

## Research Article

# Semisupervised SVM by Hybrid Whale Optimization Algorithm and Its Application in Oil Layer Recognition

Yong-ke Pan , Ke-wen Xia , Wen-jia Niu, and Zi-ping He

*School of Electronics and Information Engineering, Hebei University of Technology, Tianjin 300401, China*

Correspondence should be addressed to Ke-wen Xia; [kwxia@hebut.edu.cn](mailto:kwxia@hebut.edu.cn)

Received 4 August 2020; Revised 14 January 2021; Accepted 20 January 2021; Published 8 February 2021

Academic Editor: David Bigaud

Copyright © 2021 Yong-ke Pan 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.

In many fields, such as oil logging, it is expensive to obtain labeled data, and a large amount of inexpensive unlabeled data are not used. Therefore, it is necessary to use semisupervised learning to obtain accurate classification with limited labeled data and many unlabeled data. The semisupervised support vector machine (S3VM) is the most useful method in semisupervised learning. Nevertheless, S3VM model performance will degrade when the sample number of categories is not even or have lots of unlabeled samples. Thus, a new semisupervised SVM by hybrid whale optimization algorithm (HWOA-S3VM) is proposed in this paper. Firstly, a tradeoff control parameter is added in S3VM to deal with an uneven sample of category which can cause S3VM to degrade. Then, a hybrid whale optimization algorithm (HWOA) is used to optimize the model parameters of S3VM to increase the classification accuracy. For HWOA improvement, an opposition-based cubic mapping is used to initialize the WOA population to improve the convergence speed, and the catfish effect is used to help WOA jump out of the local optimum and obtain the global optimization ability. In the experiments, firstly, the HWOA is tested by 12 classic benchmark functions of CEC2005 and four functions of CEC2014 compared with the other five algorithms. Then, six UCI datasets are used to test the performance of HWOA-S3VM and compared with the other four algorithms. Finally, we applied HWOA-S3VM to perform oil layer recognition of three oil well datasets. These experimental results show that (1) HWOA has a higher convergence speed and better global searchability than other algorithms. (2) HWOA-S3VM model has higher classification accuracy on UCI datasets than other algorithms when combined, labeled, and unlabeled data are used as the training dataset. (3) The recognition accuracy and speed of the HWOA-S3VM model are superior to the other four algorithms when applied in oil layer recognition.

## 1. Introduction

Semisupervised learning (SSL) [1] is between supervised learning and unsupervised learning, which is taking advantage of both learning methods while using labeled samples to learn and those unmarked samples. Because there is hidden a large amount of available information in the unlabeled data used to assist the classifier's learning, the classifier's performance is better. The focus of semisupervised learning research is how to combine these labeled and unlabeled samples organically by using some distribution hypotheses or intrinsic connections between samples through speculating the category labels of unlabeled samples based on available category label sample and domain knowledge. The unlabeled samples can then be converted

into labeled samples, merging into labeled in the data, and expand the available training datasets to train a good classifier. Obviously, semisupervised learning can significantly reduce the demand for labeled data and use some structural knowledge of unlabeled samples. Therefore, semisupervised learning has become a new research direction in many fields, for example, in oil logging, the cost of obtaining labeled data is very expensive, and a large amount of inexpensive unlabeled data are not used. So, it has important application value to research semisupervised learning to obtain accurate oil layer recognition results.

The current semisupervised learning methods generally have four main paradigms [2], such as a method based on the generative model, a semisupervised SVM method (S3VM) [3], a graph-based method, and a divergent-based method.

The method based on the generative model assumes that the generative model generates labeled data and the unlabeled data, which is associated with the learning target through model parameters. In general, the model parameters are estimated by the maximum likelihood using the EM algorithm [4]. The semisupervised SVM method maximizes the interval (margin) of all training data (both labeled and unlabeled) by adjusting the tag assignment of the SVM's hyperplane and unlabeled data. The graph-based approach [5] considers the association between data samples, mapping the dataset to a graph, and then "propagating" the marked information on the graph. The divergent-based method [6] uses unlabeled data by using multiple learners and the unlabeled data using a learning process multilearning platform for information interaction.

S3VM method is widely used among the above semisupervised learning methods. The semisupervised SVM applies to text categorization. Later, the researchers proposed a series of techniques to solve the semisupervised support SVM optimization function nonconvex problem. The main research methods [7] are gradient descent, convex-concave procedure, deterministic annealing, continuation techniques, and semidefinite programming. The above-mentioned algorithms can be divided into two categories according to the different objects of the parameters. One is a semisupervised support vector machine based on the combination. The other is a semisupervised support vector machine with a continuous method [8]. Nevertheless, the performance of these S3VM models will degrade when the sample number of categories is not even or when there are lots of unlabeled samples. Thus, the S3VM model needs to be modified to deal with an uneven sample of category which can cause S3VM to degrade. In addition, the parameters of S3VM model need to be optimized to further increase the classification accuracy and using intelligent optimization algorithm to optimize the model parameters is a research hotspot.

Intelligent optimization algorithms are generally motivated by specific inspirations such as biological evolutionary mechanisms, hunting patterns, or physical laws in nature, which construct specific theoretical models through simulation and extrapolation to achieve their optimal performance. Based on the different heuristic mechanisms, intelligent optimization algorithms are classified into three categories: evolutionary-based, swarm behavior-based, and physical property-based. The evolutionary-based algorithm mainly includes genetic algorithms (GA) that simulate genetic processes [9] and evolutionary planning of evolutionary processes (evolutionary programming (EP)) [10]. Swarm behavior-based algorithm includes particle swarm optimization (PSO) mimicking bird flock predation [11] and artificial bee colony algorithm (ABC) for deducing honey harvesting behavior of bee colonies [12]. The naked mole-rat algorithm (NMR) [13] simulates the mating patterns of NMRs, etc. Physical properties-based algorithm includes the simulated annealing algorithm derived from the solid annealing principle (SA) [14] and gravitational search algorithm (GSA) [15] which is based on the law of gravity and Newton's second law. In addition, intelligent optimization

algorithms based on new heuristic mechanisms have been continuously proposed, such as the fruit fly optimization algorithm (FOA) that simulates the group foraging behavior of fruit flies [16] and the collision body optimization algorithm (CBO) that simulates the mechanism of object collision velocity change [17].

Whale optimization algorithm (WOA) [18] is a newly proposed intelligent optimization algorithm proposed in 2016, which has the advantages of simple principle, few parameter settings, and strong optimization performance. The WOA has been successfully used for optimal capacitor siting [19], carbon dioxide emission prediction [20], clinical medicine [21], image segmentation [22], and wind speed forecasting by optimized SVR [23]. It was also used for the optimization of the node and energy consumption of wireless sensor networks [24, 25], support vector machines and neural networks optimization [26], and other practical problems. The good performance in engineering, medicine, and economics shows that the algorithm has broad application prospects. The basic WOA has the advantage of fewer parameters and better searchability than other algorithms, but it is still easy to fall into local optimum and slow convergence [27, 28]. Therefore, many scholars have proposed some improved algorithms; Kaveh and Ghazaan [29] proposed enhanced WOA, which introduces a random amount in the position update later in the algorithm. Oliva et al. [30] proposed a chaotic WOA based on the chaotic map to adapt the internal parameters of WOA. Hussien et al. [31] offered binary WOA by using two transfer functions to convert the algorithm's search space into binary. Salgotra et al. [32] proposed an improved WOA based on opposition-based learning, reinitializing the worst particle and decreasing parameters. The result shows the algorithm is competitive compared with WOA. Mohammed et al. [33] combined bat and WOA algorithm; the test result shows the WOA-BAT algorithm is competitive than WOA in multimodal functions. However, these algorithms need to be further improved in convergence speed and accuracy. Most researchers have not focused on population initialization to improve WOA. In literature [32], they have achieved good results in population initialization with opposition-based learning. In literature [31], they have achieved good results in population evolution with chaotic-mapping for WOA, so we try to combine opposition-based learning and chaotic-mapping for population initialization to enhance the WOA population's local optimization ability. The catfish effect [34] proved to be an excellent strategy to improve the global optimization ability of PSO algorithm, so we introduce catfish effect to WOA to enhance the global optimization ability.

In this paper, we propose an improved S3VM, firstly, adding a tradeoff control parameter in S3VM to deal with an uneven sample of category which can cause S3VM to degrade. Then, using a hybrid whale optimization algorithm (HWOA) to optimize the model parameters of S3VM, the HWOA is mainly improved in two aspects; on the one hand, the chaotic population method is used to initialize the whale population to improve the algorithm's local optimization ability; on the other hand, the population is improved by the

catfish effect to improve the algorithm's overall optimization ability. Finally, the new S3VM model based on the improved WOA algorithm is applied in oil layer recognition.

The rest of this paper is organized as follows. The basic semisupervised support vector machine algorithm and its limitations and improvement are briefly introduced in Section 2. Section 3.1 presents the basic whale optimization algorithm (WOA). Section 3.2 presents WOA limitations and proposed hybrid WOA. In Section 3, a set of benchmarks are employed to validate the algorithm's performance by comparing the results with WOA and other algorithms. Section 4 presents the improved S3VM model's detail combined with HWOA and runs the experiment of 5 UCI datasets compared to other algorithms. Section 5 presents the oil layer recognition system's detail and how to use improved S3VM model combined with HWOA to do the oil layer recognition.

## 2. The Semisupervised Learning SVM and Its Improvement

*2.1. Principle of Semisupervised SVM.* The SVM in supervised learning tries to find a dividing hyperplane so that the interval between the support vectors on both sides is the largest, that is, the "maximum division interval" idea. For semisupervised learning, the S3VM considers that the

hyperplane needs to pass through low data density areas. Transductive support vector machine (TSVM) [35] is the most famous representative of semisupervised support vector machines. The core idea is to find suitable mark assignments for unlabeled samples so that the interval after the hyperplane division is maximized. TSVM uses a local search strategy to solve iteratively. Firstly, training an initial SVM using a labeled sample set and then using the learner to mark unlabeled samples so that all samples are tagged and based on these labeled samples retrain the SVM and then look for error-prone samples to adjust continually.

Like the standard SVM, TSVM is also a learning method for the two-class problem. TSVM attempts to consider various possible label assignments for unlabeled samples, like attempting to treat each unlabeled sample as a positive or negative example, and then in all of these results, seeking a dividing hyperplane that maximizes the spacing on all samples (including labeled samples and unlabeled samples). Once the divisional hyperplane is determined, the unlabeled sample's final marker assignment is its prediction [36]. There are labeled data  $D_l = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$  and unlabeled data  $D_u = \{(x_{l+1}, x_{l+2}, \dots, x_{l+u})\}$ , and  $y_i \in \{-1, +1\}$ ,  $l \leq u$ ,  $l + u = m$ . The goal of TSVM is to predict the label  $\hat{y}$  of  $D_u$ ,  $\hat{y} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u})$ ,  $\hat{y}_i \in \{-1, +1\}$ . The entire algorithm flow is as follows:

$$\begin{aligned} \min_{\omega, b, \hat{y}, \xi} \quad & \frac{1}{2} \|\omega\|_2^2 + C_l \sum_{i=1}^l \xi_i + C_u \sum_{i=l+1}^m \xi_i \\ \text{s.t.} \quad & y_i (\omega^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, l \\ & \hat{y}_i (\omega^T x_i + b) \geq 1 - \xi_i, \quad i = l + 1, l + 2, \dots, m \\ & \xi_i \geq 0, \quad i = 1, 2, \dots, m, \end{aligned} \quad (1)$$

where  $(\omega, b)$  is determined a divide plane,  $\xi_i$  is the slack variable,  $C_l$  and  $C_u$  are the tradeoff control parameters which are used in equilibrium model complexity,  $\xi_i$  ( $i = 1, 2, \dots, l$ ) is the labeled sample, and  $\xi_i$  ( $i = l + 1, l + 2, \dots, m$ ) is the unlabeled sample.

*2.2. Problem Statement of TSVM and Its Improvement.* The TSVM uses a local search to find an approximate solution of the equation iteratively. Specifically, it first learns an SVM using labeled samples while it ignores the terms and constraints on  $D_u$  and  $y$  in equation (1). The unlabeled data are then labeled assignment with this SVM. The result of the SVM prediction is assigned to the unlabeled sample as a "pseudolabel." At this point,  $y$  becomes known and substituting it into the equation yields a standard SVM problem, so that a new partitioned hyperplane and

relaxation vector can be solved. Note that the pseudomarker of the unlabeled sample at this time is likely to be inaccurate, so  $C_u$  should be set to be smaller than  $C_l$  which makes the labeled sample work more. Next, TSVM finds two samples whose tags are assigned to be heterogeneous and are likely to be erroneous and then they swap their tags and solve the updated partitioned hyperplane and relaxation vectors based on equation again and then find out two tags are assigned to an unlabeled sample that is heterogeneous and is likely to be in error. After the label assignment adjustment is completed,  $C_u$  is gradually increased to increase the influence of the unlabeled sample on the optimization target, and the next round of label assignment adjustment is performed until  $C_u = C_l$ .

As the unevenness of category sample will disturb the training process of S3VM, in this paper, it will split  $C_u$  into  $C_u^+$  and  $C_u^-$ ,  $C_u^+ = (u_-/u_+)C_u^-$ ,  $u^+$ , and  $u^-$  is the sample

number based on pseudolabels as positive and negative examples.

In the process of marking adjustment, it can be considered that the samples most likely to be misassigned can be adjusted each time, that is, the samples with the largest slack

$$\begin{aligned} \min & \frac{1}{2} \|w\|_2^2 + C_l \sum_{i \in D_l} \xi_i + C_u^+ \sum_{i \in D_u \cap \hat{y}_i = 1} \xi_i + C_u^- \sum_{i \in D_u \cap \hat{y}_i = -1} \xi_i \\ & \forall_{i=1}^l: y_i (\omega \cdot x_i + b) \geq 1 - \xi_i \\ & \hat{y}_i (w^T x_i + b) \geq 1 - \zeta_j, i = l + 1 \dots u - i \in D_u \cap \hat{y}_i = 1 \\ & \hat{y}_j (w^T x_i + b) \geq 1 - \zeta_n, i = l + 1 \dots u - i \in D_u \cap \hat{y}_j = -1, \quad \forall_{i=1}^l: \xi_i \geq 0, \forall_{j=1}^k: \xi_j^* \geq 0, \end{aligned} \quad (2)$$

where  $C_l$ ,  $C_u^+$ , and  $C_u^-$  are tradeoff control parameters and  $\xi$  is the slack variable.

The kernel function of S3VM is the radial basis kernel function (RBF). The equation of RBF is  $K(x, x_i) = \exp(-\gamma \|x - x_i\|^2)$ , where  $\gamma$  denotes kernel parameters and denotes regularization parameters.

### 3. The Whale Optimization Algorithm (WOA) and Its Improvement

**3.1. Whale Optimization Algorithm.** According to the hunting behavior of the humpback whale group, the WOA algorithm abstracts three behaviors including prey, bubble-net attack, and random search.

**3.1.1. Surrounded by Prey.** Each whale represents an individual, and each individual represents a solution in the location of the search space. The whale can identify the prey's position and surround the prey by echolocation. The updating equation of whale position is as follows:

$$X^{t+1} = X_{g_{best}}^t - A \cdot |C \cdot X_{g_{best}}^t - X^t|, \quad (3)$$

where  $t$  is the current number of iterations,  $X_i$  is the current position vector;  $X_{g_{best}}^t = (X_{g_{best}1}^t, X_{g_{best}2}^t, \dots, X_{g_{best}D}^t)$  is the current best position vector,  $D$  is the vector dimension,  $A \cdot |C \cdot X_{g_{best}}^t - X^t|$  are surround steps, and  $A$  and  $C$  are the coefficient vectors.

The coefficient vectors  $A$  and  $C$  are defined as follows:

$$\begin{aligned} A &= 2a \cdot \text{rand}_1 - a, \\ C &= 2 \cdot \text{rand}_2, \end{aligned} \quad (4)$$

where  $\text{rand}_1$  and  $\text{rand}_2$  are random numbers in the range of  $[0, 1]$  by using uniform distribution and  $a$  is the convergence factor.

The convergence factor  $a$  will decrease linearly from 2 to 0 while iterations increase, as follows:

$$a = 2 - \frac{2t}{t_{\max}}, \quad (5)$$

variable and greater than 1 in the positive and negative marking samples can be changed by marking, which can reduce the number of iterations.

So, equation (1) can turn into

where  $t_{\max}$  is the maximum number of iterations.

**3.1.2. Bubble-Net Attack.** As the bubble-net foraging behavior shows the whales attack the prey by spiraling and bunching them even tighter, so shrink-wrapping mechanism and spiral updating location method are designed to describe the bubble-net foraging behavior mathematically.

Shrink-wrapping mechanism is achieved by the decreasing of convergence factor which has been shown in equation (5); meanwhile, the wave range of coefficient vector  $A$  will be decreased by convergence factor  $a$ . This means when the convergence factor  $a$  is decreased from 2 to 0 within iteration, the wave range of coefficient vector  $A$  is  $[-a, a]$ . When the coefficient vector  $A$  is a random number from  $[-1, 1]$ , the whale, at time  $t + 1$ , will be in any location between the location at time  $t$  and the global optimum location at time  $t + 1$ , which means the whale always swims within the shrink encirclement. Spiral updating location method means the whales move spirally to prey; the mathematical model is shown as follows:

$$X^{t+1} = X_{g_{best}}^t + D \cdot e^{bl} \cdot \cos(2\pi l), \quad (6)$$

where  $D = |X_{g_{best}}^t - X^t|$ ,  $D$  means the distance between the whales and the global optimum individual,  $b$  is a constant to limit logarithmic spiral shape, and  $l$  is a random number in the range of  $[-1, 1]$ .

The whales swim in the prey shrink range along a spiral path. In order to simulate this synchronous behavior, during the optimization process, assume the probability of selected shrink-wrapping mechanism and the spiral update location is assumed to be 0.5; synchronous behavior mathematical model is

$$X^{t+1} = \begin{cases} X_{g_{best}}^t - A \cdot |C \cdot X_{g_{best}}^t - X^t|, & p < 0.5, \\ X_{g_{best}}^t + D \cdot e^{bl} \cdot \cos(2\pi l), & p \geq 0.5, \end{cases} \quad (7)$$

where  $p$  is a random number in the range of  $[0, 1]$  produced by uniform distribution.

**3.1.3. Random Search.** When a coefficient vector  $|A| > 1$ , it means the whales are swimming outside of the shrink encirclement; at the same time, each whale will search each other randomly according to its location; its mathematical model is

$$X^{t+1} = X_{\text{rand}}^t - A \cdot |C \cdot X_{\text{rand}}^t - X^t|, \quad (8)$$

where  $X_{\text{rand}}^t$  is an individual whale position vector, who is selected randomly.

### 3.2. A Hybrid Whale Optimization Algorithm (HWOA)

**3.2.1. Problem Statement of WOA.** As mentioned above, Section 3.1 shows the main parameters of WOA algorithm which are coefficient vectors  $A$  and  $C$ , whereby  $A$  is crucial to local development capability and global exploration capability of WOA algorithm. If  $|A| > 1$ , the whales will expand the search scope to find a better candidate solution; this is the global exploration ability of WOA algorithm. If  $|A| < 1$ , the whales will reduce the search scope; they will search accurately in a local area; this means WOA algorithm has local development capability. While parameter  $A$  depends on convergence factor  $a$ , a larger convergence factor will have a better global exploration ability to avoid the algorithm falling into local optimum situation; a smaller convergence factor will have a better local exploration ability to faster convergence speed of the algorithm [30].

The basic principle of WOA uses the random way to initial population, but this may cause the uneven distribution of population and the substandard diversity of the population. Most researchers did not focus on the population initialization of WOA, but literature [32] uses opposition-based learning for population initialization and gets good results. Luo and Shi [37] argue that higher initial population diversity can improve the heuristic search algorithm's search efficiency and that spreading the initial population in the search range as much as possible when the global optimal solution unknown can provide an excellent global search basis in the early stage of the algorithm. In [38], chaotic theory was combined with the whale algorithm which can improve the whale algorithm using several chaotic maps. In [39, 40], the swarm intelligence algorithms with improved population diversity using a chaotic map give good results. This is because the chaotic map has the characteristics of randomness, regularity, and boundedness. It is sensitive to initial values and can traverse all states in a certain range without repetition according to its own laws. So, we propose a strategy of WOA population initialization based on opposition-based learning and cubic mapping to improve the global search ability and convergence speed of WOA, which is called the opposition-based cubic mapping whale optimization algorithm (OCWOA).

Chuang et al. [34] introduced catfish effect to improve the searchability and jump out of the local optimum of PSO, and the results show that the catfish effect can improve the searchability of the algorithm. Similarly, we can also introduce catfish effect to improve the searchability of WOA.

So, we propose a WOA based on catfish effect, which is called CWOA in this paper.

Moreover, by combining OCWOA with CWOA, we further propose the hybrid whale optimization algorithm (HWOA) which is improved by opposition-based learning cubic mapping theory and catfish effect.

**3.2.2. The Opposition-Based Cubic Mapping Whale Optimization Algorithm (OCWOA).** This section presents opposition-based cubic mapping (OBCM) to initialize the whale population's position, strengthen the diversity of the population, and lay the foundation for global search. As the literature [39] shows, we choose cubic mapping as the function.

In order to show that cubic mapping generates a more uniform population than random generate, we use random generation and cubic mapping to generate 10,000 random numbers in the range  $-1$  to  $1$  to know the different distribution between cubic mapping and random generate. We used a simulation graph to show the sequence distribution comparison graph between cubic mapping generated populations and randomly generated populations in Figure 1.

Figure 1 shows that we divide the horizontal coordinate into deciles to determine the number of random numbers generated by the two methods in each interval. The horizontal coordinates represent the percentage interval ranging from  $-1$  to  $1$ , and the vertical coordinates represent the number of generations.

As seen in Figure 1, cubic mapping is more uniform than a random generation, which improves the initialization of the population and can effectively improve the global search ability of the population.

The mathematical expression of cubic mapping is shown as follows:

$$\begin{aligned} y(n+1) &= 4y(n)^3 - 3y(n), \\ -1 &\leq y(n) \leq 1, \quad n = 0, 1, 2, \dots \end{aligned} \quad (9)$$

There are five steps to generate an opposition-based cubic mapping population:

Step 1: generate a  $D$ -dimensional vector randomly according to  $M$  whales' population of  $D$ -dimensional space. As the first individual of whale population, which means  $Y = (y_1, y_2, \dots, y_d)$ ,  $y_i \in [-1, 1]$ ,  $1 \leq i \leq d$ , where  $Y$  is the individual of whale population,  $y_i$  is the dimension of  $Y$ , and the range of  $y_i$  is from  $-1$  to  $1$ .

Step 2: iterate  $M - 1$  times in every dimension of  $Y$  by equation (9), which will generate other  $M - 1$  whales' population and will generate initial population  $X = \{X_i, i = 1, 2, \dots, N\}$ , where  $X$  is the whole population of whale based on cubic mapping.

Step 3: mapping the generated chaotic variables to the solution of search space by equation (11):

$$X_{\text{id}} = L_d + (1 + y_{\text{id}}) \frac{U_d - L_d}{2}, \quad (10)$$

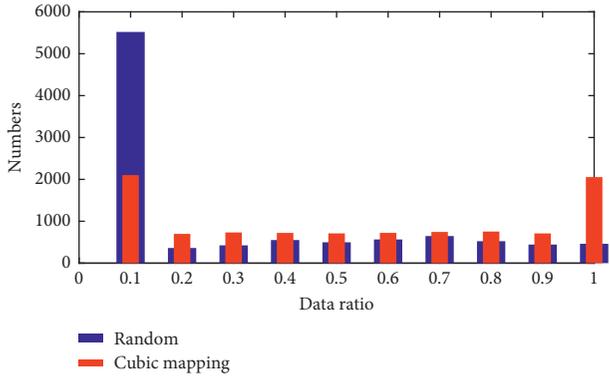


FIGURE 1: Sequence distribution of cubic mapping and random.

where  $U_d$  and  $L_d$  represent the upper and lower limits of  $D$ -dimension of search space, respectively,  $y_{id}$  is the  $D$ -dimension of  $i$ -th whale calculated by equation (9), and  $X_{id}$  is the  $D$ -dimension coordinate of the  $i$ -th whale.

Step 4: use opposition-based learning to generate a reverse population  $OX_{id}$  based on  $X_{id}$  of equation (11). The opposition-based learning equation of  $OX_{id}$  is shown as follows:

$$OX_{id} = U_d + L_d - X_{id}. \quad (11)$$

Step 5: merge the  $X$  and  $OX_{id}$  to generate a new population  $\{X \cup OX_{id}\}$ , and calculate the objective function values of the new population and rank them. Then, pick the best fitness of  $M$  individuals as the initial population.

The abovementioned algorithm is called the opposition-based cubic mapping whale optimization algorithm (OCWOA).

**3.2.3. WOA Based on Catfish Effect (CWOA).** The basic idea of catfish effect is to introduce catfish into sardine's population to eliminate the sardines with no vitality. Catfish effect supposes if the global optimal solution does not evolve in a specific algebra of the iteration process, in that case, it will initialize 60% of the worst individual solution particles (no active sardines) to restore group vitality. This strategy is to improve the global search capability of the algorithm, because in the later iteration stage of WOA algorithm, all of the whale individuals in the population aggregate to the optimal individual location. This leads to the loss of diversity within the population and premature convergence of the algorithm to a nonglobal optimal state, which will cause premature convergence. So, the WOA based on catfish effect is named CWOA. When the global optimal solution of WOA does not evolve in a certain time of iteration process, it will proceed to opposition-based cubic mapping mutation equations (9)–(11) as catfish effect. The flow chart of CWOA is shown in Figure 2.

According to the OCWOA and CWOA, here, we can get the hybrid whale optimization algorithm (HWOA), which is

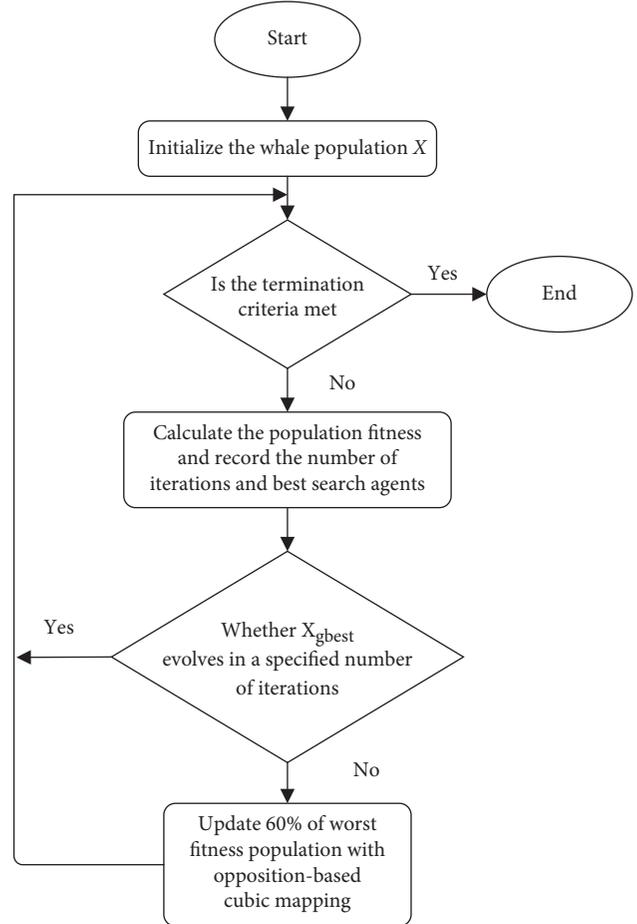


FIGURE 2: Flow chart of CWOA.

improved by opposition-based learning cubic mapping theory for population initialization and catfish effect for population update, and the flow chart of HWOA is shown in Figure 3.

So, the pseudocode of the hybrid WOA algorithm (HWOA) is shown in Algorithm 1.

### 3.3. Experiment and Result Analysis of HWOA

**3.3.1. Experiment Environment, Parameter Setting, and Test Suit.** In the paper, all of the experimental environment includes a 2.40 GHz Intel(R) Core TM i7-3630QM CPU with RAM of 8 GB in the Windows 7 system. Each of the experiments was repeated 30 times independently on MATLAB 2017.

In order to verify the effectiveness and generalization of the proposed algorithm, we compare the HWOA with WOA [18], OEWOA [32], WOA-BAT [33], OCWOA, and CWOA on 16 benchmark functions: F1–F12 from CEC2005 [41] and F13–F16 from CEC2014 [42], where F1–F4 are unimodal benchmark functions, F5–F8 are multimodal benchmark functions, and F9–F12 are fixed-dimensions multimodal benchmark functions. F13–F16 are composite functions. The results of function F1–F4 will show the exploitation and convergence ability of the algorithm, the results of function

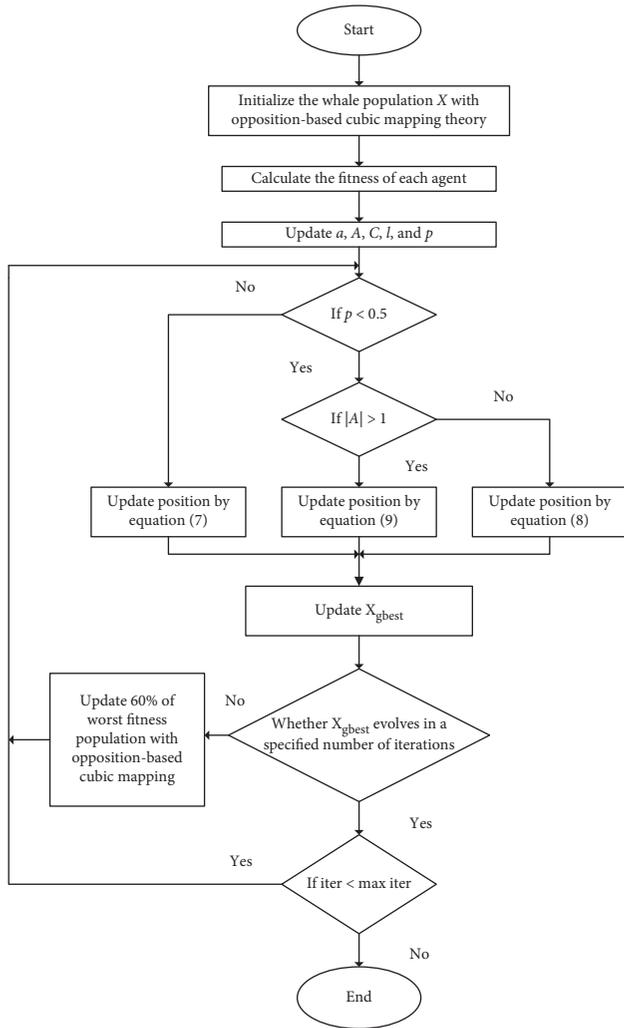


FIGURE 3: Flow chart of HWOA.

F5–F12 will reflect the exploitation capability of the algorithm, and function F13–F16 will test the algorithm to balance exploration and exploitation.

The detailed information of benchmark functions is shown in Table 1. Table 1 includes the function type, range of variation of optimization variables, optimal value  $f_{\min}$ , and design variables, Dim. The parameter settings of the five algorithms are shown in Table 2.

This experiment will compare the speed of convergence, and the optimization accuracy of six algorithms on the benchmark function at the same population size, number of iterations, and number of runs. So, the maximum iteration is set to 500, and each algorithm will run 30 times independently. We use the mean and standard deviation of the optimal target value as the basis for performance evaluation.

At first, we use different populations of 30 and 60 when dimension = 30 to test the performance of the six algorithms. In this part, we will find out whether different population sizes affect the test results. The results are shown in Tables 3 and 4.

Then, we use the different dimensions of 50 and 100 when the population size is 30. In this part, we will find out

whether different dimension sizes affect the test results. The results are shown in Tables 5 and 6.

**3.3.2. Analysis of Exploitation Capability When Population Size Changes.** In this section, it compares six algorithms in different situations when population changes and dimension = 30, and the fixed-dimension function (F9–F12) will stay the fixed-dimension. The result of the experiment will be discussed separately.

Table 3 shows the optimization results of 16 benchmark functions. Among the unimodal functions (F1–F4), HWOA can get the exact optimal solution of F1–F3, which proves that HWOA has profitable exploitation among six algorithms. For multimodal function (F5–F8), WOA-BAT obtains better optimal solution in F5 and F8. The exact optimal solutions of HWOA, WOA, OEWOA, CWOA, and OCWOA are obtained in F6. HWOA gets best mean and standard deviation of the optimal target value in F7. For the fixed-dimension function (F9–F12), HWOA outperforms the other five algorithms in F9, F11, and F12. WOA-BAT is good in F10. For composite function (F13–F16), HWOA is competitive with other algorithms. The results can infer that HWOA which is better than the other algorithms on different types of benchmark functions. Comparing CWOA, OCWOA, and WOA algorithm, F1 to F15 show that the mean and standard deviation of the optimal target value of OCWOA CWOA is much better than that of WOA. It means that the opposition-based cubic mapping strategy and catfish strategy to improve WOA are successful.

Table 4 shows the optimization results of 16 benchmark functions. Among the unimodal functions (F1–F4), HWOA is still competitive compared with other algorithms in F1–F3, but WOA-BAT is the only one that can jump out of local optimum in F4. For multimodal function (F5–F8), HWOA gives the best optimal value on F6 and F7. WOA-BAT reaches the better result in F5 and F8. For fixed-dimension function (F9–F12), HWOA outperforms on F11, but still has an advantage on the other three functions. For the composite function (F13–F16), HWOA is still competitive compared with another algorithm.

In summary, the most results of Tables 3 and 4 just have some minor differences. But as the population increases, the computational complexity grows. As the population changes from 30 to 60, the results of all algorithms have been improved a bit, and still, HWOA can get the best optimal value of 11 functions in Table 3 and 10 functions in Table 4. The performance of CWOA and OCWOA is still better than that of WOA in most functions, which shows that the proposed strategy to improve the WOA is a success. Overall, HWOA is better than the other five algorithms in different population sizes.

**3.3.3. Analysis of Exploitation Capability When Dimension Changes.** In this section, we compare six algorithms in different situations when the dimension changes and population size = 30. The fixed-dimension function (F9–F12) cannot do the experiment because the dimension is fixed. The result of the experiment will be discussed separately.

Input: number of search agents: 30

Maximum of iterations: 500

Output: best search agent:  $X_{gbest}$

Procedure:

- (1) Initialize the whale's population  $X_i$  ( $i = 1, 2, 3, \dots, n$ ) with the opposition-based cubic mapping in Section 3.2.1. Calculate the fitness of each search agent; select  $X_{gbest}$ .  
 $X_{gbest}$  is the best search agent
- (2) while ( $t < \text{maximum iterations}$ )
- (3) for each search agent
- (4) update  $a$ ,  $A$ ,  $C$ ,  $L$ , and  $p$ 
  - if  $1(p < 0.5)$
  - if  $2(|A| < 1)$
- (5) update the position of search agent using equation (3)
  - else  $2(|A| > 1)$
- (6) Select a random search agent ( $X_{rand}$ )
- (7) Update the position of search agent using equation (9)
  - end if 2
  - else  $1(p > 0.5)$
- (8) update the position with spiral equation (8)
  - end if 1
- (9) CHECK if any search agent goes beyond the search space and amend it. Calculate fitness of each search agent
- (10) Update  $X_{gbest}$  if there is a better solution and record  $X_{gbest}$  and its iteration.
  - If 3  $X_{gbest}$  did not evolve in 5 iterations, then ranking the search agent fitness forms best to worst, and initialize the position of 60% worst search agent with opposition-based cubic mapping in Section 3.2.1.
  - end if 3
  - end for
  - $t = t + 1$
  - end while
  - return  $X_{gbest}$

ALGORITHM 1: The hybrid WOA algorithm (HWOA). The pseudocode of HWOA.

Table 5 shows the optimization results of 16 benchmark functions when the dimension is 50.

From Table 5, HWOA provides best results for the functions F1, F2, and F3. For function F4, results of HWOA are still comparable while WOA-BAT gets better result. Results are quite comparable for function F5, but HWOA has a better mean value. For function F7, HWOA is better and provides good value of best solution and standard deviation. Only HWOA provides the exact solution for function F8. HWOA provides good results in F13, F14, and F15. In total, for this case, WOA is good for only one function, HWOA is good for eight functions, and OEWOA is good for three functions, WOA-BAT is good for three functions, OCWOA is good for four functions, and CWOA is good for three functions.

Table 6 shows the optimization results of 16 benchmark functions when the dimension is 100.

In Table 6, for functions F1 and F2, only HWOA is able to provide global solution, while other algorithms fall into at some local values. OEWOA and HWOA provide better results for function F3. The best solution obtained by WOA-BAT is better for function F4. For function F5, HWOA is considered as the best algorithm though the results. OEWOA outperforms other algorithms for function F6 and F7, but HWOA is still competitive compared with other four algorithms. For function F8, WOA-BAT provides better value. HWOA provides good results in F13, F14, and F15.

Thus, for the dimension size of 100, WOA is good for four functions, HWOA is good for eight functions, CWOA is good for five functions, OCWOA is good for four functions, WOA-BAT is good for two functions, and OEWOA is good for seven functions.

From Tables 5 and 6, it can be inferred that the complexity of the problem increases with the increase of dimension size. Due to this reason, the performance of all algorithms decreases. HWOA is good for eight functions in Tables 5 and 6. The result shows that HWOA has better performance over other algorithms.

**3.3.4. Analysis of Convergence Behavior.** The convergence curves of the CWOA, WOA, HWOA, OEWOA, WOA-BAT, and OCWOA are provided in Figure 4 to see the convergence speed of the algorithms.

The data of convergence graph is from Table 3.

From Figure 4, compared with standard WOA, CWOA, OCWOA, and OEWOA, WOA-BAT and HWOA have the fastest convergence speed in F1, F2, F3, F6, F7, F9, F10, F12, F13, F14, and F15, which means HWOA can effectively save the search time. The convergence curve of HWOA proves that the ability of HWOA to jump out of the local optimum is effectively enhanced compared with WOA. Therefore, the HWOA algorithm has a more vital ability to jump out of the local optimum than the other five algorithms.

TABLE 1: Description of 16 benchmark functions.

| Function type   | Dim | Range         | $f_{\min}$    |
|---|-----|---------------|---------------|
| $F_1(x) = \sum_{i=1}^n x_i^2$   | 30  | [-100, 100]   | 0             |
| $F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $   | 30  | [-100, 100]   | 0             |
| $F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$  | 30  | [-100, 100]   | 0             |
| $F_4(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$  | 30  | [-100, 100]   | 0             |
| $F_5(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$   | 30  | [-500, 500]   | -418.9829 × 5 |
| $F_6(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$  | 30  | [-5.12, 5.12] | 0             |
| $F_7(x) = -20 \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2}) - \exp((1/n) \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$  | 30  | [-32, 32]     | 0             |
| $F_8(x) = (1/10) \left\{ 10 \sin^2(3\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} + \sum_{i=1}^D u(x_i, 5, 100, 4)$  | 30  | [-50, 50]     | 0             |
| $F_9(x) = ((1/500) + \sum_{j=1}^{25} (1/j + \sum_{i=1}^2 (x_i - a_{ij})^6))^{-1}$   | 2   | [-65, 65]     | 1             |
| $F_{10}(x) = \sum_{i=1}^{11} [a_i - (x_1(b_i^2 + b_1 x_2)/b_i^2 + b_i x_3 + x_4)]^2$  | 4   | [-5, 5]       | 0.00030       |
| $F_{11}(x) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$  | 2   | [-5, 5]       | -1.0316       |
| $F_{12}(x) = (x_2 - (5.1/4\pi^2)x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/8\pi)\cos x_1) + 10$  | 2   | [-5, 5]       | 0.398         |
| F13 composition function 1 [43]<br>$N = 5,$<br>$\sigma = [10, 20, 30, 40, 50]$<br>$\lambda = [1, 1e - 6, 1e - 26, 1e - 6, 1e - 6]$<br>Bias = [0, 100, 200, 300, 400]<br>$g_1$ : rotated Rosenbrock's function $F'_4$<br>$g_2$ : high conditioned elliptic function $F'_1$<br>$g_3$ : rotated bent cigar function $F'_2$<br>$g_4$ : rotated discuss function $F'_3$<br>$g_5$ : high conditioned elliptic function $F'_1$ | D   | [-100,100]    | 2300          |
| F14 composition function 2 [43]<br>$N = 3,$<br>$\sigma = [20, 20, 20]$<br>$\lambda = [1, 1, 1]$<br>Bias = [0, 100, 200]<br>$g_1$ : Schwefel's function $F'_{10}$<br>$g_2$ : rotated Rastrigin's function $F'_9$<br>$g_3$ : rotated HGBat function $F'_{14}$   | D   | [-100,100]    | 2400          |
| F15 composition function 3 [43]<br>$N = 3,$<br>$\sigma = [10, 30, 50]$<br>$\lambda = [0.25, 1, 1e - 7]$<br>Bias = [0, 100, 200]<br>$g_1$ : rotated Schwefel's function $F'_{11}$<br>$g_2$ : rotated Rastrigin's function $F'_9$<br>$g_3$ : rotated high conditioned elliptic function $F'_1$  | D   | [-100,100]    | 2500          |
| F16 composition function 4 [43]<br>$N = 5,$<br>$\sigma = [10, 10, 10, 10, 10]$<br>$\lambda = [0.25, 1, e - 7, 2.5, 10]$<br>Bias = [0, 100, 200, 300, 400]<br>$g_1$ : rotated Schwefel's function $F'_{11}$<br>$g_2$ : rotated happy cat function $F'_9$<br>$g_3$ : rotated high conditioned elliptic function $F'_1$<br>$g_4$ : rotated Weierstrass function $F'_6$<br>$g_5$ : rotated Griewank's function $F'_7$       | D   | [-100, 100]   | 2600          |

TABLE 2: Parameter setting of six algorithms.

| Algorithm | Parameter settings   |
|-----------|--|
| WOA       | Number of iteration is 500, a decreases linearly from 2 to 0 |
| OEWOA     | Number of iteration is 500, a decreases linearly from 2 to 0 |
| CWOA      | Number of iteration is 500, a decreases linearly from 2 to 0 |
| OCWOA     | Number of iteration is 500, a decreases linearly from 2 to 0 |
| WOA-BAT   | Number of iteration is 500, a decreases linearly from 2 to 0 |
| HWOA      | Number of iteration is 500, a decreases linearly from 2 to 0 |

TABLE 3: Result of HWOA compared with other algorithms with population size 30.

| Function |      | WOA       | OEWOA      | WOA-BAT   | OCWOA     | CWOA       | HWOA      |
|----------|------|-----------|------------|-----------|-----------|------------|-----------|
| F1       | Mean | 6.981E-73 | 3.73E-172  | 1.14E-06  | 3.28E-89  | 0          | 0         |
|          | Std  | 3.66E-75  | 6.66E-125  | 3.98E-05  | 4.58E-79  | 0          | 0         |
| F2       | Mean | 6.475E-56 | 5.12E-287  | 0.004769  | 4.75E-166 | 8.88E-175  | 0         |
|          | Std  | 4.58E-55  | 3.24E-252  | 0.002587  | 3.14E-133 | 7.78E-155  | 0         |
| F3       | Mean | 5.71E+04  | 0          | 9.53E-06  | 1.55E-168 | 8.47E-300  | 0         |
|          | Std  | 3.52E+03  | 0          | 7.56E-05  | 1.57E-156 | 6.74E-289  | 0         |
| F4       | Mean | 28.0135   | 27.8847    | 1.19E+01  | 28.8066   | 27.7564    | 28.7891   |
|          | Std  | 0.4217    | 0.3547     | 1.12E+01  | 0.3352    | 0.3575     | 0.2231    |
| F5       | Mean | -12554.44 | -12568.45  | -12569.48 | -11821.96 | -12569.41  | -12568.62 |
|          | Std  | 695.12    | 1152.32    | 257.21    | 493.25    | 351.21     | 321.15    |
| F6       | Mean | 0         | 0          | 4.28E-04  | 0         | 0          | 0         |
|          | Std  | 0         | 0          | 3.56E-03  | 0         | 0          | 0         |
| F7       | Mean | 7.99E-15  | 8.88E-16   | 0.001232  | 8.88E-16  | 8.88E-16   | 8.88E-16  |
|          | Std  | 6.98E-14  | 7.98E-15   | 0.1258    | 7.55E-15  | 7.43E-15   | 5.61E-15  |
| F8       | Mean | 0.1711    | 0.22211    | 1.073E-02 | 1.2747    | 0.007376   | 0.24186   |
|          | Std  | 0.0512    | 0.1547     | 2.63E-02  | 2.9631    | 0.01563    | 0.01423   |
| F9       | Mean | 2.1191    | 0.998      | 1.1251    | 2.588     | 1.2381     | 0.998     |
|          | Std  | 2.4981    | 1.57E-16   | 0.5812    | 1.5782    | 0.8932     | 2.3E-17   |
| F10      | Mean | 0.0012438 | 0.00061546 | 0.0003076 | 0.0003644 | 0.00032927 | 0.0003145 |
|          | Std  | 0.000987  | 0.000251   | 0.000157  | 0.000169  | 0.000188   | 0.000144  |
| F11      | Mean | -1.0316   | -1.0313    | -1.0316   | -1.0316   | -1.0316    | -1.0316   |
|          | Std  | 4.21E-07  | 5.87E-12   | 3.21E-13  | 2.23E-17  | 4.34E-11   | 1.24E-19  |
| F12      | Mean | 0.39789   | 0.39795    | 0.39789   | 0.39792   | 0.3984     | 0.39801   |
|          | Std  | 3.98E-05  | 0          | 0         | 1.25E-04  | 2.25E-06   | 2.21E-07  |
| F13      | Mean | 2767.1779 | 2500       | 2500.0302 | 2500      | 2786.6705  | 2500      |
|          | Std  | 0.6681    | 0.6681     | 0.4572    | 0.3581    | 0.1258     | 0.1158    |
| F14      | Mean | 2604.0435 | 2600       | 2600.0075 | 2600      | 2603.3545  | 2600      |
|          | Std  | 0.1234    | 0          | 0.0021    | 0         | 0.1523     | 0         |
| F15      | Mean | 2723.3227 | 2700       | 2700.0006 | 2700      | 2700       | 2700      |
|          | Std  | 1.731     | 0          | 1.86E-04  | 0         | 0          | 0         |
| F16      | Mean | 2701.3939 | 2798       | 2722      | 2755      | 2769       | 2706.4615 |
|          | Std  | 3.43E-01  | 7.13       | 8.32      | 5.21      | 9.32       | 3.53E-01  |

3.3.5. *Summary of Results.* The result of OCWOA shows that using the opposition-based cubic mapping strategy to initializing the population of WOA is success. The OCWOA has faster convergence speed and better searchability than original WOA.

The result of CWOA shows that using the catfish effect in WOA has effectively enhanced the convergence speed and searchability of WOA.

From the results of Section 3.3.2–3.3.4, the proposed version of CWOA and OCWOA is better than the original WOA. The result demonstrates the success of the improvements to WOA using catfish effect and opposition-based cubic mapping.

Although HWOA is the best algorithm compared to other five algorithms in the experiment part, however, HOWA does not get good results in multimodal benchmark

functions part. So, there will be more work to improve the performance of HOWA in multimodal benchmark functions. We could introduce the different exponentially decreasing  $\vec{a}$  to improve the exploration and exploitation of HWOA on multimodal benchmark functions.

#### 4. Improved S3VM Model Combines with Hybrid WOA

4.1. *HWOA-S3VM Model.* In Section 2, we improved S3VM with split tradeoff control parameter  $C_u$  into  $C_u^+$  and  $C_u^-$ ; it will increase the iteration time and classification accuracy of S3VM. In Section 3, we modified WOA with catfish effect and opposition-based cubic mapping. In this section, we will use HWOA hybridized with S3VM for classification.

TABLE 4: Result of HWOA compared with other algorithm with population size 60.

| Function |      | WOA       | OEWOA     | WOA-BAT     | OCWOA     | CWOA        | HWOA      |
|----------|------|-----------|-----------|-------------|-----------|-------------|-----------|
| F1       | Mean | 1.50E-93  | 3.73E-202 | 2.24E-08    | 3.28E-98  | 0           | 0         |
|          | Std  | 5.72E-83  | 7.44E-125 | 5.88E-06    | 6.78E-82  | 0           | 0         |
| F2       | Mean | 8.85E-57  | 6.22E-289 | 0.004872    | 1.75E-106 | 2.1006E-181 | 0         |
|          | Std  | 3.68E-57  | 4.24E-261 | 0.002587    | 3.14E-103 | 7.78E-155   | 0         |
| F3       | Mean | 2.27E+04  | 2.87E+04  | 1.051E-05   | 3.71E+04  | 6.91E-316   | 0         |
|          | Std  | 1.32E+03  | 1.02E+04  | 2.24E-05    | 1.22E+03  | 5.54E-310   | 0         |
| F4       | Mean | 27.0112   | 27.7841   | 11.35E+01   | 28.7789   | 27.7895     | 26.4571   |
|          | Std  | 0.4221    | 0.789     | 10.3598     | 0.3714    | 0.3447      | 0.2778    |
| F5       | Mean | -12569.38 | -12569.32 | -12569.4866 | -11921.66 | -12569.0193 | -12569.45 |
|          | Std  | 221.23    | 152.32    | 124.31      | 313.75    | 351.36      | 321.24    |
| F6       | Mean | 0         | 0         | 2.17E-04    | 0         | 0           | 0         |
|          | Std  | 0         | 0         | 4.33E-03    | 0         | 0           | 0         |
| F7       | Mean | 7.99E-15  | 8.88E-16  | 0.005772    | 8.88E-16  | 8.88E-16    | 8.88E-16  |
|          | Std  | 6.72E-14  | 7.88E-15  | 4.72E-02    | 6.32E-15  | 1.22E-15    | 6.55E-15  |
| F8       | Mean | 0.05127   | 0.02041   | 1.99E-02    | 1.0178    | 0.012393    | 0.07941   |
|          | Std  | 0.003212  | 0.01568   | 3.77E-03    | 0.9872    | 0.00763     | 0.01231   |
| F9       | Mean | 0.998     | 0.998     | 0.998       | 1.3245    | 0.998       | 0.998     |
|          | Std  | 1.22E-32  | 1.57E-16  | 1.87E-23    | 1.5782    | 0.8932      | 3.2E-17   |
| F10      | Mean | 0.0004764 | 0.0003094 | 0.00030749  | 0.0003212 | 0.0003324   | 0.0003224 |
|          | Std  | 0.0002237 | 0.0001457 | 0.0001547   | 0.0001457 | 0.0001441   | 0.000122  |
| F11      | Mean | -1.0316   | -1.0316   | -1.0316     | -1.0316   | -1.0316     | -1.0316   |
|          | Std  | 2.31E-22  | 5.45E-18  | 3.77E-19    | 4.33E-21  | 1.25E-22    | 2.55E-23  |
| F12      | Mean | 0.39789   | 0.3979    | 0.39789     | 0.39792   | 0.3983      | 0.39801   |
|          | Std  | 3.88E-05  | 0         | 0           | 1.45E-04  | 2.78E-06    | 2.31E-07  |
| F13      | Mean | 2833.1779 | 2500      | 2500.0541   | 2500      | 2500        | 2500      |
|          | Std  | 33.221    | 0         | 0.3372      | 0         | 0           | 0         |
| F14      | Mean | 2600.3512 | 2600      | 2600.0075   | 2600      | 2600.3478   | 2600      |
|          | Std  | 0.1234    | 0         | 0.01221     | 0         | 0.0124      | 0         |
| F15      | Mean | 2700      | 2700      | 2700.0002   | 2700      | 2700        | 2700      |
|          | Std  | 0.9231    | 0         | 2.33E-04    | 0         | 0           | 0         |
| F16      | Mean | 2700.56   | 2800.52   | 2705.32     | 2755.23   | 2800        | 2755.11   |
|          | Std  | 2.13E-03  | 4.13E+01  | 6.13E+01    | 2.25E+00  | 6.13E+01    | 9.63E+00  |

TABLE 5: Result of HWOA compared with another algorithm with dimension 50.

| Function |      | WOA        | OEWOA       | WOA-BAT     | OCWOA      | CWOA        | HWOA      |
|----------|------|------------|-------------|-------------|------------|-------------|-----------|
| F1       | Mean | 1.2278E-80 | 2.79E-127   | 7.83E-06    | 8.6008E-54 | 1.9763E-323 | 0         |
|          | Std  | 4.63E-83   | 5.54E-115   | 4.33E-05    | 7.54E-51   | 2.48E-319   | 0         |
| F2       | Mean | 7.4197E-54 | 1.1564E-275 | 0.0044593   | 2.714E-57  | 1.1326E-177 | 0         |
|          | Std  | 4.79E-53   | 6.31E-267   | 0.002587    | 3.14E-53   | 7.78E-155   | 0         |
| F3       | Mean | 5.23E+04   | 0           | 8.18E-06    | 8.81E+04   | 3.34E-321   | 0         |
|          | Std  | 2.26E+03   | 0           | 6.34E-05    | 3.21E+03   | 6.58E-319   | 0         |
| F4       | Mean | 48.3528    | 48.342      | 19.46929    | 48.6378    | 48.5195     | 48.3739   |
|          | Std  | 0.3121     | 0.5782      | 18.4231     | 0.4521     | 0.4571      | 0.2425    |
| F5       | Mean | -19273.83  | -18485.0977 | -20881.1444 | -14998.407 | -20147.344  | -20865.71 |
|          | Std  | 159.54     | 163.33      | 135.21      | 211.52     | 354.45      | 225.25    |
| F6       | Mean | 0          | 0           | 0.00092049  | 0          | 0           | 0         |
|          | Std  | 0          | 0           | 5.24E-05    | 0          | 0           | 0         |
| F7       | Mean | 4.44E-15   | 8.88E-16    | 0.0013637   | 1.04E-13   | 8.88E-16    | 8.88E-16  |
|          | Std  | 5.77E-14   | 6.85E-15    | 4.86E-02    | 7.25E-15   | 2.43E-15    | 6.55E-16  |
| F8       | Mean | 0.77343    | 0.07841     | 5.66E-02    | 1.3178     | 0.09185     | 0.0875    |
|          | Std  | 0.004512   | 0.01787     | 1.71E-02    | 0.8971     | 0.001457    | 0.0254    |
| F13      | Mean | 2911.1798  | 2500        | 2500.08     | 2500       | 2500        | 2500      |
|          | Std  | 9.81E+00   | 0           | 4.56E-03    | 0          | 0           | 0         |
| F14      | Mean | 2600.8812  | 2600        | 2600.0021   | 2600       | 2600.0083   | 2600      |
|          | Std  | 0.1021     | 0           | 0.00335     | 0          | 0.00324     | 0         |
| F15      | Mean | 2700       | 2700        | 2700.00018  | 2700       | 2700        | 2700      |
|          | Std  | 0          | 0           | 3.65E-07    | 0          | 0           | 0         |
| F16      | Mean | 2800.0858  | 2800        | 2751.23     | 2755.23    | 2800        | 2800      |
|          | Std  | 3.31E-02   | 5.56E+01    | 7.65E+00    | 3.35E+00   | 7.54E+01    | 8.63E+00  |

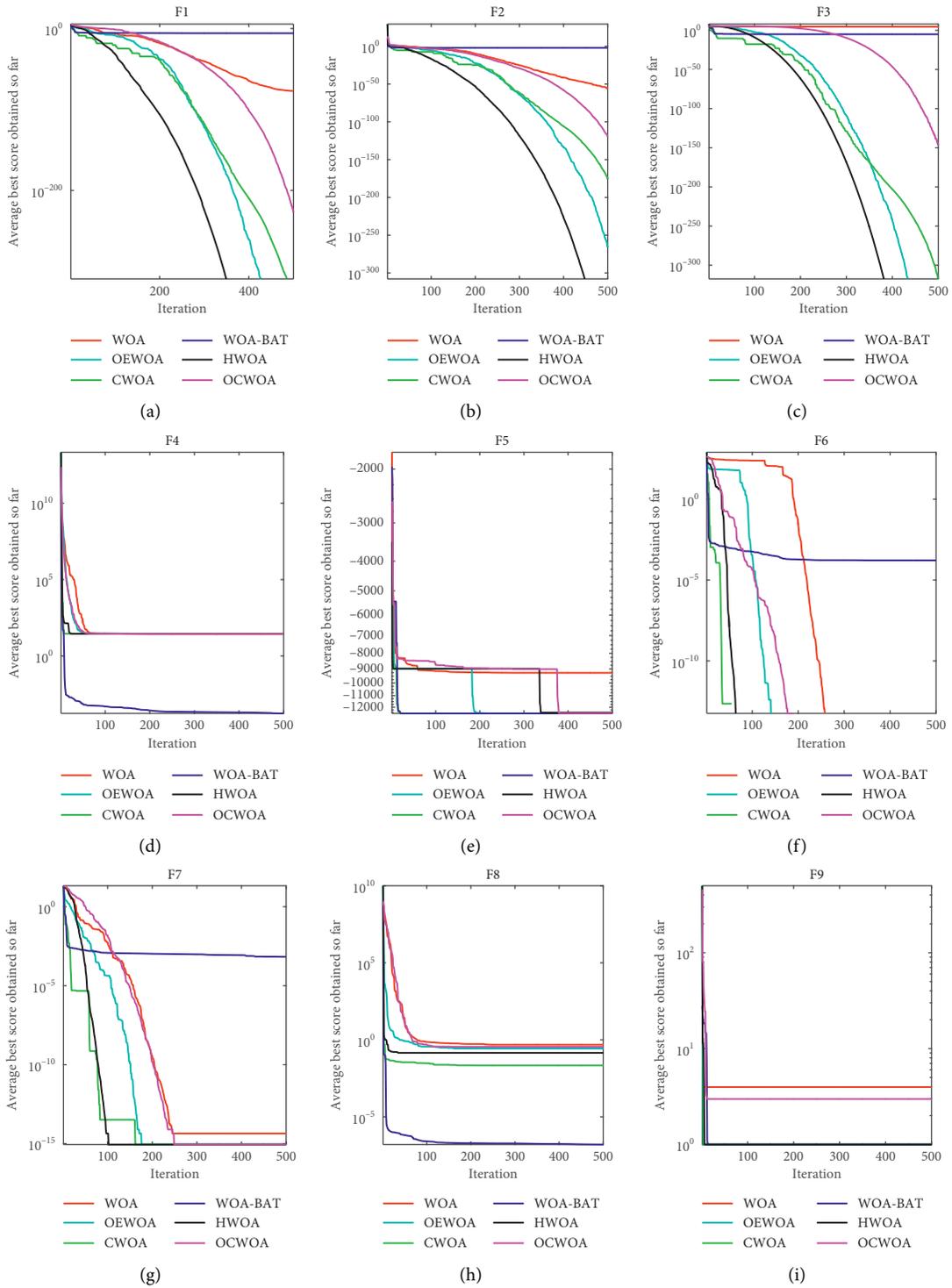


FIGURE 4: Continued.

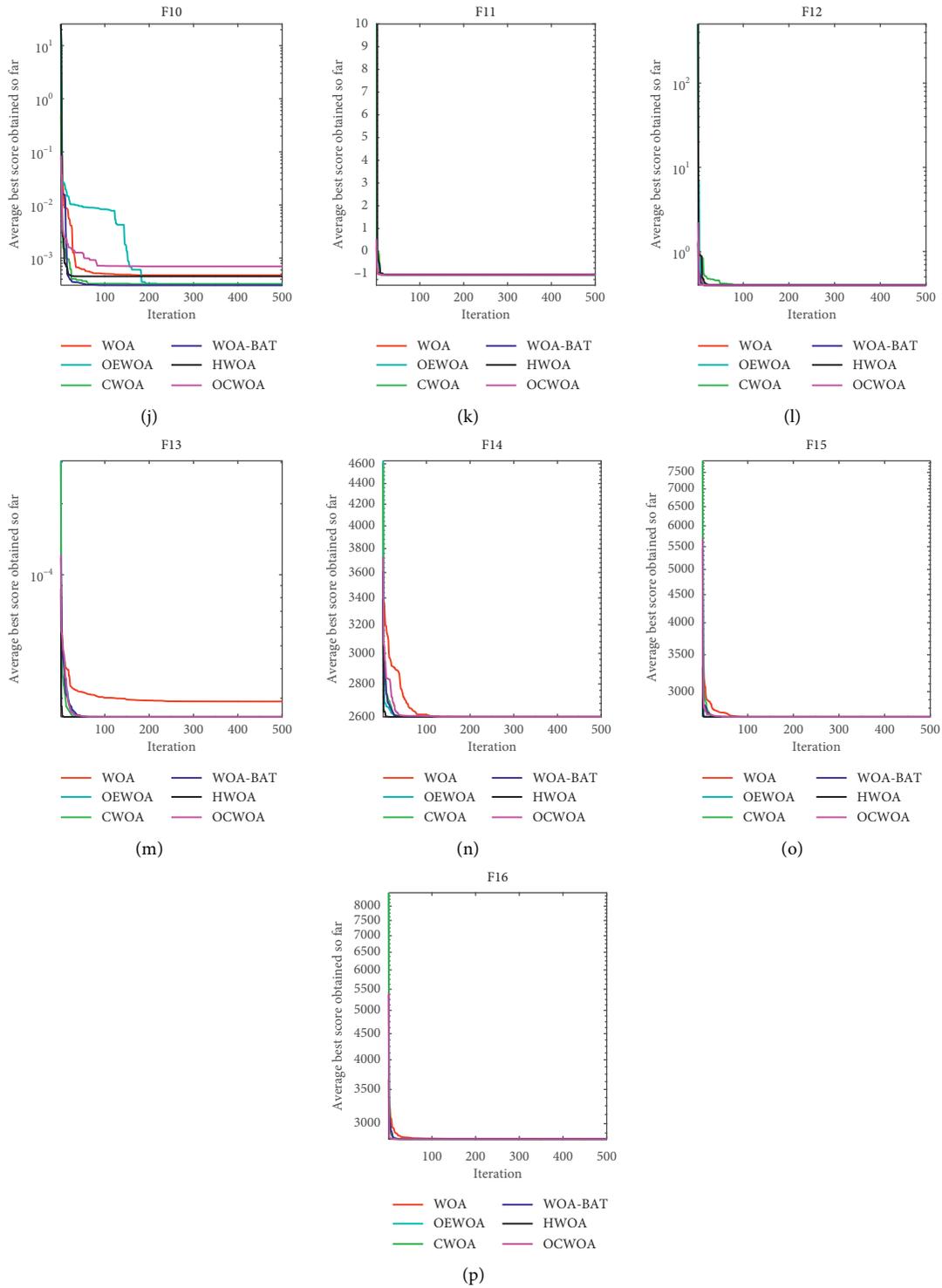


FIGURE 4: Average best fitness for six algorithms on 16 benchmark functions. (a) F1, (b) F2, (c) F3, (d) F4, (e) F5, (f) F6, (g) F7, (h) F8, (i) F9, (j) F10, (k) F11, (l) F12, (m) F13, (n) F14, (o) F15, and (p) F16.

Literature [44] shows that the parameter  $\gamma$  and regularization parameter  $C$  of RBF kernel will determine the classification accuracy of SVM, but the numbers of the two parameters have to be set manually. The different parameters will cause different classification accuracies

because HWOA will change the parameter set based on the classification accuracy (best fitness of HWOA) of SVM. So, we will use HWOA to optimize the RBF kernel parameter of SVM. The range of the parameter is from 0 to 1.

Algorithm 2 shows the pseudocode of HWOA-S3VM. Since the RBF kernel of SVM needs to specify its values to operate, HWOA is used to optimize the RBF kernel of SVM to select the most suitable values for processing.

**4.2. Experiment and Result Analysis of HWOA-S3VM.** In order to test the classification accuracy of the algorithm, it chooses 5 two-dimension UCI datasets to verify the classification performance of HWOA-S3VM compared with SVM, S3VM, RF [43], and WOA-S3VM.

Details of these datasets are shown in Table 7. In the experiment, firstly, each dataset is randomly divided into two parts: 30% of the data is training set and the remaining 70% is the testing set. In order to verify the ability to process labeled and unlabeled samples simultaneously of HWOA-S3VM, we set 10% and 20% of the training set as labeled data, and the rest of the training set are unlabeled data. The classification result is shown in Tables 8 and 9.

Tables 8 and 9 show that the HWOA-S3VM is superior to the other four algorithms at classification accuracy. Therefore, intelligent algorithm can improve the S3VM model classification accuracy. The accuracy of HWOA-S3VM will increase with the increase of the labeled sample. It means that the modified S3VM with WOA is successful.

## 5. Application of HWOA-S3VM in Oil Layer Recognition

**5.1. Design Model of Oil Layer Recognition.** Block diagram of the oil layer recognition system based on HWOA-S3VM is shown in Figure 5; according to the logging requirements, oil layer recognition can be divided into five steps.

Step 1: selection of sample information and preprocessing: it is important to select sample information because of the large amount of logging information. We can choose the data which are closely related to the oil layer information, and the data will randomly be divided into training data set and test dataset. In order to avoid the compute saturation, we will be normalizing all of the data.

Step 2: generalization of attribute: in this step, we set decision attribute  $D = \{d\}$ ,  $d = \{d_i = i, i = 0, 1\}$  (0, 1 represent nongas and gas reservoirs, respectively).

Step 3: attribute reduction: usually, there are at least 10 kinds of well logging information in well logging data, but there are redundant and sufficient attributes in most of the data; in this paper, we choose a rough set [45] to do the attribute reduction.

Step 4: HWOA-S3VM model: input the actual oil layer data to the HWOA-S3VM to train the model and get the HWOA-S3VM oil layer recognition model.

Step 5: output the result of HWOA-S3VM oil layer recognition model when the whole test data of the oil layer is inputted.

**5.2. Application on the Actual Oil Layer Datasets.** In Section 3, the performance of the proposed HWOA is simulated and analyzed on benchmark functions, and in Section 4.2, the effectiveness of the improved S3VM classification model optimized by the proposed HWOA is tested and verified on 5 two-dimension UCI datasets.

In this section, we use the three actual oil layer datasets (W1, W2, and W3) to verify the effectiveness of the HWOA-S3VM model. The detailed datasets of the oil layer are shown in Table 10. The output decision attribute  $D = \{d\}$ ,  $d = \{d_i = i, i = 0, 1\}$ , where 0 and 1 represent dry layer and oil layer, respectively. Table 11 shows the attribute reduction of three oil layer datasets. Figure 6 shows the curves of seven normalized attributes of W1; we use 20% of the training dataset as labeled data to train the classification model. The classification result is shown in Table 10. In Table 10, we compare the result of the supervised algorithms (RF and SVM) and semisupervised algorithms (S3VM, WOA-S3VM, and HWOA-S3VM).

In the oil layer recognition, it will use RMSE and MAE, to measure the performance of the recognition model. The equations of those are as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N e_i^2}, \quad (12)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |e_i|.$$

In those equations,  $e_i = y_i - \hat{y}_i$ , where  $\hat{y}_i$  is the real output value and  $y_i$  is the expected output value. The smaller the RMSE and MAE, the better the performance of the algorithm model will be.

**5.2.1. Result of Oil Layer Recognition.** The classification result is shown in Table 12; we compare the result of the supervised algorithms (RF algorithm and SVM) and semisupervised algorithms (S3VM, WOA-S3VM, and HWOA-S3VM).

The result of Table 12 shows that the intelligent algorithm can improve the performance of the S3VM model. The HWOA can improve the accuracy and running time compared to WOA-S3VM. HWOA-S3VM gets the best accuracy of W1, W2, and W3 compared to the other four algorithms. HWOA-S3VM can deal with a large number of unlabeled data in oil layer recognition. The results show that the HWOA can effectively improve the S3VM model.

Figure 7 shows the real distribution of oil layer and the distribution of oil layer detected by the HWOA-S3VM model. The HWOA-S3VM still gets high performance on oil recognition. The extra layer of HWOA-S3VM means that the algorithm can detect more data which the machine cannot detect. It means that HWOA can more effectively recognize the oil layer distribution accurately when there are an amount of unlabeled data in the oil layer.

TABLE 6: Result of HWOA compared with other algorithms with dimension 100.

| Function |      | WOA       | OEWOA     | WOA-BAT     | OCWOA      | CWOA        | HWOA      |
|----------|------|-----------|-----------|-------------|------------|-------------|-----------|
| F1       | Mean | 4.27E-75  | 0         | 3.09E-05    | 2.65E-85   | 0           | 0         |
|          | Std  | 3.65E-74  | 0         | 5.12E-04    | 2.66E-83   | 0           | 0         |
| F2       | Mean | 2.73E-51  | 2.23E-274 | 0.044423    | 2.67E-71   | 3.04E-178   | 0         |
|          | Std  | 1.78E-50  | 8.34E-261 | 0.003697    | 2.55E-69   | 4.78E-156   | 0         |
| F3       | Mean | 1.18E+06  | 0         | 1.24E-04    | 3.71E+05   | 0           | 0         |
|          | Std  | 2.45E+03  | 0         | 3.75E-05    | 7.28E+03   | 0           | 0         |
| F4       | Mean | 98.1806   | 98.1007   | 39.46923    | 98.4042    | 98.0242     | 98.3011   |
|          | Std  | 6.21E-01  | 7.89E-01  | 8.55E+01    | 3.71E-01   | 3.56E-07    | 3.21E-01  |
| F5       | Mean | -29948.77 | -41897.40 | -41898.2887 | -31886.764 | -31896.7649 | -41898.71 |
|          | Std  | 256.12    | 186.31    | 109.32      | 325.21     | 345.78      | 112.45    |
| F6       | Mean | 0         | 0         | 1.25E-03    | 0          | 0           | 0         |
|          | Std  | 0         | 0         | 3.34E-02    | 0          | 0           | 0         |
| F7       | Mean | 4.44E-15  | 8.88E-16  | 0.0019321   | 7.88E-16   | 8.88E-16    | 8.88E-16  |
|          | Std  | 3.23E-14  | 6.45E-16  | 3.55E-02    | 7.45E-15   | 1.34E-15    | 6.23E-15  |
| F8       | Mean | 3.0521    | 0.65848   | 1.70E-02    | 2.0965     | 3.0318      | 0.46097   |
|          | Std  | 3.21E-01  | 1.56E-02  | 2.21E-03    | 9.87E-01   | 7.63E-03    | 1.23E-01  |
| F13      | Mean | 2500      | 2500      | 2500.1481   | 2500       | 2500        | 2500      |
|          | Std  | 0         | 0         | 0.012489    | 0          | 0           | 0         |
| F14      | Mean | 2600.041  | 2600      | 2600.0477   | 2600       | 2600.5487   | 2600      |
|          | Std  | 0.002134  | 0         | 0.005677    | 0          | 0.009851    | 0         |
| F15      | Mean | 2700      | 2700      | 2700.0003   | 2700       | 2700        | 2700      |
|          | Std  | 0         | 0         | 3.65E-04    | 0          | 0           | 0         |
| F16      | Mean | 2789      | 2800      | 2791        | 2798       | 2800        | 2800      |
|          | Std  | 3.13E+00  | 1.54E+01  | 7.23E+01    | 3.78E+00   | 5.22E+00    | 8.45E+00  |

**Input:** labeled dataset  $D_l = \{(x_1, y_1), \dots, (x_l, y_l)\}$   
 Unlabeled dataset  $D_u = \{(x_{l+1}, \dots, x_{l+u})\}$   
 Tradeoff control parameters  $C_l, C_u^+$ , and  $C_u^-$   
**Output:** prediction result of unlabeled sample:  $\hat{y} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u})$   
 Final model of HWOA-S3VM

**Procedure:**

- (1) Train model  $SVM_l$  by  $D_l$  when using HWOA to get the kernel parameters  $\gamma$  and regularization parameters  $C$  of SVM
- (2) Using  $SVM_l$  to predict the  $D_u$ , get  $\hat{y} = (\hat{y}_{l+1}, \hat{y}_{l+2}, \dots, \hat{y}_{l+u})$
- (3) Initialization  $C_u^+ + C_u^- \ll C_l$
- (4) **While**  $C_u^+ + C_u^- < C_l$ , **do**
- (5) Based on  $D_l, D_u, \hat{y}, C_l, C_u^+$ , and  $C_u^-$ , solve equation (2)  
 To get  $(w, b), \zeta$
- (6) **While**  $\exists \{i, j, n \mid (\hat{y}_i \hat{y}_j < 0) \wedge (\zeta_i > 1) \wedge (\zeta_j > 1) \wedge (\zeta_n > 1) \wedge (\zeta_i + \zeta_j + \zeta_n > 3)\}$ , **do**  $y_i = -y_i; y_j = -y_j$  // label exchange;
- (7) Based on  $D_l, D_u, \hat{y}, C_l, C_u^+$ , and  $C_u^-$  to solve equation (2), to get  $(w, b), \zeta$
- (9) **End while**  $C_u^+ + C_u^- = \min\{2(C_u^+ + C_u^-), C_l\}$
- (10) **End while**

ALGORITHM 2: The pseudocode of the improved S3VM model (HWOA-S3VM).

TABLE 7: Characteristics of datasets.

| Name of data | Number of data | Attribution | Training set | Testing set |
|--------------|----------------|-------------|--------------|-------------|
| Haberman     | 306            | 3           | 90           | 210         |
| Pima         | 768            | 8           | 210          | 490         |
| Blood        | 748            | 4           | 210          | 490         |
| WDBC         | 569            | 2           | 150          | 350         |
| Seeds        | 210            | 7           | 60           | 140         |

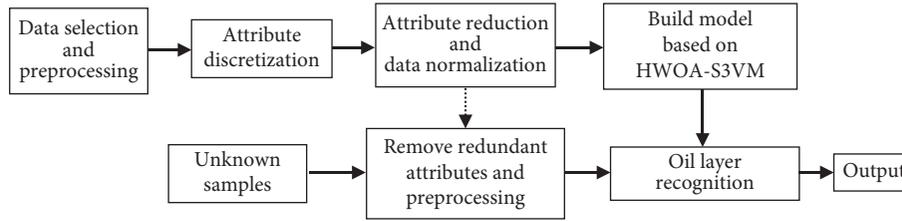


FIGURE 5: Diagram of oil layer recognition model.

TABLE 8: Comparison of numerical testing results on five UCI datasets when using 10% of training set as labeled data.

| Dataset  | RF (%) | SVM (%) | S3VM (%) | WOA-S3VM (%) | HWOA-S3VM (%) |
|----------|--------|---------|----------|--------------|---------------|
| Haberman | 75.2   | 81.10   | 79.05    | 81.43        | 83.80         |
| Pima     | 81.4   | 86.32   | 86.32    | 84.90        | 87.96         |
| Blood    | 81.5   | 85.51   | 85.91    | 88.78        | 88.78         |
| WDBC     | 78.00  | 86.30   | 90.28    | 89.50        | 92.57         |
| Seeds    | 82.86  | 87.50   | 88.64    | 88.57        | 90.00         |

TABLE 9: Comparison of numerical testing results on five UCI datasets when using 20% of training set as labeled data.

| Dataset  | RF (%) | SVM (%) | S3VM (%) | WOA-S3VM (%) | HWOA-S3VM (%) |
|----------|--------|---------|----------|--------------|---------------|
| Haberman | 79.52  | 81.80   | 82.50    | 83.5         | 86.67         |
| Pima     | 86.33  | 82.32   | 87.42    | 87.42        | 88.78         |
| Blood    | 86.12  | 83.47   | 82.48    | 89.10        | 90.61         |
| WDBC     | 87.14  | 83.15   | 85.10    | 90.00        | 95.43         |
| Seeds    | 84.20  | 87.85   | 88.64    | 89.29        | 91.43         |

TABLE 10: Oil layer dataset.

| Well |                  | Depth (m) | Number of data | Oil layer data | Dry layer data |
|------|------------------|-----------|----------------|----------------|----------------|
| W1   | Training dataset | 3020–3050 | 260            | 200            | 60             |
|      | Test dataset     | 3300–3350 | 804            | 106            | 698            |
| W2   | Training dataset | 2600–2606 | 30             | 10             | 20             |
|      | Test dataset     | 2900–2938 | 191            | 99             | 92             |
| W3   | Training dataset | 1200–1270 | 280            | 52             | 228            |
|      | Test dataset     | 1320–1460 | 1100           | 105            | 895            |

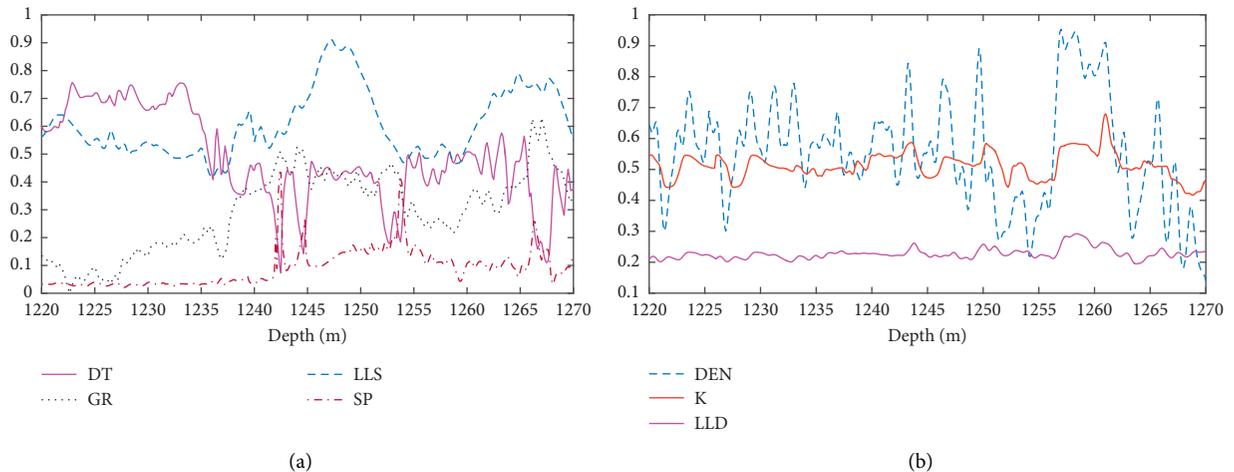


FIGURE 6: The normalized curves of seven attributes. (a) DT, GR, SP, and LLS. (b) DEN, K, and LLD.

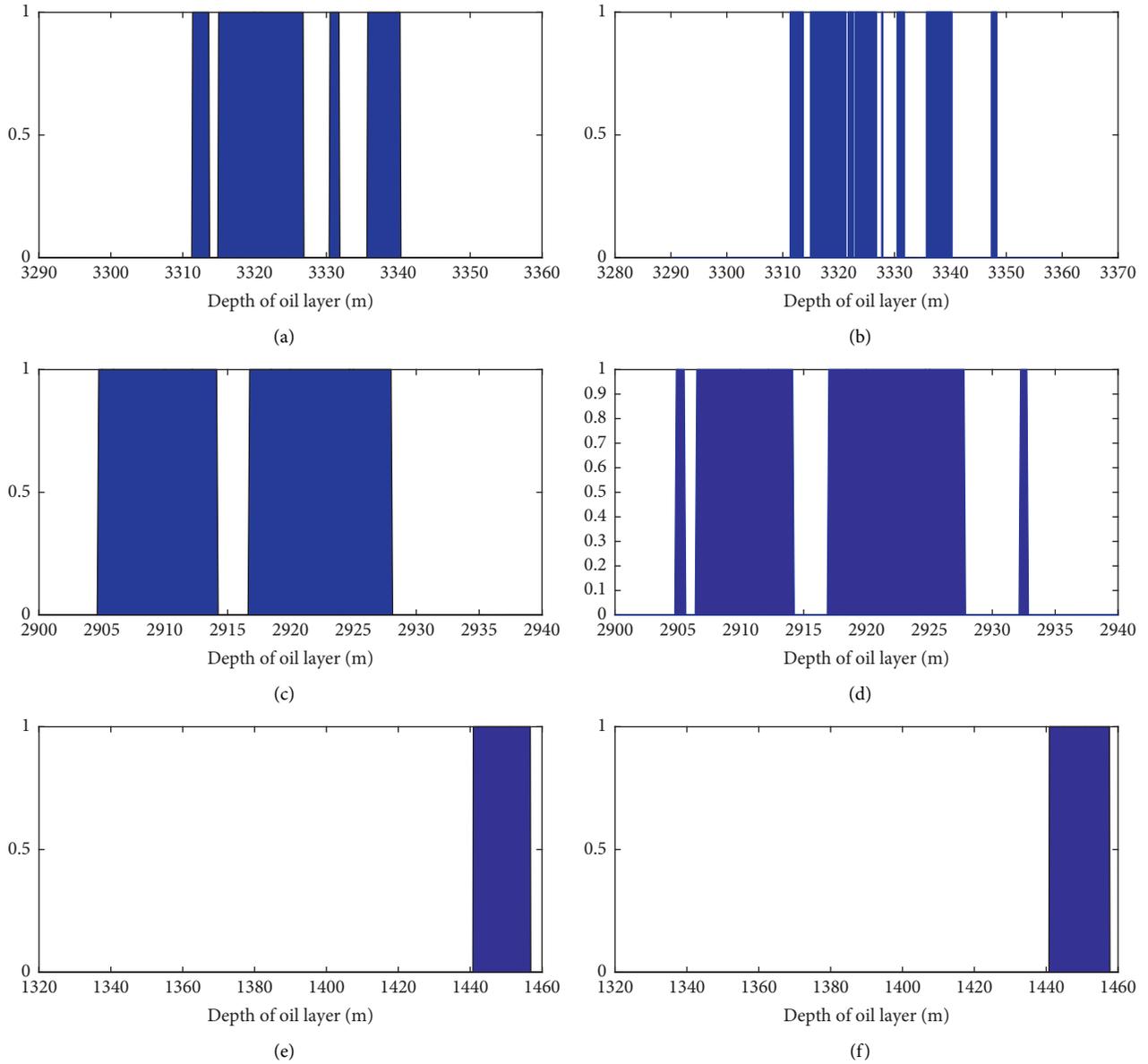


FIGURE 7: Classification of the HWOA-S3VM model. (a) Real distribution of oil layer of W1. (b) Distribution of oil layer detected by the HWOA-S3VM model of W1. (c) Real distribution of oil layer of W2. (d) Distribution of oil layer detected by the HWOA-S3VM model of W2. (e) Real distribution of oil layer of W3. (f) Distribution of oil layer detected by the HWOA-S3VM model of W3.

TABLE 11: Attribute reduction of oil layer dataset.

| Well | Attributes          |
|------|---------------------|
| W1   | Original attribute  |
|      | Reduction attribute |
| W2   | Original attribute  |
|      | Reduction attribute |
| W3   | Original attribute  |
|      | Reduction attribute |

GR, DT, SP, WQ, LLD, LLS, DEN, NPHI, PE, U, TH, K, CALI

GR, DT, SP, LLD, LLS, DEN, K

DENSITY, GAMM, VCLOK, NEUTRO, PERM, POR, RESI, SONIC, SP, SW

NEUTRO, PERM, POR, RESI, SW

AC, CALI, GR, NG, RA2, RA4, RI, RM, RT, RXO, SP

AC, NG, RI, SP

TABLE 12: Performance of various well data.

| Well | Model of classification | RMSE   | MAE    | Accuracy (%) | Time (s) |
|------|-------------------------|--------|--------|--------------|----------|
| W1   | RF                      | 0.3316 | 0.1317 | 89.05        | 1.5153   |
|      | SVM                     | 0.2751 | 0.0719 | 91.92        | 0.5861   |
|      | S3VM                    | 0.2921 | 0.1522 | 86.07        | 1.6821   |
|      | WOA-S3VM                | 0.3223 | 0.1892 | 92.66        | 2.0938   |
|      | CCOWA-S3VM              | 0.2234 | 0.1732 | 93.53        | 2.1958   |
| W2   | RF                      | 0.3129 | 0.1890 | 81.77        | 3.1478   |
|      | SVM                     | 0.2799 | 0.0895 | 85.66        | 0.5415   |
|      | S3VM                    | 0.3756 | 0.1261 | 91.58        | 1.1483   |
|      | WOA-S3VM                | 0.3566 | 0.1315 | 92.67        | 3.0255   |
|      | CCOWA_S3VM              | 0.2883 | 0.1722 | 96.32        | 3.8556   |
| W3   | RF                      | 0.3454 | 0.1514 | 83.48        | 1.6483   |
|      | SVM                     | 0.3245 | 0.1452 | 89.49        | 1.1214   |
|      | S3VM                    | 0.3537 | 0.1455 | 87.78        | 1.5723   |
|      | WOA-S3VM                | 0.3215 | 0.1257 | 91.00        | 2.1156   |
|      | CCOWA-S3VM              | 0.2231 | 0.1221 | 96.00        | 3.2123   |

## 6. Conclusions

In this paper, a new semisupervised SVM by hybrid whale optimization algorithm (HWOA-S3VM) is proposed, which includes the improved S3VM model with the tradeoff control parameter, S3VM model parameters optimized with HWOA, and HWOA improved with the opposition-based cubic mapping and catfish effect.

The experimental results show that (1) HWOA has higher convergence speed and better global searchability than other algorithms; (2) HWOA-S3VM model has higher classification accuracy on UCI datasets than other algorithms when combining labeled and unlabeled data as the training dataset; (3) the recognition accuracy and speed of the HWOA-S3VM model are superior to that of other four algorithms when applied in oil layer recognition.

## Data Availability

All data included in this study are available upon request by contact with the corresponding author.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. U1813222 and 42075129), Tianjin Natural Science Foundation (no. 18JCYBJC16500), and Key Research and Development Project from Hebei Province (nos. 19210404D and 20351802D).

## References

- [1] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised Learning*, MIT Press, Cambridge, MA, USA, 2006.
- [2] S. Ding, Z. Zhu, and X. Zhang, "An overview on semi-supervised support vector machine," *Neural Computing & Applications*, vol. 28, pp. 1–10, 2015.
- [3] X. Tian, G. Gasso, and S. Canu, "A multiple kernel framework for inductive semi-supervised SVM learning," *Neuro-computing*, vol. 90, pp. 46–58, 2012.
- [4] X. Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, University of Wisconsin–Madison, Madison, WI, USA, 2006.
- [5] B. Zu, K. Xia, Y. K. Pan et al., "A novel graph constructor for semi supervised discriminant analysis: combined low-rank and k-nearest neighbor graph," *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 9290230, 11 pages, 2017.
- [6] W.-P. Luo, H.-Q. Li, and N. Shi, "Semi-supervised least squares support vector machine algorithm: application to offshore oil reservoir," *Applied Geophysics*, vol. 13, no. 2, pp. 406–415, 2016.
- [7] Z. Dan, J. Licheng, B. Xue et al., "A robust semi-supervised SVM via ensemble learning," *Applied Soft Computing Journal*, vol. 65, pp. 632–643, 2018.
- [8] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowledge and Information Systems*, vol. 24, no. 3, pp. 415–439, 2010.
- [9] D. E. Goldberg, *GENETIC Algorithms*, Beowulf: Pearson Education India, Delhi, India, 2006.
- [10] T. Back, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic algorithms*, Concurrency: Oxford University Press, Oxford, UK, 1996.
- [11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Western Australia, November 1995.
- [12] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *Proceedings of the IEEE Swarm Intelligence Symposium*, vol. 8, no. 1, pp. 687–697, Indianapolis, IN, USA, January 2006.
- [13] R. Salgotra and U. Singh, "The naked mole-rat algorithm," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8837–8857, 2019.
- [14] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 971–680, 1983.
- [15] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

- [16] W.-T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example," *Knowledge-Based Systems*, vol. 26, pp. 69–74, 2012.
- [17] A. Kaveh and V. R. Mahdavi, "Colliding bodies optimization: a novel meta-heuristic method," *Computers & Structures*, vol. 139, pp. 18–27, 2014.
- [18] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [19] D. B. Prakash and C. Lakshminarayana, "Optimal siting of capacitor in radial distribution network using whale optimization algorithm," *Alexandria Engineering Journal*, vol. 56, no. 4, pp. 499–509, 2016.
- [20] H. Zhao, S. Guo, and H. Zhao, "Energy-Related CO<sub>2</sub> emissions forecasting using an improved LSSVM model optimized by whale optimization algorithm," *Energies*, vol. 10, no. 7, pp. 874–888, 2017.
- [21] R. K. Saidala and N. Devarakonda, "Improved whale optimization algorithm case study: clinical data of anemic pregnant woman," *Advances in Intelligent Systems and Computing, Data Engineering and Intelligent Computing*, vol. 542, pp. 271–281, 2016.
- [22] M. A. E. Aziz, A. A. Ewees, and A. E. Hassanien, "Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation," *Expert Systems with Applications*, vol. 83, pp. 242–256, 2017.
- [23] S. Osama, E. H. Essam, A. Darwish et al., "An optimized support vector regression using whale optimization for long term wind speed forecasting," *Series in Machine Perception and Artificial Intelligence, Hybrid Metaheuristics*, vol. 84, pp. 171–196, 2018.
- [24] M. M. Ahmed, H. Essam, E. H. Hassanien et al., "Maximizing lifetime of wireless sensor networks based on whale optimization algorithm," in *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, pp. 724–733, Cairo, Egypt, September 2017.
- [25] M. M. Ahmed, H. Essam, E. H. Houssein et al., "Maximizing lifetime of large-scale wireless sensor networks using multi-objective whale optimization algorithm," *Telecommunication Systems*, vol. 72, no. 2, pp. 243–259, 2019.
- [26] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Computing*, vol. 22, no. 1, pp. 1–15, 2018.
- [27] A. E. D. Oliva, "Parameter estimation of solar cells diode models by an improved opposition-based whale optimization algorithm," *Energy Conversion & Management*, vol. 171, pp. 1843–1859, 2018.
- [28] M. Abdel-Basset, G. Manogaran, D. El-shahat, and S. Mirjalili, "A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem," *Future Generation Computer Systems*, vol. 85, pp. 129–145, 2018.
- [29] A. Kaveh and M. I. Ghazaan, "Enhanced whale optimization algorithm for sizing optimization of skeletal structures," *Mechanics Based Design of Structures and Machines*, vol. 45, no. 3, pp. 345–362, 2017.
- [30] D. Oliva, M. Abd El Aziz, and A. E. Hassanien, "Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm," *Applied Energy*, vol. 200, pp. 141–154, 2017.
- [31] A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, and A. T. Azar, "New binary whale optimization algorithm for discrete optimization problems," *Engineering Optimization*, vol. 52, no. 6, pp. 945–959, 2020.
- [32] R. Salgotra, U. Singh, and S. Saha, "On some improved versions of whale optimization algorithm," *Arabian Journal for Science and Engineering*, vol. 44, no. 11, pp. 9653–9691, 2019.
- [33] H. M. Mohammed, U. Umarshahla, and A. Rashidtarik, "A systematic and meta-analysis survey of whale optimization algorithm," *Computational Intelligence and Neuroscience*, vol. 2019, Article ID 8718571, 25 pages, 2019.
- [34] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang, "Improved binary particle swarm optimization using catfish effect for feature selection," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12699–12707, 2011.
- [35] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, NY, USA, 1998.
- [36] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the 16th International Conference on Machine Learning*, pp. 200–209, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, August 1999.
- [37] J. Luo and B. Shi, "A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems," *Applied Intelligence*, vol. 49, no. 1, pp. 1–19, 2019.
- [38] G. Kaur and S. Arora, "Chaotic whale optimization algorithm," *Journal of Computational Design and Engineering*, vol. 5, no. 3, pp. 275–284, 2018.
- [39] S. Saremi, S. Mirjalili, and A. Lewis, "Biogeography-based optimisation with chaos," *Neural Computing and Applications*, vol. 25, no. 5, pp. 1077–1097, 2014.
- [40] D. Yang, G. Li, and G. Cheng, "On the efficiency of chaos optimization algorithms for global optimization," *Chaos, Solitons & Fractals*, vol. 34, no. 4, pp. 1366–1375, 2007.
- [41] P. N. Suganthan, N. Hansen, J. Liang et al., "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," *Natural Computing*, vol. 2005, pp. 341–357, 2005.
- [42] J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Technical report, Nanyang Technological University, Singapore, 2013.
- [43] L. Breiman, "Random forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [44] L. Zhang and B. Zhang, "Relationship between support vector set and kernel functions in SVM," *Journal of Computer Science and Technology*, vol. 17, no. 5, pp. 549–555, 2002.
- [45] J. Bai, K. Xia, Y. Lin, and P. Wu, "Attribute reduction based on consistent covering rough set and its application," *Complexity*, vol. 2017, no. 4, 9 pages, Article ID 8986917, 2017.