the environmental noise. All these have caused some difficulties for the accurate

classification of early faults of bearings. At present, there are three main types of fault diagnosis and classification methods for bearing: model-based methods, knowledge-based reasoning methods, and data-driven methods. The model-based method explored the operating laws of bearings by constructing accurate mathematical models and obtained the related information of normal and abnormal operation by studying the internal relationship between dynamic parameters and response signs under fault conditions. In the literature [1], a dynamic model considering the time-varying deflection excitation (TDE) and the time-

varying contact stiffness excitation (TCSE) is introduced to investigate vibration responses of a lubricated rolling bearing with a uniform and a non-uniform surface waviness on its races. To investigate the vibration mechanism of cylindrical roller bearings with localized defects, Cao et al. [2] developed the dynamic models with gyroscopic moment, centrifugal force, and lubrication traction/slip between roller/raceway, roller/cage, and cage/raceway. However, with the increasing complexity of mechanical systems and more coupling factors, the accurate mathematical models are becoming more and more difficult to establish. The method based on knowledge reasoning does not require the accurate mathematical model of the system, but carries out computational reasoning and classification, such as fault tree-based reasoning [3], DS theory [4], and expert system [5]. These methods usually require a large amount of domain expert knowledge, knowledge expression, and reasoning mechanism. Therefore, it is difficult for its large-scale application in case of incomplete or lack of domain knowledge. The data-driven diagnosis method, such as machine learning, does not need to understand the physical structure and the precise mathematical model, nor does it require a large amount of accumulation of expert knowledge. It only uses measurable condition monitoring signals or historical data to analyze and extract characteristic information, so as to realize the fault diagnosis, classification, and performance evaluation of bearings. A condition monitoring index of bearings was developed based on self-organizing map (SOM) and improved hidden Markov model so as to detect incipient bearing faults quickly [6]. In the literature [7], a three-stage semisupervised learning approach using data augmentation and metric learning is proposed for an intelligent bearing fault diagnosis under limited labeled data. Wen et al. [8] proposed a novel method that integrates graph theory and singular value decomposition (SVD) with the purpose of inspection of dynamic rolling element bearings health conditions. The data-driven method has attracted the great attention of researchers, and its application in the intelligent fault diagnosis has become more and more extensive.

In recent years, various pattern recognition models emerge in an endless stream, such as the support vector machine (SVM) [9], variable prediction model [10], neural network [11], and deep learning [12, 13]. Ravikumar et al. [14] proposed a fault diagnosis model that includes a multi-scale deep residual learning with a stacked long short-term memory (MDRL-SLSTM) to cope with sequence data in a gearbox health prediction process in an internal combustion engine. A novel learning rate scheduler based on the reinforcement learning (RL) for convolutional neural network (RL-CNN) [15] is presented and applied in fault classification of three bearing datasets. As a single hidden layer feedforward neural network, extreme learning machine (ELM) has also been widely used by researchers. But it also has some disadvantages. For example, the input weights and thresholds of the ELM network are randomly generated, and some almost worthless nodes will inevitably be generated, which will cause the solution accuracy of ELM model to decrease seriously, or even fall into local convergence [16].

Some optimization algorithms have been tried in the literature to improve the performance of ELM networks. For example, Zeng et al. [17] combined ELM with an improved particle swarm optimization (PSO) algorithm that added a delay factor to short-term load forecasting of power systems, to avoid the addition of unnecessary hidden layer nodes and overtraining, and to improve the generalization of ELM networks. Jiang et al. [18] proposed an ELM model combined with a compound differential evolution algorithm and applied it to the prediction of the silicon content of molten iron in a blast furnace. Kadri et al. [19] applied an ant colony (ACO) algorithm with ELM in the fault detection and identification of cement rotary kilns. Wang et al. [20] introduced a fault diagnosis scheme based on multi-scale diversity entropy (MDE) and ELM, which was verified by simulated signals and two experimental data from the bearing and the dual-rotator of the aeroengine test in gears.

Swarm optimization algorithm, such as the PSO and ACO, is a kind of algorithm inspired by the intelligent cooperation of biological groups. It has the advantages of easy implementation, good parallelism, and strong adaptability. Recently, new meta-heuristics optimization algorithms based on bionics have sprung up one after another, such as slap swarm algorithm (SSA) [21], the monarch butterfly optimization (MBO) [22], slime mould algorithm (SMA) [23], moth search algorithm (MSA) [24], hunger games search (HGS) [25], colony predation algorithm (CPA) [26], marine predator algorithm (MPA) [27], Harris hawks optimization (HHO) [28], mayfly algorithm (MA) [29], reptile search algorithm(RSA) [30], and honey badger algorithm (HBA) [31]. As a bionic intelligence algorithm, the antlion optimizer (ALO) simulates the predation mechanism of the antlions in the natural environment [32]. Due to few parameters and high convergence accuracy, it has been used in motor control [33, 34], path planning [35], multi-objective optimization [36], scheduling optimization [37, 38], and other fields [39, 40]. According to the *no free lunch* theorem [41], there is no optimization algorithm that can be universally used in all engineering fields, and the performance of all optimization algorithms on all possible function sets is equivalent. When ALO is used to solve actual complex engineering problems, there are several shortcomings, such as easy precocity of populations. Therefore, some variants of ALO have been proposed in different applications. For example, Yue and Zhang [42] employed a new inertial weight strategy and integrated invasive weed optimization algorithm (IWOA) and bat algorithm (BA) to further improve the performance of the original ALO. Kilic et al. [43] proposed an improved random walk model for the long time-consuming process of ALO, which uses the tournament selection method instead of roulette to improve the global search efficiency of ALO. An improved ALO based on spiral complex path searching pattern is proposed to improve the diversity of the population and the ability of the algorithm to balance exploration and exploitation [44]. Tharwat and Hassanien [45] proposed a new ALO algorithm based on chaos theory and applied it to optimize the initial parameters of SVM model. The classical Lévy flight is also used to improve the ALO [46–48]. The Lévy flight random

walk was suggested as a permutation for performing a local search in Ref. [46], and it is utilized to replace the random walk as the motion of ants in Ref. [47]. But in Ref. [48], it is essentially the fusion of the group search optimization (GSO) algorithm and ALO; that is, the better solutions are selected from the initial random solutions, and they are updated using GSO, whereas the remaining initial solutions are updated using ALO. In the above 3 literatures, the step size control factor w of Lévy distribution is constant and is simply used. The deep fusion with the Lévy strategy is not carried out according to the actual situation.

Therefore, in order to accurately classify the early faults of rolling bearings, a new pattern recognition model based on the improved ALO algorithm and ELM network is proposed in this paper. Firstly, the estimation of distribution algorithm (EDA) and a variable-step Lévy flight strategy are introduced into the initial ALO, called ELALO. To avoid the problem of local optimum and instability during ELM training process, the new ELALO is used to optimize the initial parameters of ELM network and the new pattern recognition is formed, denoted as ELALO-ELM. The experimental results show that the recognition rate of ELALO-ELM model is improved significantly through the case of rolling bearing early fault classification.

The remainder of this paper is organized as follows. The proposed ELALO algorithm is dictated in Section 2, where the superiority of ELALO over conventional ALO is demonstrated by a comparative study. Section 3 introduces the ELALO-ELM network model and its structure, and then applies the hybrid model to the early fault classification of rolling bearings in the area. Section 4 introduces the experimental comparative study using ELM and ELALO-ELM network models. Section 5 summarizes this article.

2. Improved Antlion Optimizer

2.1. Antlion Optimizer. ALO is a meta-heuristic intelligent algorithm that mimics the hunting mechanism of antlions in nature [32]. Five hunting steps are extracted on the predation process, that is, the random walk of ants, trap mining of antlion, ants falling into the trap, antlion preying on ants, and trap reconstruction. According to the above steps, a mathematical model of antlion hunting and ALO algorithm is established. The interactions between ants and antlions in the above steps are all reflected in vectors in multi-dimensional space. In order to expound the meaning of the formula, all the following models are constructed based on dimension $D = 1$.

- (1) The ants walk randomly in the solution space, and the walk step formula can be described as follows:

$$rw(1) = \begin{cases} 1, & \text{rand} < 0.5, \\ -1, & \text{rand} \geq 0.5. \end{cases} \quad (1)$$

$$rw(t+1) = \sum_{t=1}^T rw(t). \quad (2)$$

In Equations (1) and (2), $rw(1)$ represents the initial step value of the first generation of ants to walk. T

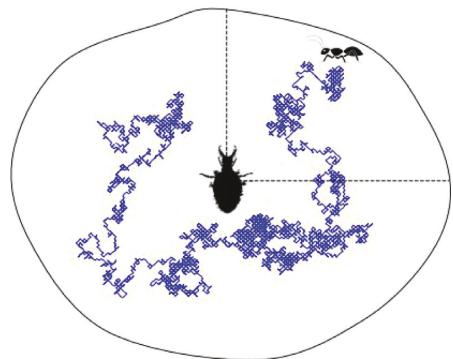


FIGURE 1: A schematic diagram of a random walk around the trap of ALO.

represents the maximum number of iterations, and t is the number of current iterations. $rw(t)$ is the walking step length of the ant from the starting point in $(t+1)^{\text{th}}$ and the step length value is equal to the cumulative sum of all previous generation steps. The rand takes a random number between 0 and 1.

- (2) Antlion digs traps with itself as the center. The process maps to a multi-dimensional space equivalent to constructing a super-sphere trap model near the antlion. The two-dimensional form of the model is shown in Figure 1. Equations (3) and (4) can be used to transform the initial boundary range of the algorithm into a trap enclosure centered on the antlion as follows:

$$c_r(t) = al_s(t) + c(t), \quad (3)$$

$$d_r(t) = al_s(t) + d(t). \quad (4)$$

In Equations (3) and (4), $c(t)$ and $d(t)$, respectively, represent the upper and lower bounds of the generation t trap. $al_s(t)$ represents the antlion individual, which is selected as X_{rs} by the roulette strategy or X_e by the elite strategy in the generation t trap. $c_r(t)$ and $d_r(t)$, respectively, represent the upper and lower bounds of the coordinate of the generation t trap relative to the coordinates of the antlion $al_s(t)$.

- (3) To prevent ants from randomly walking beyond the boundary, the standardization of the walking range is carried out by the following:

$$R_{sd}(t) = \frac{(rw(t) - a(T)) \cdot (b(T) - c_r(t))}{d_r(t) - a(t)} + c_r(t), \quad (5)$$

where $R_{sd}(t)$ represents the standardized random walking step length of the t^{th} generation. $a(T)$ and $b(T)$, respectively, represent the maximum and minimum of the walking step length of the ant after reaching T times of walking around the antlion.

- (4) The random walk of each ant in the range of X_{rs} and X_e can be expressed as

$$\text{ant}(t+1) = \frac{R_{rs}(t) + R_e(t)}{2}, \quad (6)$$

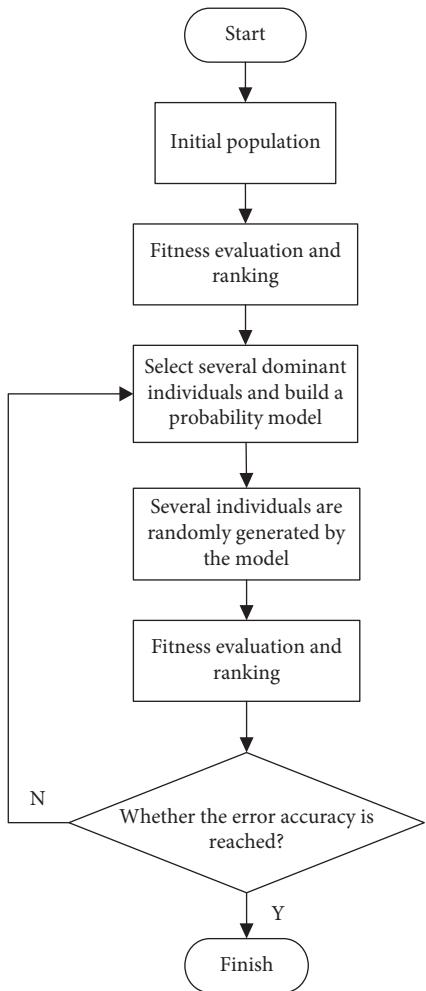


FIGURE 2: Flow chart of EDA.

where $\text{ant}(t+1)$ represents the updated ant individual of the generation $t+1$. When the center antlion, which is surrounded by ants, is determined by the $X_{rs}, R_{sd}(t)$ can be recorded as $R_{rs}(t)$. Similarly, when the center antlion, which is surrounded by ants, is the elite antlion $X_e, R_{sd}(t)$ can be recorded as $R_e(t)$.

- (5) The antlion pulls the ant inside sand and consumes its body. This process occurs under the condition that the fitness value of the ant is better than that of the antlion, and the antlion will replace the ant's position. If the minimum fitness value is optimal, the process can be described as

$$\text{al}(t+1) = \begin{cases} \text{ant}(t+1) & \text{if } f(\text{ant}(t+1)) < f(\text{al}(t)), \\ \text{al}(t) & \text{otherwise,} \end{cases} \quad (7)$$

- (6) When the grain reserves are exhausted, the antlion will repair the traps to wait for the next hunting. For this behavior, ALO algorithm is assumed that the scope of the trap will be adaptively reduced with the iterative process to improve the ability of local

optimization. Mathematical models are shown in Equations (8) and (9):

$$c(t) = \frac{c(t-1)}{I(t)}, \quad (8)$$

$$d(t) = \frac{d(t-1)}{I(t)}, \quad (9)$$

where $I(t) = 10wt/T$, and w is given different constant values in different iterative stages of the algorithm.

2.2. Improved Antlion Optimizer

2.2.1. Introduction of an Estimation of Distribution Algorithm. The EDA [49] is an algorithm like genetic algorithm that uses the strategy of establishing a probability model to replace the population optimization change of the genetic operator. According to the different learning and sampling forms of probabilistic models, EDA has developed many different forms of algorithms, which are widely used in multi-objective optimization [50], path planning [51], machine learning [52], hierarchical cluster analysis [53, 54], and other fields.

According to the differences of the variable forms constructed by the probability model, EDA can be divided into the discrete type and the continuous type. This article mainly deals with the usage of the continuous EDA. The algorithm flow chart is shown in Figure 2.

Based on the probability model calculated by each generation, EDA generates new individuals based on the probability model calculated by each generation, which breaks through the range limitation of ALO random walk and makes it easier to achieve the global optimum. Since each generation of EDA is generated based on a probabilistic model, it is not only effective in solving nonlinear coupling problems, but also can directly describe the evolutionary trend of the population from a macro level. On this basis, the EDA probability model is introduced into the initial ALO model to realize the idea of population generation. The ALO algorithm in the population generation stage includes the initial population stage and the stage where the antlion and ant populations are mixed to form the next generation population at the end of each generation. In these newly generated populations, individuals with a certain fitness value are selected (the selection rate in this paper is 50%). And the average and variance of each individual dimension are calculated and used to build a Gaussian probability model. Individuals randomly generated by the probability model replace some of the individuals in the population that are lagging in fitness. Given that z dominant individuals are selected to construct the model, the set of positions of the i^{th} dimension z individual positions is $U^i = (u_1^i, u_2^i, \dots, u_z^i)$.

$$A^i = \text{gauss}\left(\bar{u}^i, \varepsilon^i\right), \quad (10)$$

where $\text{gauss}()$ indicates that a new population A^i that meets the requirements of mean \bar{u}^i and variance ε^i is generated within the scope of the i^{th} dimensional solution space.

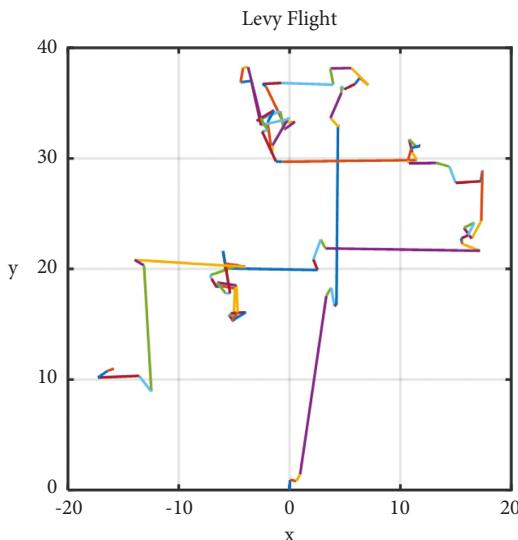


FIGURE 3: Lévy flight trajectory of the generation 100.

2.2.2. Introduction of Variable Step Size Lévy Flight Strategy. The Lévy flight is an optimization strategy obeying the Lévy probability distribution [55]. The principle of Brownian motion in physics can correspond to this search strategy well. In the cuckoo algorithm [56], the bird renewal formula is based on this model. Lévy flight can quickly change the orientation in the solution space, so when applied to the optimization field, it can guide the algorithm to quickly jump out of the local optimum and improve the performance of the algorithm. The position updating formula of Lévy flight is as follows [57]:

$$p_i^{t+1} = p_i^t + \omega \oplus s, \quad i = 1, 2, \dots, A, \quad (11)$$

where p_i represents the i^{th} individual position in the population and the superscript indicates the current generation of the individual. The product \oplus means entry-wise multiplications and ω represents the step adjustment factor which should be related to the scales of the problem of interest. A is the population size and s represents a random step. As the implementation of the Lévy distribution mathematical model is relatively complicated, the Mantegna algorithm is commonly used in practical applications to simulate the Lévy distribution. The algorithm model is as follows.

The random step length s is calculated as

$$s = \frac{\mu}{|\nu|^{1/\zeta}}, \quad (12)$$

where μ and ν are Gaussian distributions and satisfy Equations (13) and (14):

$$\mu \sim N(0, \sigma_\mu^2), \quad (13)$$

$$\nu \sim N(0, \sigma_\nu^2), \quad (14)$$

where $\sigma_\nu = 1$ and σ_μ is expressed as follows:

$$\sigma_\mu = \left[\frac{\Gamma(1 + \zeta)^* \sin(\pi\zeta/2)}{\Gamma[(1 + \zeta)/2]\zeta^* 2^{(\zeta-1)/2}} \right]^{1/\zeta}. \quad (15)$$

In (15), the value of symbol ζ is generally 1.5.

Taking the coordinate origin $(0, 0)$ as the initial point, the Lévy flight trajectory of the generation 100 is shown in Figure 3.

Figure 3 verifies the characteristics of the combination of long and short jump distances of Lévy flight, and it is very helpful to fully explore the solution space. After the Gaussian probability model of EDA is added to ALO, the search accuracy of ALO is improved, but the number of iterations is limited in practical application, and it is particularly important to accelerate the speed of jumping out of the local optimum. According to the Lévy perturbation method made by the PSO algorithm in [58], the new strategy, which integrated the step shrinkage factor and the particle swarm inertia weight parameter with similar function characteristics (which should follow the previous step to strengthen the global convergence ability and the feature of strengthening the local optimization ability in the later stage), is proposed and applied to ALO. Specific steps are as follows.

Step 1. After each iteration of ALO, it is judged whether the fitness value of the elite individual changes within 3 generations. If there is no change, enter Step 2. Otherwise, enter the next iteration.

Step 2. Take the elite individual as the parent and perform g Lévy flights to generate a candidate point set containing g points. Elite individuals follow (16)–(18) for Lévy disturbance. One can see that

$$P_t = \{p_t^1, p_t^2, \dots, p_t^g\}, \quad (16)$$

$$p_t^i = e_t + \omega_t \oplus s, \quad i = 1, 2, \dots, g, \quad (17)$$

$$\omega_t = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \frac{t}{T}, \quad (18)$$

where P_t is the candidate set of the g elements of the t^{th} generation and P_t^i represents the i^{th} element in P_t . e_t represents the elite individuals obtained by the generation t fused with EDA and the symbol ωt represents a linearly decreasing random step adjustment factor, where its maximum and minimum values are $\omega_{\max} = 0.8$ and $\omega_{\min} = 0.1$, respectively.

Step 3. Judge whether the candidate point set is superior to the elite individual of the current generation. And if so, replace the elite individual of the current generation with the elite individual of the next generation. Otherwise, elite individuals of this generation are retained. One can see that

$$\begin{cases} e_{t+1} = \min(f(P_t)) & \text{if } \min(f(P_t)) < f(e_t), \\ e_{t+1} = e_t & \text{otherwise.} \end{cases} \quad (19)$$

The new improved algorithm based the above two strategies is called as ELALO. To summarize, the process of ELALO algorithm is shown in Figure 4.

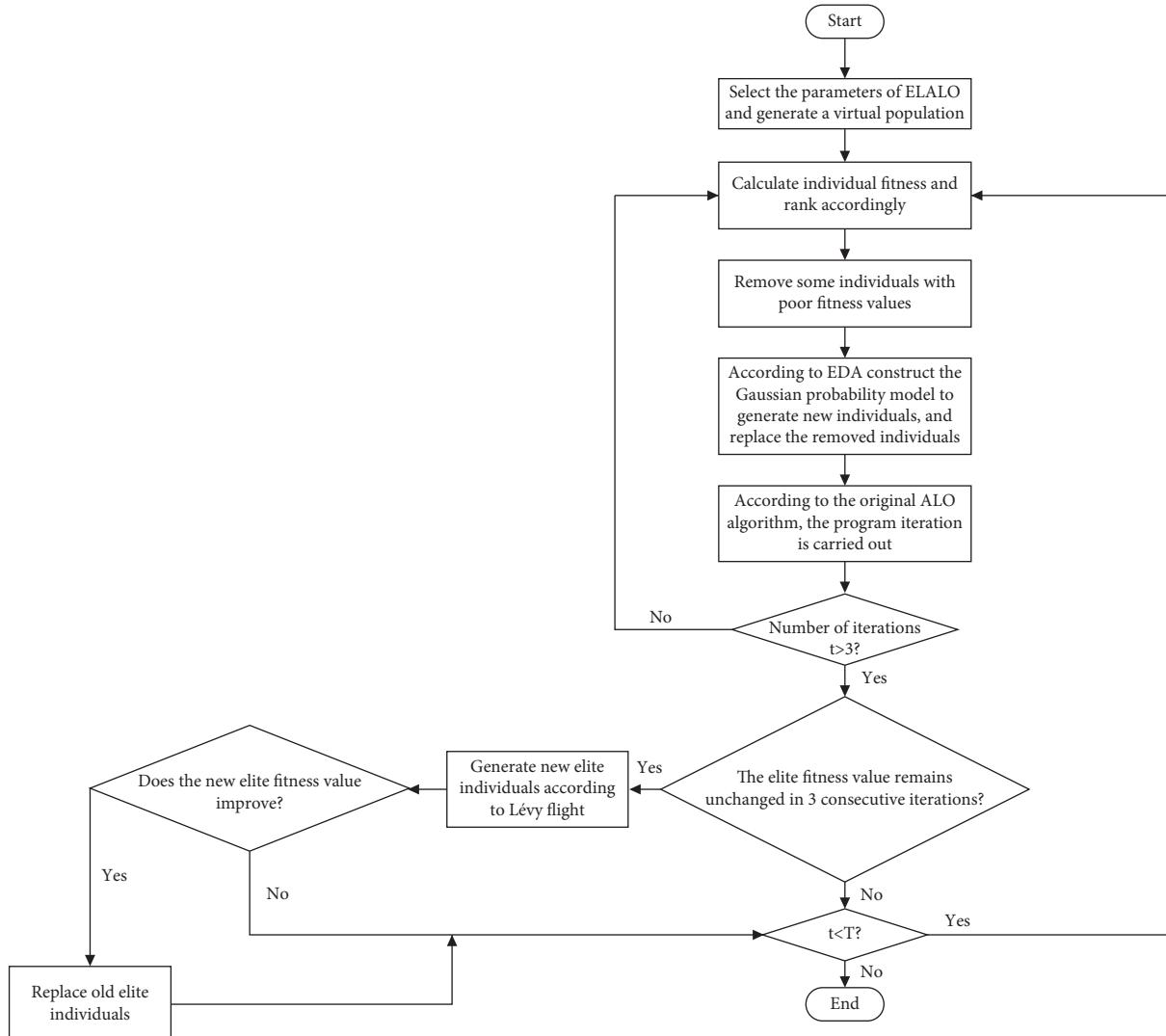


FIGURE 4: Flow chart of ELALO algorithm.

2.3. Verification with Simulation Studies. In order to test the optimize performance of ELALO, sphere function (F_1), Ackley function (F_2), Lévy function (F_3), and Rosenbrock's valley function (F_4) are used in this simulation experiment. The specific expressions are shown in reference [59].

The sphere function is a typical nonlinear symmetric benchmark function and is mainly used to test the solution accuracy and speed of the algorithm in practical applications. The Ackley function is a multimodal function with particularly dense peaks and troughs. Lévy function is a complex multimodal function, where many algorithms will fall into the premature state and cannot get the global optimal solution. Multimodal functions are mainly used to test the algorithm's ability to jump out of local optimum. The Rosenbrock's valley function is a typically ill-conditioned, non-convex, and uni-modal function which is difficult to minimize, and there is an obvious correlation between variables. The mesh graphic of 4 benchmark functions ($D=2$) is shown in Figure 5.

According to the recommended values in references [32, 55–57], the basic parameters of the four algorithms are

the same, such as the population size $A = 500$, and the maximum number of iterations $T = 2000$. And in Lévy flight process, the time g is set to 100. Other parameters of PSO and GA are set according to Ref. [17, 21], as shown in Table 1.

Each algorithm runs 50 times independently. The dimension of the four benchmark functions is $D = 15$. The mean is used as the accuracy index, the best is used as the global optimization performance index, and the variance is as the stability index. The p -values mean the result of the Wilcoxon rank-sum test over benchmark functions. The statistical results are shown in Table 2.

It can be seen from Table 2 that the mean value of ELALO algorithm in function F_3 is 4.76E-15. Compared with 1.79E-01 corresponding to initial ALO, ELALO algorithm has 14 orders of magnitude advantages and has similar performance in functions F_1 , F_2 , and F_4 , which fully indicates that the convergence accuracy of the improved algorithm has been significantly improved. In the test function F_2 , the optimal value of ELALO algorithm is 2.95E-13. Compared with 1.30E-06 corresponding to initial ALO, its global optimization performance is improved. And there are similar

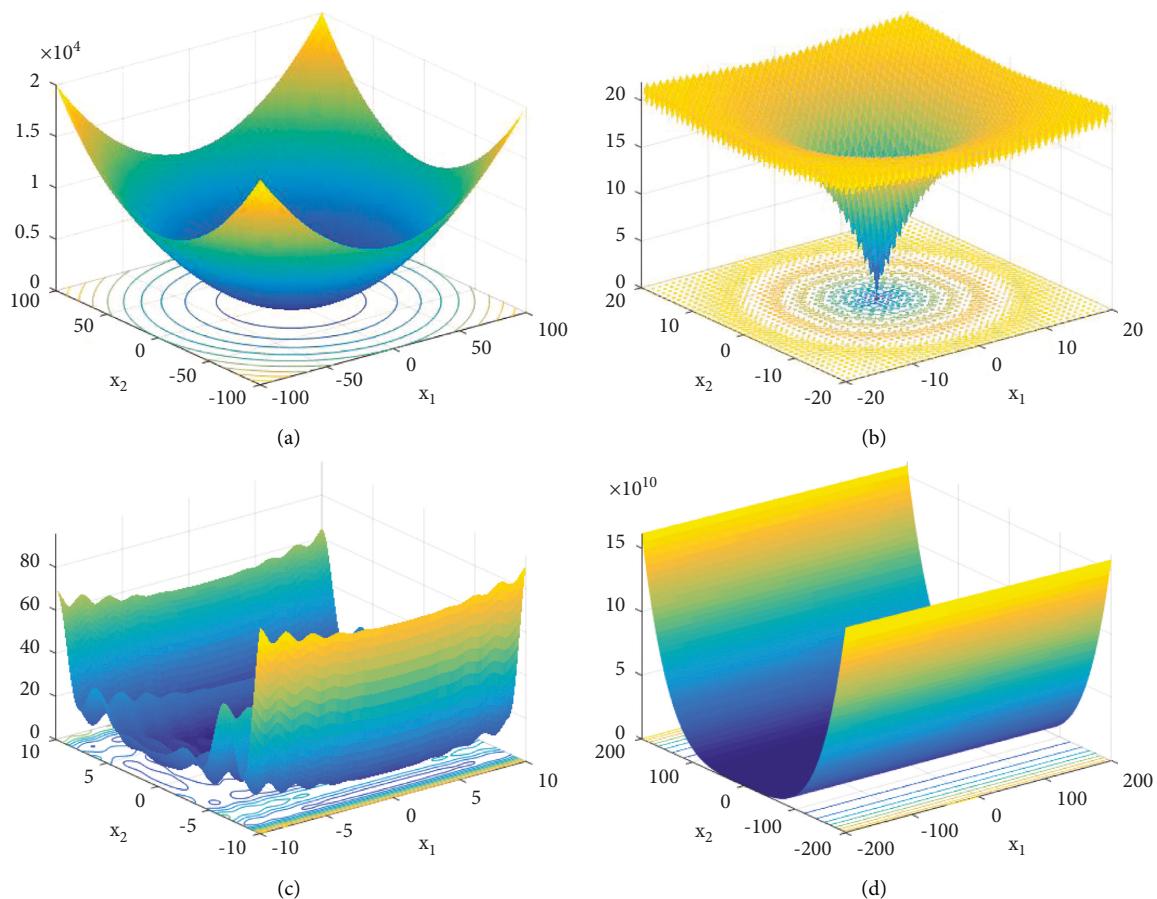


FIGURE 5: Mesh graphic of four benchmark functions ($D=2$):(a) Sphere function(F_1). (b) Ackley function(F_2). (c) Lévy function(F_3). (d) Rosenbrock's valley function(F_4).

TABLE 1: Other parameters setting of GA and PSO.

algorithm	Parameters setting
PSO	$w_{\min} = 0.1, w_{\max} = 1.1, c_1 = c_2 = 1.49$
GA	$n_{elite} = 25, p_c = 0.8, p_m = 0.01$

conclusions on functions F_1 , F_3 , and F_4 . The variance of ELALO algorithm in function F_3 is 3.21E-25. Compared with the corresponding variance of ALO algorithm (4.73E+04), it has an advantage of nearly 29 orders of magnitude, which strongly proves the stability of ELALO and has similar conclusions on functions F_1 , F_2 , and F_4 . Compared with PSO, ELALO algorithm is better than PSO except the F_3 function. When solving the multimodal function F_3 , the average accuracy of the solution is the same order of magnitude, but its variance is less than PSO, indicating that the stability of ELALO algorithm is better than PSO. Compared with GA, there is a similar situation, except for solving F_4 . When solving the ill-conditioned uni-modal function F_4 , the stability of GA is better than ELALO. Overall, the performance of ELALO algorithm has been significantly improved.

The Wilcoxon rank-sum test is a famous nonparametric test in statistics theory that can be applied to determine if two sets of solutions (population) are different statistically

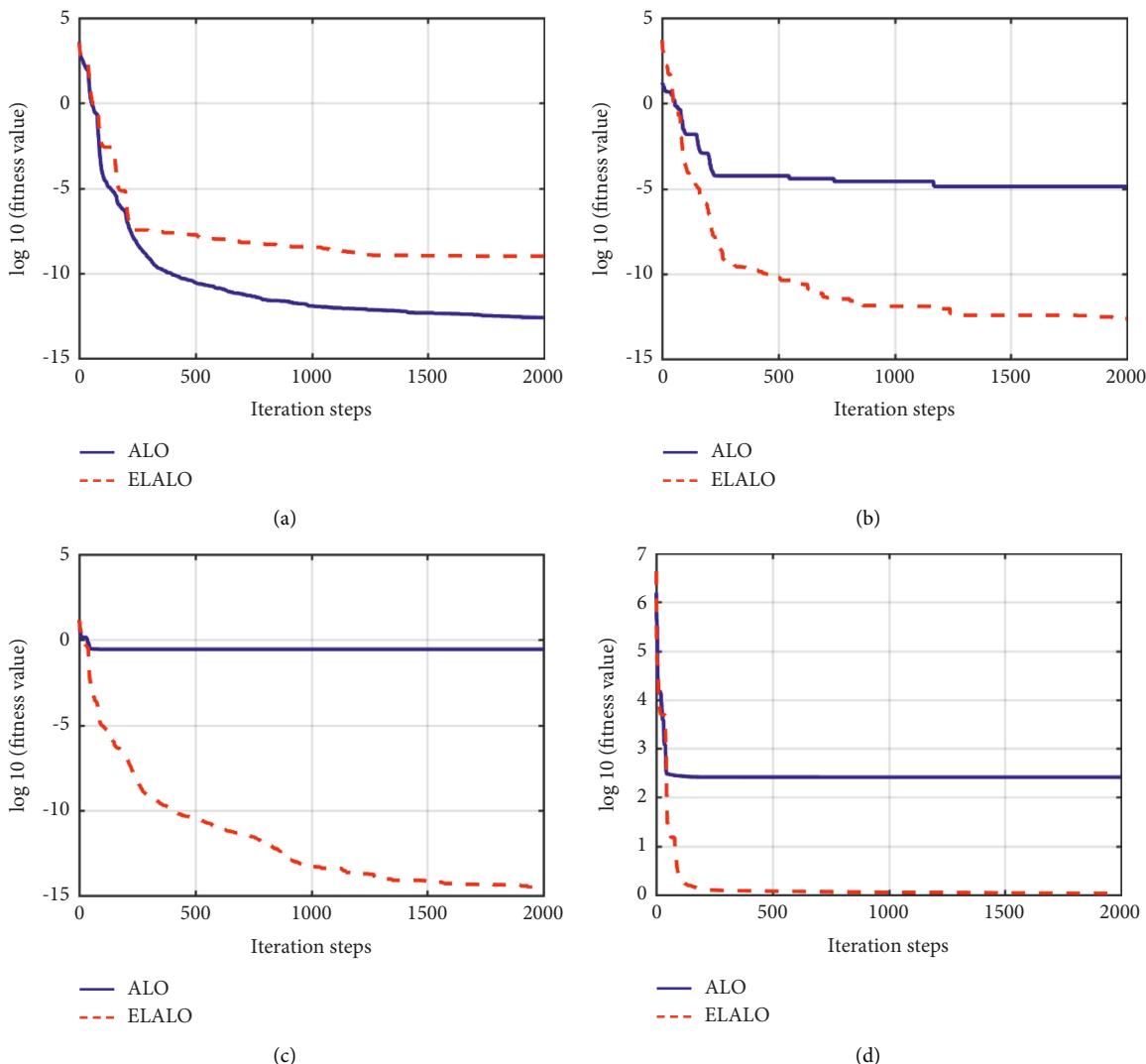
significant or not. In this way, each set of pairs in both populations is compared to calculate and analyze their numerical differences. It also tests the null hypothesis as to whether both populations are of the same distribution. In short, the Wilcoxon rank-sum test returns a numerical result called p -value. A p -value determines the significance level of two algorithms. In this work an algorithm is statistically significant if and only if it results in a p -value less than 5%. The p values in Table 2 also show that this superiority is statistically significant since all p -values are much less than 5%, which further reflect the robustness of ELALO algorithm.

In order to intuitively distinguish the performance characteristics of the two algorithms, the convergence curves of the two algorithms in each test function are shown in Figure 6. In particular, the value on the convergence curve is the average after 50 independent operations of the two algorithms.

In Figure 6, the abscissa is the number of iterations of the algorithm, and the maximum number of iterations is 2000 generations. In order to make a clear distinction between the algorithm iteration curves, the logarithm of the fitness value with base 10 is taken as the ordinate. The optimization accuracy of ELALO has been significantly improved. When the algorithm reaches the same accuracy in the early stage,

TABLE 2: Comparison results in four benchmark functions.

benchmark function	Algorithm	Mean	Best	Variance	<i>p</i> values
F_1	ELALO	4.46E-13	6.23E-15	9.46E-22	—
	ALO	7.19E-09	2.13E-10	1.57E-19	7.07E-18
	PSO	5.80E-08	9.77E-09	6.60E-08	7.07E-18
	GA	3.01E-06	1.43E-07	2.94E-06	7.07E-18
F_2	ELALO	1.33E-12	2.95E-13	3.17E+00	—
	ALO	4.84E-05	1.30E-06	1.08E+01	3.31E-20
	PSO	2.08E+01	2.05E+01	1.22 E-01	3.31E-20
	GA	7.37E-05	3.71E-05	1.88E-05	3.31E-20
F_3	ELALO	4.76E-15	1.87E-15	3.21E-25	—
	ALO	1.79E-01	4.69E-03	4.73E+04	1.52E-05
	PSO	3.45E-15	2.46E-29	2.28E-14	1.60E-02
	GA	1.83E-11	1.75E-16	4.77E-11	4.68E-04
F_4	ELALO	1.42E+00	8.28E-01	1.56E+02	—
	ALO	2.30E+02	5.07E+01	2.44E+02	2.27E-02
	PSO	5.65E+02	1.77E-02	1.38E+03	7.07E-18
	GA	1.96E+00	5.57E-04	2.26E+00	2.05E-09

FIGURE 6: Comparison of convergence curve. (a) Curve comparison on F_1 , (b) curve comparison on F_2 , (c) curve comparison on F_3 , and (d) curve comparison on F_4 .

the convergence speed of the ELALO is also higher than the initial ALO. Hence, one can see that the proposed ELALO algorithm has a good convergence.

2.4. Time Complexity Analysis. Time complexity is the key factor of algorithm performance and reflects the operation efficiency of the algorithm. In ALO, the population size is A , the individual dimension is D , and the maximum number of iterations is T . Suppose that the time to generate the random number and calculate the position of randomly distributed ants in each dimension is δ_1 , the time to calculate the fitness value of the objective function according to the ant position is $f(D)$, and the time to sort according to the ant fitness and select the optimal solution is δ_2 . Because the initial position of the ant and antlion is the same, the initial position of the elite antlion is the optimal solution selected from ant population. Therefore, the time complexity of population initialization stage is $O(A(D\delta_1 + f(D) + \delta_2)) = O(D + f(D))$. After entering the iterative process, the time when the antlion is selected is δ_3 , the time when the ants generate the position and judge the walking direction around an antlion is δ_4 and δ_5 , respectively, the time for the ant to change the boundary in the process of walking is δ_6 , and the time when the ant obtains the new position after walking is δ_7 . The time to calculate the fitness value according to the ant position is $f(D)$ and the time of fitness value comparison and position replacement between ant and antlion is set to δ_8 . So, the time complexity of this stage is $O(T(A(\delta_4 + \delta_5 + \delta_7 + f(D) + \delta_8) + \delta_3 + \delta_6)) = O(f(D))$. Therefore, the maximum time complexity of ALO algorithm is $O(D + f(D)) + O(f(D)) = O(D + f(D))$ [60].

The population size, dimension, maximum number of iterations, and the time to solve the value of objective or fitness function of ELALO algorithm are the same as those of ALO. Therefore, the time complexity of population initialization stage is the same as that of ALO; that is, $O(A(D\delta_1 + f(D) + \delta_2)) = O(D + f(D))$. After entering the iteration, suppose that the time to construct the Gaussian distribution probability model with EDA is τ_1 , the time to eliminate the individuals with poor fitness value is τ_2 , the time to generate new individuals according to EDA is τ_3 , the time to generate elite individuals by Lévy flight is τ_4 , the time to compare the fitness values of the old and new elite individuals is τ_5 , and the time to judge whether to replace is τ_6 . Other calculation times are the same as those in ALO algorithm. So, the time complexity of this stage is $O(T(A(\delta_4 + \delta_5 + \delta_7 + 2f(D) + \delta_8) + 3\tau_1 + 3\tau_2 + 3\tau_3 + \tau_4 + \tau_5 + \tau_6 + \delta_3 + \delta_6)) = O(f(D))$. Therefore, the total time complexity of ELALO algorithm is also $O(D + f(D)) + O(f(D)) = O(D + f(D))$.

It can be seen from the above analysis that compared with the initial ALO, the time complexity of ELALO algorithm does not change, nor does it reduce the operation efficiency of the original algorithm.

3. ELALO-ELM Network Model

In this section, the mentioned ELALO algorithm is merged with the ELM network to develop a new ELALO-ELM network model for pattern classification.

3.1. ELM Network Fundamentals. The ELM is a new artificial neural network model for single-hidden layer feedforward neural networks (SLFNs), which randomly chooses hidden nodes and analytically determines the output weights of SLFNs [61]. In theory, ELM tends to provide good generalization performance at extremely fast learning speed.

Suppose the set of N training samples is $\{X, Y\} = \{x_i, y_i\}_{i=1}^N$, where $x_i = [x_{i1}, x_{i2}, \dots, x_{iD}]^T \in R^D$, $y_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$, D is the feature dimension contained in the sample set, and m is the total number of categories in the sample set. Supposing the number of hidden layer nodes is L and the excitation function is $g(x)$, then the network output expression is shown as follows:

$$f_L(x) = \sum_{i=1}^L \xi_i g(a_i \cdot x_i + b_i), \quad x_i \in R^D, a_i \in R^D, \\ \xi_i \in R^m, i = 1, 2, \dots, L, \quad (20)$$

where $a_i = [a_{i1}, a_{i2}, \dots, a_{in}]^T$ represents the weight matrix from the input layer to the i^{th} hidden layer node, b_i is the threshold of the i^{th} hidden layer node, and $\xi_i = [\xi_{i1}, \xi_{i2}, \dots, \xi_{im}]^T$ is the output weight of the i^{th} hidden layer node.

If the network can accurately approximate N samples, there is a_i, b_i and ξ_i satisfying

$$f_L(x) = \sum_{i=1}^L \xi_i g(a_i \cdot x_i + b_i) = y_i, \quad i = 1, 2, \dots, L. \quad (21)$$

Then, (21) can be simplified as follows:

$$H\xi = Y. \quad (22)$$

H represents the output matrix of the hidden layer of the network. In operation, the initial weight and threshold matrix are randomly assigned by the program, so that the training of the ELM model is completely transformed into the least squares solution problem in the mathematical sense. By (22), the least square solution form of the output weight matrix can be expressed as

$$\tilde{\xi} = H^+ Y, \quad (23)$$

where H^+ represents the Moore-Penrose generalized inverse of matrix H .

3.2. ELALO-ELM Network Model. The steps to construct the ELALO-ELM model are as follows:

- (1) Determine the number k of ELM hidden layer nodes and the number of input layer features d .
- (2) Set the ELALO individual dimension $D = k(d + 1)$ and initialize the parameters of ELALO, and then any individual θ in the algorithm can be expressed as

$$\theta = [\rho_1, \rho_2, \dots, \rho_{k(d+1)}]. \quad (24)$$

- (3) Note that the initial weight matrix of ELM is W and the threshold matrix is B_i , input the individual θ into ELM, and calculate the fitness value of θ by selecting

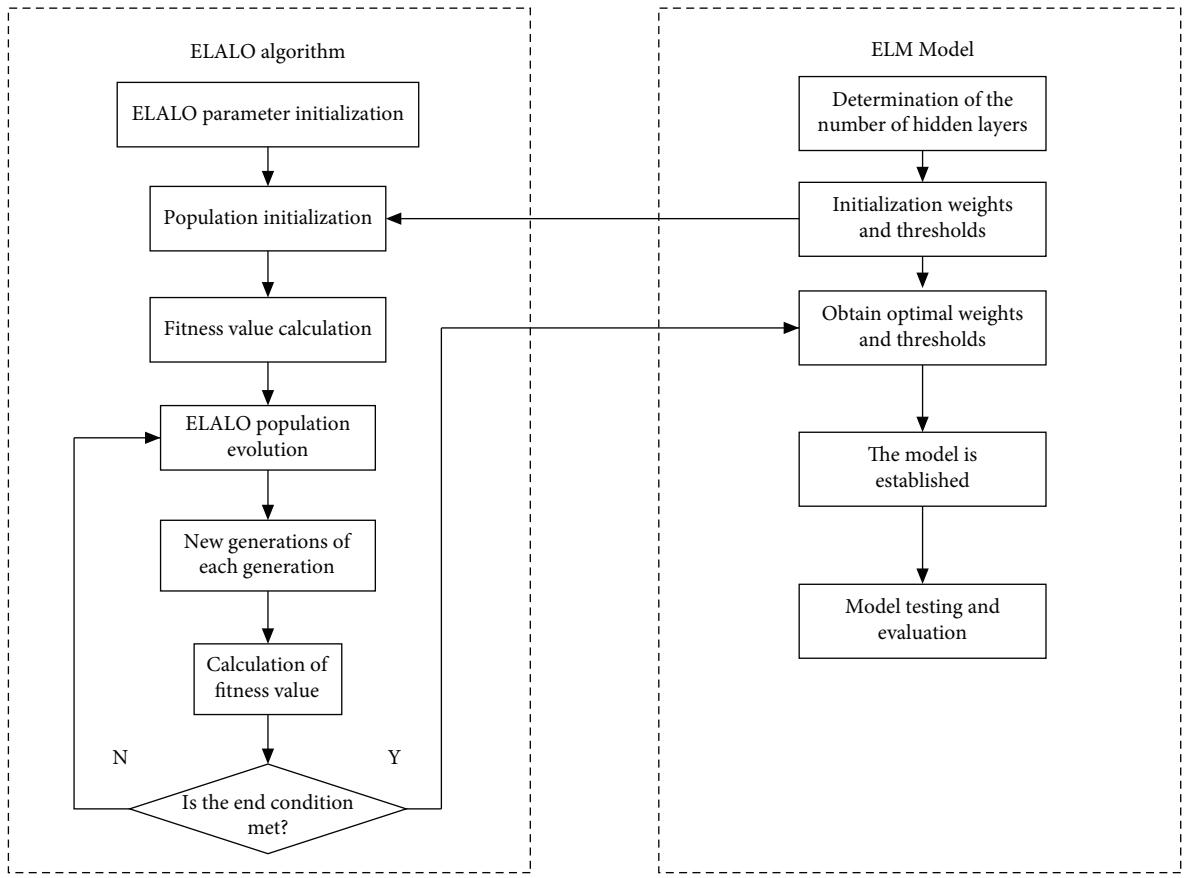


FIGURE 7: Flow chart of ELALO-ELM.

the appropriate fitness function. Then, W and B_i can be expressed as

$$W = \begin{bmatrix} \rho_1 & \rho_2 & \dots & \rho_k \\ \rho_{k+1} & \rho_{k+2} & \dots & \rho_{2k} \\ \vdots & \vdots & \dots & \vdots \\ \rho_{(d-1)k+1} & \rho_{(d-1)k+2} & \dots & \rho_{dk} \end{bmatrix}, \quad (25)$$

$$B_i = [\rho_{dk+1} \rho_{dk+2} \dots \rho_{k(d+1)}].$$

- (4) Repeat step 3 for each generation of ELALO population individuals to find the weight and threshold corresponding to the optimal fitness value.
- (5) Assign the optimal weights and thresholds calculated by ELALO to ELM, complete the establishment of the ELALO-ELM model, and input the test set into the model to evaluate the performance of the model.

Based on the above steps, the realization process of the ELALO-ELM model is shown in Figure 7.

4. Using ELALO-ELM Network for Early Fault Diagnosis of Roller Bearings

This section uses the fault data from the drive end of the rolling bearing fault platform of Case Western Reserve University in the United States [62]. The experiment uses electric spark technology to cause damage to different parts

of the bearing in different sizes to simulate the early fault of the roller bearing. The acceleration vibration sensor is used to collect bearing signal, and sampling frequency is 12 kHz [63].

The damage degree of the rolling bearing fault is distinguished by the diameter of the fault. 8 types of rolling bearing states of different sizes are extracted, rolling ball faults, outer ring faults, and normal bearings to form a fault set, which are denoted as $FM = \{fm_1, fm_2, fm_3, fm_4, fm_5, fm_6, fm_7, fm_8\}$. The fm_1, fm_2, fm_3 , and fm_4 , respectively, represent the faults of the rolling ball with diameters of 0.1778 mm, 0.3556 mm, 0.5334 mm, and 0.7112 mm, and fm_5, fm_6 , and fm_7 represent the faults of the outer ring with diameters of 0.1778 mm, 0.3556 mm, and 0.5334 mm, respectively. fm_8 means normal bearing. Using data of different fault categories as the input samples of the ELALO-ELM model can make more accurate performance evaluation of the model. According to literature [64], the sampling points of each group of data to the original data are 2048. The time-domain waveforms of the 8 bearing states are shown in Figure 8.

4.1. Fault Feature Extraction Based on LMD and Sample Entropy

4.1.1. Introduction of Estimation of Distribution Algorithm. Local mean decomposition (LMD) [65] is a feature extraction method for adaptive time-frequency analysis of non-stationary signal. The vibration signals of early failures

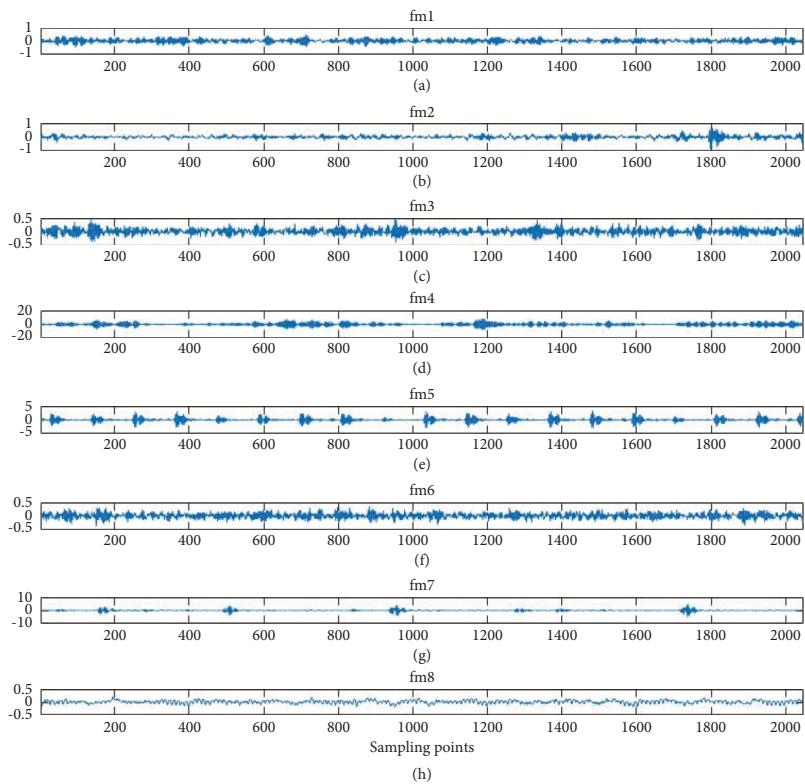


FIGURE 8: Bearing waveforms on time-domain.

of rolling bearings are not easy to find obvious features, but they have the characteristics of non-linearity and non-stationarity. LMD can efficiently extract early fault characteristics of rolling bearings, and thus improve the effectiveness and correlation of candidate feature sets. The LMD generally decomposes the original signal $x(t)$ through (26) and decomposes it into several PF(t) components that conform to the essential law of characteristics and a residual component $\mu_k(t)$ with weak characteristic performance. This component can describe the signal characteristics in more details. The detailed steps of this method can be found in the literature [65], which will not be repeated here

$$x(t) = \sum_{p=1}^k \text{PF}_p(t) + \mu_k(t). \quad (26)$$

4.1.2. Symbolic Entropy Theory. Symbol entropy [66] is a signal complexity measurement method based on information entropy, which is simple and efficient. A random sequence $\{a_n\}$ is provided, and the sequence is symbolized by

$$s_n = \begin{cases} 1 & a_n > \bar{m}, \\ 0 & a_n < \bar{m}, \end{cases} \quad (27)$$

where \bar{m} is the mean value of sequence $\{a_n\}$. After converting $\{a_n\}$ into symbol sequence $\{s_n\}$, which is composed of 0 and 1, it needs to be decomposed into multiple short symbol sequences by

$$\bar{s}(i) = s(i), s(i+1), \dots, s(i+(L-1)) \quad i = 1, 2, \dots, B-L+1. \quad (28)$$

L represents the length of each symbol sequence after decomposition, and B represents the total length of the original sequence $\{s_n\}$. Convert all sequences $\bar{s}(i)$ into decimal sequences and the code sequence can be obtained [67]. For example, the original symbol sequence $\{s_n\}$ is $\{100101001\}$, when $L=3$, it can be decomposed into a short sequence of $\{100,001,010,101,010,100,001\}$ and converted into a decimal coding sequence as $\{4,1,2,5,2,4,1\}$.

Then, the entropy of the above symbol sequences is calculated according to

$$H_s(L) = -\frac{1}{\log M} \sum P_{m,L} \log(P_{m,L}), \quad (29)$$

where $H_s(L)$ represents the symbol entropy value, $P_{m,L}$ is the relative overall occupancy rate of each digital code in the decimal code sequence, and M represents the number of 0~9 digital categories in the sequence.

Figures 9(a)–9(h) show the LMD exploded view of 8 different states of the rolling bearing, respectively.

It can be seen from Figure 9 that the bearing signal is mainly concentrated in the first 3 PF component signals after LMD. Therefore, the symbol entropy of the PF component is calculated separately and used as the feature quantity of the input classifier. And 50 sets of samples are taken for each state of the bearing, and the PF symbol entropy is calculated as shown in Table 3.

It can be seen from Table 3 that the difference between the PF symbol entropy features based on LMD is obvious, and the distribution is stable and the aggregation degree is



FIGURE 9: Continued.

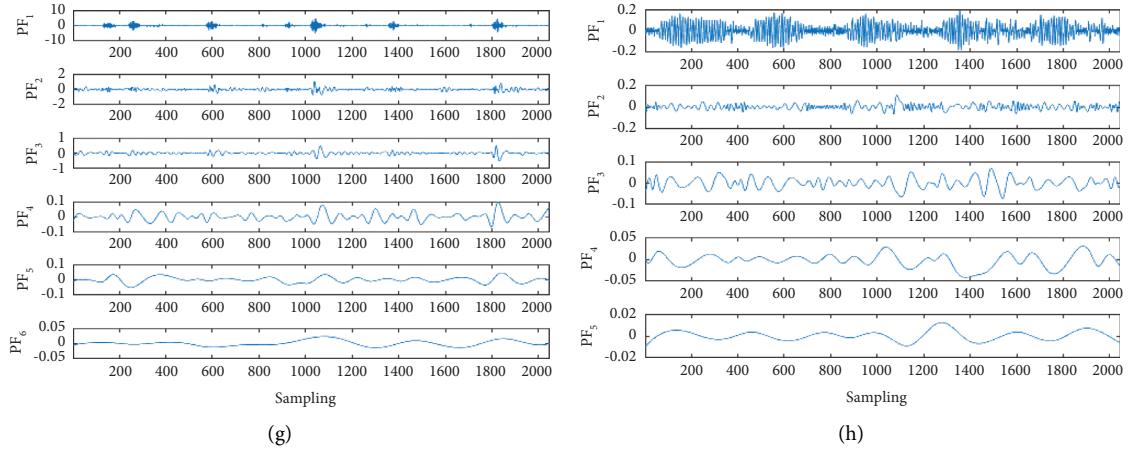


FIGURE 9: FF1–PF6 signal of the 8 bearing states: (a) PF1–PF6 signal of LMD on fm₁. (b) PF1–PF6 signal of LMD on fm₂. (c) PF1–PF6 signal of LMD on fm₃. (d) PF1–PF6 signal of LMD on fm₄. (e) PF1–PF6 signal of LMD on fm₅. (f) PF1–PF6 signal of LMD on fm₆. (g) PF1–PF6 signal of LMD on fm₇. (h) PF1–PF6 signal of LMD on fm₈.

TABLE 3: The first 3 PF symbol entropy of 8 states of bearings.

bearing status	Index	PF1	PF2	PF3
fm ₁	Mean	1.938	1.913	1.565
	Variance	0.0002	0.0074	0.0019
fm ₂	Mean	2.020	1.884	1.542
	Variance	0.0015	0.0054	0.0023
fm ₃	Mean	2.031	1.872	1.563
	Variance	0.0003	0.0035	0.0026
fm ₄	Mean	1.917	1.920	1.622
	Variance	0.0003	0.0034	0.0022
fm ₅	Mean	2.026	1.873	1.559
	Variance	0.0038	0.0055	0.0020
fm ₆	Mean	2.096	1.872	1.564
	Variance	0.0003	0.0061	0.0025
fm ₇	Mean	2.076	1.801	1.493
	Variance	0.0004	0.0054	0.0019
fm ₈	Mean	1.933	1.585	1.227
	Variance	0.0002	0.0040	0.0003

high. Using it as the input feature of the classifier can significantly reduce the learning difficulty of the classifier (Table 4).

4.2. Experimental Results. Since the content of this section mainly uses the bearing data set to evaluate the performance of the ELALO-ELM model, the performance evaluation index can be measured by the mean value of the failure recognition rate. So, the fitness function is constructed as

$$\max f(x) = \frac{1}{\text{count}} \sum_{i=1}^{\text{count}} \text{accuracy}_i, \quad (30)$$

where the count represents the bearing state category and accuracy i represents the data recognition rate percentage of the i th category.

TABLE 4: Feature input and ideal output.

Bearing status	Input feature			Ideal output
	PF1	PF2	PF3	
fm ₁	1.943	2.087	1.575	1000 0000
fm ₂	2.029	1.899	1.426	0100 0000
fm ₃	2.051	1.809	1.569	0010 0000
fm ₄	1.903	1.978	1.488	0001 0000
fm ₅	2.026	1.798	1.538	0000 1000
fm ₆	2.111	1.955	1.578	0000 0100
fm ₇	2.093	1.851	1.400	0000 0010
fm ₈	1.944	1.656	1.229	0000 0001

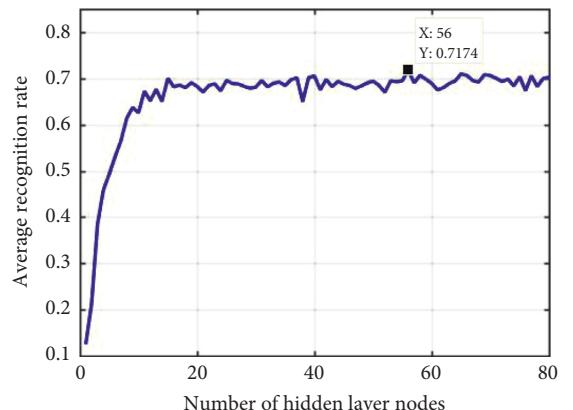


FIGURE 10: The average recognition rate curve.

Because the number of hidden layer nodes of ELM has a great influence on the results of the model. According to the literature [68], the number of hidden layer nodes of ELM is selected in order from 1 to 80. And taking (26) as the evaluation index, the curve of the average recognition rate with hidden layer nodes is obtained. As shown in Figure 10, the peak value of the curve shows that the best number of hidden layer nodes is 56.

TABLE 5: Actual output comparison of ELM and ELALO-ELM.

status	Network	Actual network output						
		elm	0.512	0.362	0.004	0.112	0.043	-0.139
fm ₁	ELALO-ELM	0.790	0.243	-0.100	-0.032	0.066	-0.040	0.073
	ELALO-ELM	0.025	0.383	-0.135	0.042	0.048	0.348	0.061
fm ₂	ELALO-ELM	-0.069	-0.003	0.505	0.053	0.315	0.303	-0.104
	ELALO-ELM	0.025	0.383	-0.135	0.042	0.048	0.348	-0.014
fm ₃	ELALO-ELM	0.148	0.485	0.079	-0.230	0.442	-0.107	0.183
	ELALO-ELM	-0.216	0.336	0.533	0.173	0.204	-0.118	0.364
fm ₄	ELALO-ELM	0.358	0.132	0.065	0.420	0.029	-0.003	-0.002
	ELALO-ELM	0.237	-0.091	-0.060	0.956	0.050	-0.061	-0.030
fm ₅	ELALO-ELM	-0.029	0.336	0.369	-0.025	0.393	-0.156	0.112
	ELALO-ELM	-0.194	-0.173	0.133	0.397	0.858	-0.425	0.404
fm ₆	ELALO-ELM	-0.009	0.149	0.072	0.021	0.159	0.479	0.126
	ELALO-ELM	0.039	0.237	-0.109	-0.017	0.102	0.659	0.086
fm ₇	ELALO-ELM	0.058	0.178	-0.006	-0.045	-0.041	0.135	0.721
	ELALO-ELM	-0.023	0.034	-0.001	0.004	0.001	-0.214	0.987
fm ₈	ELALO-ELM	0.034	0.492	0.231	0.020	0.317	0.025	-0.121
	ELALO-ELM	-0.082	0.401	0.339	0.061	0.149	-0.036	0.218

Note. The bold characters represent the classification results of the network, in which italics and underline indicate misclassification.

TABLE 6: Recognition rates comparison of 3 networks.

Network	Status								Mean
	fm ₁	fm ₂	fm ₃	fm ₄	fm ₅	fm ₆	fm ₇	fm ₈	
ELM	86.83	23.33	44.03	87.39	56.38	89.07	90.95	95.94	71.74
ALO-ELM	91.22	62.78	67.03	97.74	77.65	96.20	98.41	96.25	85.91
ELALO-ELM	94.40	62.78	72.39	99.02	78.46	98.53	100	98.42	88.00

The ELM, ALO-ELM, and ELALO-ELM network models are used to conduct comparative experiments. Set population size $A = 500$, Lévy flight times $g = 100$, $T = 300$. The three PF symbol entropy features are input to the classifier, so the input node is 3. And the number of bearing status labels is 8. So, the output node is also 8. Since the hidden layer node is set to 56, the method ELALO-ELM model establishment according to Section 3.3 is as follows: $D = 56 * (3 + 1) = 224$.

The actual output results of the standard ELM and ELALO-ELM networks are shown in Table 5. The actual output of the ELM network is far from the ideal value, and classification-errors occur in faults fm₂ and fm₃. The actual output of the ELALO-ELM network is more consistent with the ideal output, indicating that the network model of the fusion improved algorithm ELALO has excellent performance and can more accurately distinguish the different fault characteristics of rolling bearings.

The comparison results of the recognition rates of the three networks solved 20 times are shown in Table 6. The recognition rate of ELM is the lowest when the bearing status are fm₂ and fm₃, reaching 23.33% and 44.03%, respectively. However, the overall recognition rate of new ELM models after parameter optimization is no less than 62.78%, and the recognition rate of each bearing status is significantly improved. When the fault type is fm₇, the recognition rate reaches 98.41%, which is 7.46% higher than the original ELM network. It shows that the ALO algorithm fused in ELM parameter optimization can significantly improve the classification performance. Among them, ELALO-ELM has the

best network recognition rate, with an average recognition rate of 88.00% and has even a recognition rate of 100% for fm₇, which is 16.26% and 2.09% higher than ELM and ALO-ELM respectively. It verifies the effectiveness and superiority of the new ELALO-ELM to improve accuracy of classification model on the bearing early fault.

The broken lines of the recognition rates obtained by using the three networks model for 20 times are shown in Figure 11. The abscissa represents the experiment number, and the ordinate is the bearing status recognition rate calculated by the 3 networks model in each experiment. The fault recognition rate obtained by using only the ELM network reached a peak of 79.65% at the experimental number 12, and the worst value of the ELM fault recognition rate based on the ALO algorithm was 80.94% at the experimental number 17, which shows that the recognition rate of ELM has been significantly improved after the ALO parameter optimization. And most points in the broken line corresponding to ELALO-ELM network model are higher than ALO-ELM, which shows that the overall classification accuracy on bearing early fault of ELALO-ELM is better than that of ALO-ELM.

In order to reflect the advantages of the ELALO-ELM network more intuitively, in this paper, the average of the 20 experimental iteration curves of the two ALO network fusion is shown in Figure 12. It can be seen from Figure 11 that the two kinds of network iteration curves stabilized and converged before 50 generations. Finally, the average recognition rate of ALO-ELM stabilized at 85.91%, and the average recognition rate of ELALO-ELM stabilized at

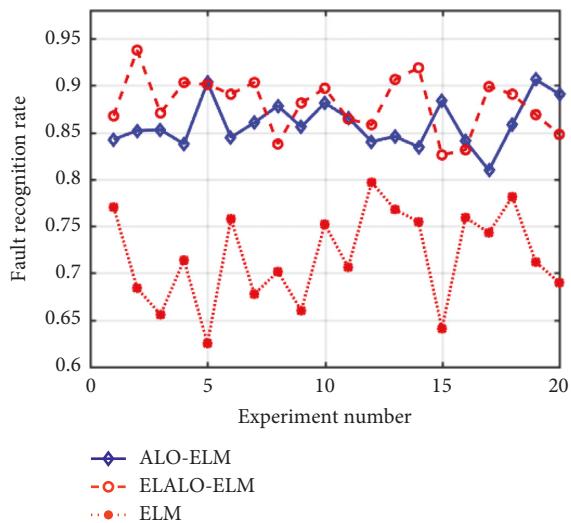


FIGURE 11: The recognition rate results of the three types of networks 20 times.

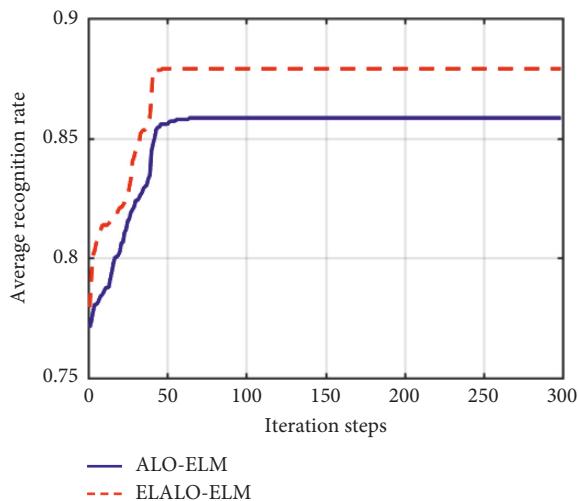


FIGURE 12: Comparison of two network iteration curves.

88.00%. When the number of iteration steps is the same, the iteration curve corresponding to ELALO-ELM is always higher than that of ALO-ELM, indicating that the convergence accuracy of ELALO-ELM is always higher than that of ALO-ELM. When the two network models reach the same recognition rate in the early stage of iteration, the number of iteration steps required by ELALO-ELM is less than that of ALO-ELM, indicating that ELALO-ELM has higher convergence efficiency and faster convergence speed.

5. Conclusions

- (1) This work proposes an improved ALO algorithm, called ELALO, integrates the EDA and the variable-step Lévy flight strategy. The test results on 4 benchmark functions and theoretical analysis show that the ELALO is the same as the original ALO in

terms of computational complexity, but is not easy to fall into the local optimum and superior to ALO algorithm in terms of solution accuracy and convergence speed.

- (2) Aiming at the highly nonlinear and strong coupling characteristics of early fault data of rolling bearing, a method combining LMD and symbolic entropy is proposed to extract the most effective features as the input features of classifier model, which effectively reduces the difficulty of classifier pattern recognition.
- (3) Aiming at the defect of random initialization of initial weight and threshold in the calculation process of ELM network, ELALO is used to optimize its initial parameters to form a new pattern recognition model called ELALO-ELM. The experimental results show that new model can effectively overcome the inherent defect of falling into local optimization caused by random initialization of ELM network, resulting in high accuracy of early fault classification of rolling bearing. The ELALO-ELM model can also be used for other weak fault classification problems.
- (4) In our future work, we will focus on more complex and weaker fault pattern classification and recognition, which is challenging to the global convergence performance of ELALO algorithm and the classification ability of ELALO-ELM model.

Appendix

A

```

Pseudo-code of ELALO algorithm
{
//ELALO args initialization
Init all args of ELALO: static GAUSS_POP, ALO_POP,
EDA_POP, ...
//Interface implementation
function void __ALO_INIT(void)
do
    Initialize ALO_POP
done
function EDA_POP __GAUSS_RND(ALO_POP)
do
    EDA_POP = Initialize GAUSS_POP and Replace
    ALO_POP
done
function void __EDA_INIT(void)
do
    EDA_POP = __ GAUSS_RND(ALO_POP)
done
function void __ELALO_INIT(void)
do
    EDA_POP = __ GAUSS_RND(ALO_POP)
done
}

```

```

__ALO_INIT()
__EDA_INIT()
done
function ELITE_POP __ALO_SEARCH(ELALO_
TEMP_POP)
do
    According to ALO walk
done
function FVAL __FITNESS_VALUE(POP)
do
    Calculate the fitness of POP
done
function CANDIDATE_SET __LEVY_FLIGHT
(ELITE_POP)
do
    According to formula (11)
done
function BEST_POP __ELALO_SEARCH(void)
do
    ELALO_TEMP_POP = EDA_POP
    while ( $t < \text{max\_iterations}$ )
        do
            ELITE_POP = __ALO_SEARCH(ELALO_
TEMP_POP)
            FVAL = __FITNESS_VALUE(ELITE_POP)
            if (FVAL get trapped in continuous 3 generation)
                do
                    CANDIDATE_SET = __LEVY_
FLIGHT(ELITE_POP);
                    if (__FITNESS_VALUE(CANDIDATE_SET) is
good than __FITNESS_VALUE(ELITE_POP))
                        do
                            Replace CANDIDATE_SET to ELITE_POP
                            Update ELITE_POP
                        done
                    else
                        do
                            Reserve ELITE_POP
                        done
                done
            Update ELALO_TEMP_POP
        done
    end while
    BEST_POP = ELALO_TEMP_POP
    return BEST_POP
done
//Main function implementation
void main

```

```

do
__ELALO_INIT()
BEST_POP = __ELALO_SEARCH()
done
}

```

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

This work was supported in part by the Anhui Province Natural Science Foundation under Grant No. 1808085ME127, the Overseas Visiting and Research Project for Outstanding Young Backbone Talents in Universities of Anhui Province under Grant No. gxgwx2019041, the Innovation Project for Returned Overseas Students in Anhui Province under Grant No. 2020LCX013, the Key Research and Development Projects of Anhui Province under Grant No. 202004b11020006, the Scientific Research Foundation of Anhui Polytechnic University under Grant No. 2020YQQ010, the Anhui Polytechnic University Research Initiation Fund for Introducing Talents under Grant No. 2019YQQ004, the Anhui Polytechnic University Research Project under Grant no. Xjky019201905, the Industrial Collaborative Innovation Fund of Anhui Polytechnic University and Jiujiang District under Grant No. 2021cyxtb9, the Open Project of Anhui Provincial Engineering Laboratory on Information Fusion and Control of Intelligent Robot under Grant No. IFCIR2020001, the Science and Technology Planning Project of Anhui Market Supervision and Administration Bureau under Grant no. 2021MK005, and the Open Project of Key Laboratory of Industrial Equipment Quality Big Data Ministry of Industry and Information Technology under Grant No. 2021-IEQBD-05.

References

- [1] J. Liu, Z. Shi, and Y. Shao, "Vibration characteristics of a ball bearing considering point lubrication and nonuniform surface waviness," *International Journal of Acoustics and Vibration*, vol. 23, no. 3, pp. 355–361, 2018.
- [2] H. Cao, S. Su, X. Jing, and D. Li, "Vibration mechanism analysis for cylindrical roller bearings with single/multi defects and compound faults," *Mechanical Systems and Signal Processing*, vol. 144, Article ID 106903, 2020.
- [3] M. Amarnath, V. Sugumaran, and H. Kumar, "Exploiting sound signals for fault diagnosis of bearings using decision tree," *Measurement*, vol. 46, no. 3, pp. 1250–1256, 2013.
- [4] X. Tang, X. Gu, J. Wang, Q. He, F. Zhang, and J. Lu, "A bearing fault diagnosis method based on feature selection feedback network and improved D-S evidence fusion," *IEEE Access*, vol. 8, Article ID 20536, 2020.

- [5] T. Berredjem and M. Benidir, "Bearing faults diagnosis using fuzzy expert system relying on an improved range overlaps and similarity method," *Expert Systems with Applications*, vol. 108, pp. 134–142, 2018.
- [6] Z. Li, H. Fang, M. Huang, Y. Wei, and L. Zhang, "Data-driven bearing fault identification using improved hidden Markov model and self-organizing map," *Computers & Industrial Engineering*, vol. 116, pp. 37–46, 2018.
- [7] K. Yu, T. R. Lin, H. Ma, X. Li, and X. Li, "A multi-stage semi-supervised learning approach for intelligent fault diagnosis of rolling bearing using data augmentation and metric learning," *Mechanical Systems and Signal Processing*, vol. 146, Article ID 202143, 2021.
- [8] X. Wen, G. Lu, J. Liu, and P. Yan, "Graph modeling of singular values for early fault detection and diagnosis of rolling element bearings," *Mechanical Systems and Signal Processing*, vol. 145, Article ID 106956, 2020.
- [9] X. Tang, Q. He, X. Gu, C. Li, H. Zhang, and J. Lu, "A novel bearing fault diagnosis method based on GL-mRMR-SVM," *Processes*, vol. 8, no. 7, p. 784, 2020.
- [10] T. Tang, L. Bo, X. Liu, B. Sun, and D. Wei, "Variable predictive model class discrimination using novel predictive models and adaptive feature selection for bearing fault identification," *Journal of Sound and Vibration*, vol. 425, pp. 137–148, 2018.
- [11] G. Xu, D. Hou, H. Qi, and L. Bo, "High-speed train wheel set bearing fault diagnosis and prognostics: a new prognostic model based on extendable useful life," *Mechanical Systems and Signal Processing*, vol. 146, Article ID 202150, 2021.
- [12] S. Zhang, S. Zhang, B. Wang, and T. G. Habetsler, "Deep learning algorithms for bearing fault diagnostics-A comprehensive review," *IEEE Access*, vol. 8, Article ID 29881, 2020.
- [13] A. Duan, L. Guo, H. Gao, X. Wu, and X. Dong, "Deep focus parallel convolutional neural network for imbalanced classification of machinery fault diagnostics," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 11, pp. 8680–8689, 2020.
- [14] K. N. Ravikumar, A. Yadav, H. Kumar, K. V. Gangadharan, and A. V. Narasimhadhan, "Gearbox fault diagnosis based on Multi-Scale deep residual learning and stacked LSTM model," *Measurement*, vol. 186, Article ID 110099, 2021.
- [15] L. Wen, X. Li, and L. Gao, "A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 12, Article ID 12900, 2021.
- [16] Z. Pan, Z. Meng, Z. Chen, W. Gao, and Y. Shi, "A two-stage method based on extreme learning machine for predicting the remaining useful life of rolling-element bearings," *Mechanical Systems and Signal Processing*, vol. 144, Article ID 106899, 2020.
- [17] N. Zeng, H. Zhang, W. Liu, J. Liang, and F. E. Alsaadi, "A switching delayed PSO optimized extreme learning machine for short-term load forecasting," *Neurocomputing*, vol. 240, pp. 175–182, 2017.
- [18] Z. Jiang, J. Yin, W. Gui, and C. Yang, "Prediction for blast furnace silicon content in hot metal based on composite differential evolution algorithm and extreme learning machine," *Control Theory & Applications*, vol. 33, no. 8, pp. 1089–1095, 2016.
- [19] O. Kadri and L. Mouss, "Identification and detection of the process fault in a cement rotary kiln by extreme learning machine and ant colony optimization," *Academic Journal of Manufacturing Engineering*, vol. 15, no. 2, pp. 43–50, 2017.
- [20] X. Wang, S. Si, and Y. Li, "Multiscale diversity entropy: a novel dynamical measure for fault diagnosis of rotating machinery," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5419–5429, 2021.
- [21] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.
- [22] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.
- [23] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.
- [24] G.-G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.
- [25] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.
- [26] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionics Engineering*, vol. 18, no. 3, pp. 674–710, 2021.
- [27] A. Faramarzi, M. Heidarnejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, 2020.
- [28] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [29] K. Zervoudakis and S. Tsafarakis, "A mayfly optimization algorithm," *Computers & Industrial Engineering*, vol. 145, Article ID 106559, 2020.
- [30] L. Abualigah, M. A. Elaziz, P. Sumari, Z. W. Geem, and A. H. Gandomi, "Reptile search algorithm (RSA): a nature-inspired meta-heuristic optimizer," *Expert Systems with Applications*, vol. 191, Article ID 116158, 2022.
- [31] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: new metaheuristic algorithm for solving optimization problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84–110, 2022.
- [32] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [33] T. Spoljaric, C. Lusetic, and V. Simovic, "Optimization of PID controlling in AVR system by using ant lion optimizer algorithm," in *Proceedings of the 41st International Convention on Information and Communication Technology*, pp. 1522–1526, Electronics and Microelectronics, Opatija, Croatia, May 2018.
- [34] M. Raju, L. C. Saikia, and N. Sinha, "Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller," *International Journal of Electrical Power & Energy Systems*, vol. 80, pp. 52–63, 2016.
- [35] A. Rout, G. B. Mahanta, B. Gunji, B. B. V. L. Deepak, and B. B. Biswal, "Kinematic and dynamic optimal trajectory planning of industrial robot using multi-objective ant lion optimizer," *Advances in Mechanical Engineering*, vol. 2020, pp. 1475–1486, 2020.
- [36] A. Kaveh and Y. Vazirinia, "Smart-home electrical energy scheduling system Smart-home electrical energy scheduling

- system using multi-objective ant lion optimizer and evidential reasoning,” *Scientia Iranica*, vol. 27, pp. 177–201, 2020.
- [37] R. Mei, M. Sulaiman, and Z. Mustaffa, “Ant lion optimizer for optimal reactive power dispatch solution,” *Journal of Electrical Systems*, vol. 2015, no. 3, pp. 68–74, 2015.
- [38] J. Junior, M. Nunes, M. Nascimento, C. Freitas, and N. Moraes, “Ant lion optimizer applied to economic emission load dispatch problems turning off the engines,” in *Proceedings of the 13th IEEE International Conference on Industry Applications*, pp. 829–836, IEEE, Sao Paulo, Brazil, November 2018.
- [39] T. George and V. Ganesan, “An effective technique for tuning the time delay system with PID controller-ant lion optimizer algorithm with ANN technique,” *International Journal of COMADEM*, vol. 23, no. 1, pp. 39–48, 2020.
- [40] R. Mei, M. Sulaiman, H. Daniyal, and Z. Mustaffa, “Application of moth-flame optimizer and ant lion optimizer to solve optimal reactive power dispatch problems,” *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, pp. 1-2, 2018.
- [41] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [42] X. Yue and H. Zhang, “A novel industrial image contrast enhancement technique based on an improved ant lion optimizer,” *Arabian Journal for Science and Engineering*, vol. 46, no. 4, pp. 3235–3246, 2020.
- [43] H. Kilic, U. Yuzgec, and C. Karakuzu, “A novel improved antlion optimizer algorithm and its comparative performance,” *Neural Computing & Applications*, vol. 32, no. 8, pp. 3803–3824, 2018.
- [44] M. W. Guo, J. S. Wang, L. F. Zhu, S. S. Guo, and W. Xie, “Improved ant lion optimizer based on spiral complex path searching patterns,” *IEEE Access*, vol. 8, Article ID 22126, 2020.
- [45] A. Tharwat and A. E. Hassanien, “Chaotic antlion algorithm for parameter optimization of support vector machine,” *Applied Intelligence*, vol. 48, no. 3, pp. 670–686, 2018.
- [46] E. Emary and H. M. Zawbaa, “Feature selection via Lèvy Antlion optimization,” *Pattern Analysis & Applications*, vol. 22, no. 3, pp. 857–876, 2019.
- [47] P. Yao and H. Wang, “Dynamic adaptive ant lion optimizer applied to route planning for unmanned aerial vehicle,” *Soft Computing*, vol. 21, no. 18, pp. 5475–5488, 2017.
- [48] K. NavnathDattatraya and K. R. Rao, “Maximising network lifetime and energy efficiency of wireless sensor network using group search ant lion with Levy flight,” *IET Communications*, vol. 14, no. 6, pp. 914–922, 2020.
- [49] P. Larrañaga and J. A. Lozano, “Estimation of distribution algorithms,” *Genetic Algorithms & Evolutionary Computation*, vol. 64, no. 5, pp. 454–468, 2001.
- [50] X. Wang, H. Zhao, T. Han, H. Zhou, and C. Li, “A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem,” *Applied Soft Computing*, vol. 78, pp. 240–260, 2019.
- [51] R. Pérez-Rodríguez and A. Hernández-Aguirre, “A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows,” *Computers & Industrial Engineering*, vol. 130, pp. 75–96, 2019.
- [52] L. Padierna, M. Carpio, A. Rojas, H. Paga, R. Baltazar, and H. Fraire, *Hyper-parameter Tuning for Support Vector Machines by Estimation of Distribution Algorithms*, Springer, Berlin, Germany, 2017.
- [53] J. Fan, “OPE-HCA: an optimal probabilistic estimation approach for hierarchical clustering algorithm,” *Neural Computing & Applications*, vol. 31, no. 7, pp. 2095–2105, 2019.
- [54] Q. Yang, W.-N. Chen, Y. Li, C. L. P. Chen, X.-M. Xu, and J. Zhang, “Multimodal estimation of distribution algorithms,” *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 636–650, 2017.
- [55] M. Ghaemi, Z. Zabihinpour, and Y. Asgari, “Computer simulation study of the Levy flight process,” *Physica A: Statistical Mechanics and Its Applications*, vol. 388, no. 8, pp. 1509–1514, 2009.
- [56] R. Rajabioun, “Cuckoo optimization algorithm,” *Applied Soft Computing*, vol. 11, no. 8, pp. 5508–5518, 2011.
- [57] X. S. Yang and S. Deb, “Engineering optimisation by cuckoo search,” *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.
- [58] H. Uğuz and H. Uguz, “A novel particle swarm optimization algorithm with Levy flight,” *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.
- [59] X. Yue, H. Zhang, and H. Yu, “A hybrid grasshopper optimization algorithm with invasive weed for global optimization,” *IEEE Access*, vol. 8, pp. 5928–5960, 2020.
- [60] J. Liu, Y. Huo, and Y. Li, “Preferred strategy based self-adaptive ant lion optimization algorithm,” *Pattern Recognition and Artificial Intelligence*, vol. 33, no. 2, pp. 121–132, 2020, in Chinese.
- [61] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, “Extreme learning machine: theory and applications,” *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.
- [62] “Case Western Reserve University Bearing Data Center Website,” 2020, <https://www.eecs.cwru.edu/laboratory/bearing/>.
- [63] X. An and H. Zeng, “Fault diagnosis method for spherical roller bearing of wind turbine based on variational mode decomposition and singular value decomposition,” *Journal of Vibroengineering*, vol. 18, no. 6, pp. 3548–3556, 2016.
- [64] Z. Zhao, Q. Xu, and M. Jia, “Improved shuffled frog leaping algorithm-based BP neural network and its application in bearing early fault diagnosis,” *Neural Computing & Applications*, vol. 27, no. 2, pp. 375–385, 2016.
- [65] J. S. Smith, “The local mean decomposition and its application to EEG perception data,” *Journal of The Royal Society Interface*, vol. 2, no. 5, pp. 443–454, 2005.
- [66] G. Dong and J. Pei, *Sequence Data Mining*, Springer-Verlag, New York, USA, 2007.
- [67] Y. Li, Y. Yang, X. Wang, B. Liu, and X. Liang, “Early fault diagnosis of rolling bearings based on hierarchical symbol dynamic entropy and binary tree support vector machine,” *Journal of Sound and Vibration*, vol. 428, pp. 72–86, 2018.
- [68] B. Battogtokh, M. Mojirsheibani, and J. Malley, “The optimal crowd learning machine,” *BioData Mining*, vol. 10, no. 1, pp. 16–12, 2017.