Hindawi
Scientific Programming
Volume 2021, Article ID 7589756, 13 pages
https://doi.org/10.1155/2021/7589756

Research Article

Adaptive Particle Swarm Optimization Algorithm Ensemble Model Applied to Classification of Unbalanced Data

Dawei Zheng,[1] Chao Qin ,[2] and Peipei Liu [2]

[1]*Monash University, School of Business, Clayton VIC 3800, Melbourne, Australia*
[2]*Shandong University of Finance and Economics, School of Computer Science and Technology, Jinan 250014, China*

Correspondence should be addressed to Chao Qin; 182115011@mail.sdufe.edu.cn

Received 19 July 2021; Revised 3 September 2021; Accepted 6 September 2021; Published 5 October 2021

Academic Editor: Pengwei Wang

Unbalanced data classification is a major challenge in the field of data mining. Random forest, as an ensemble learning method, is usually used to solve the problem of unbalanced data classification. For the existing random forest-based classification prediction model, its hyperparameters are dependent on empirical settings, which leads to the problem of unsatisfactory model performance. In order to make random forest find the optimum modelling corresponding to the character of unbalanced data sets and improve the accuracy of prediction, we apply the improved particle swarm optimization to set reasonable hyperparameters of the model. This paper proposes a random forest-based adaptive particle swarm optimization on data classification, and an adaptive particle swarm used to optimize the hyperparameters in the random forest to ensure that the model can better predict the unbalanced data accurately. Aiming at the premature convergence that appears in the particle swarm optimization algorithm, the population is adaptively divided according to the population fitness and the adaptive update strategy is introduced to enhance the ability of particles to jump out of the local optimum. Experimental results show that our proposed algorithms outperform the traditional ones, especially regarding the evaluation criterion of F1-measure and accuracy. The results on the six keel unbalanced data set the advantages of our proposed algorithms are presented.

1. Introduction

Classification [1] is a research field that has a place for data mining and is widely used in real life. For example, network intrusion detection, financial deception, spam filtering, and other issues have used classification technology [2–4]. The classification task is to obtain an objective model by learning the examples in the data set containing the instance and the label and use the objective model to predict the label of the next unknown instance.

The problem of unbalanced data classification [5] often exists in the field of data classification and has become one of the hot issues in recent years.

Unlike balanced data, the number of samples in different categories in unbalanced data varies greatly. In general, the category with more samples in unbalanced data is called negative class, while the category with fewer samples is called positive class. With a small number, the information provided to the classifier is relatively less. On the contrary, there are more negative sample data, which can provide more information to the classifier. In the case of unbalanced classification of data sets, the standard classifier is usually unable to achieve good classification results. Unbalanced data set classification often appears in many practical applications. For example, compared with people with good credit, default samples are usually small, and the identification target should be the default samples in credit scoring. A good classification model should be able to produce a high recognition accuracy for the default application. Misclassification of positive samples in unbalanced data classification will lead to serious consequences. So, it is very important to choose a classification model that can deal with unbalanced data.

The most commonly used methods to solve the problem of class imbalance are as follows: (1) resampling method [6], which is through undersampling and oversampling methods to eliminate most class instances or increase a few class instances to change the original class distribution of

unbalanced data; (2) cost-sensitive learning method [7], which assigns different values to the misclassification costs of different categories, generally, the minority in the categories are expensive, and the cost of majority is low; (3) ensemble strategy, which improves the generalization performance of existing learning algorithms and effective strategies, such as ensemble methods based on Bagging and Boosting. According to the famous "No Free Lunch Theorem" [8], a single classifier is not an effective solution for classification, as the characteristics of different data are disparate due to the size of the data set, data structure, and features. The concept of the ensemble learning is to combine multiple classifications, process different hypotheses to form a better hypothesis, and make predictions. Dietterich [9] explained the three basic reasons for the success of the ensemble method from a mathematical point of view: statistics, calculation, and representativeness. Kearns and Valiant [10] proved that as long as there is enough data, single learning algorithms can generate arbitrarily high precision estimates through ensemble. These studies show that the ensemble classifier has better learning ability than a single classifier.

The Random Forest (RF) algorithm is a bagging ensemble learning algorithm based on the random subspace method proposed by Bierman et al. [11]. This algorithm is a combined classification method. It is based on the Bootstrap sampling principle and randomly selects several different ones from the original data set. It trains sample subsets by using the unpruned decision tree as the base classifier to model each Bootstrap sample subset and ensemble the generated multiple decision trees; then, it uses the constructed decision tree group to classify and vote on the test data. Finally, the voting result determines the final classification result of the sample. The advantage of RF is that it can handle a large number of data features and generate unbiased estimates for generalized errors within the model; it can deal with the problem of data missing, especially for unbalanced classification data sets, RF can balance errors, and the algorithm is modelled in parallel, which runs fast. For imbalanced datasets, RF can balance errors. When there is classification imbalance, RF can provide an effective method to balance the data set error. Alhudhaif [12] used RF to classify the EEG signals of landlords with unbalanced data distribution. An adaptive sampling method is used to stabilize each sample, and then the RF is used to classify each balance block. The experimental results show that the RF effectively classifies unbalanced data signals.

The better performance of RF depends on suitable hyperparameter settings. Artificial adjustment on parameters is time-consuming and laborious. The better performance of RF depends on the appropriate hyperparameter settings. Manual adjustment of parameters takes time and effort. When in the data classification, the selection of hyperparameters such as the maximum number of features used by a single decision tree and the number of subtrees will directly control the tree structure of the model, which has a great impact on the performance of the classifier, and unsuitable parameter values may lead to overlearning or underlearning. Especially, in the face of unbalanced data sets, reasonable hyperparameter settings can help the model pay more attention to a small number of samples, so that the model can more effectively balance the error.

The learning-based intelligent optimization algorithm (LIOA) refers to an intelligent optimization algorithm with certain learning abilities. A large number of the LIOA algorithms have been proposed via inspiring by behavior of natural biological swarms (e.g., bees, flock of birds, schools of fish, herds of elephant, monarch butterfly, etc.) [13]. Such as elephant herding optimization (EHO) [14] is based on the herding behavior of elephants, and EHO uses a clan operator to update the distance of the elephants in each clan with respect to the position of a matriarch elephant. The superiority of the EHO method to several state-of-the-art metaheuristic algorithms has been demonstrated for many benchmark problems and in various application areas [15]. Krill herd (KH) is a novel swarm-based metaheuristic optimization algorithm inspired by the krill herding behavior. The objective function in the KH optimization process is based on the least distance between the food location and position of a krill. KH has drawn the attention of scholars and engineers due to its excellent performance [16].

Literature [13] has proven the effectiveness of the application of PSO, which belongs to LIOA in complex optimization scenarios and engineering applications. Particle Swarm Optimization (PSO) is an evolutionary computation coming from studying the behavior of flocks of birds [17]. Compared with other commonly swarm intelligence algorithms, the PSO algorithm has the advantages of simple algorithm principle, fewer adjustment parameters, small calculation amount, and strong global optimization ability [18]. The optimization method has wider applications. Song et al. [19] proposed a new three-phase hybrid feature selection (FS) algorithm based on correlation-guided clustering and PSO to tackle the "curse of dimensionality" and the high computational cost. Three kinds of FS methods are effectively integrated into the proposed algorithm based on their respective advantages. The improved integer PSO is used to improve the performance of the three phases. Experimental results show that the proposed algorithm can obtain a good feature subset with the lowest computational cost. Ji et al. [20] studied a dual-surrogate-assisted cooperative PSO algorithm to tackle EMMOPs. The proposed DCPSO mechanism uses the two populations to seek multiple modalities simultaneously, effectively balancing exploration and exploitation of the algorithm. Experimental results on 11 benchmark problems show that the proposed algorithm can find multiple globally/locally optimal solutions while obtaining the best optimal solution. A new bare-bones multiobjective particle swarm optimization algorithm was proposed by Zhang et al. [21] to solve the environmental/economic dispatch problems. The method with a particle updating strategy does not require tuning up control parameters, and the mutation operator and the constraint handling strategy were proposed to make the algorithm more effective in dealing with multiobjective optimization problems. The above applications of improved particle swarm optimization effectively solve corresponding problems in specific fields.

However, the faster convergence of PSO algorithm can easily cause parameter search to fall into local optimality, resulting in premature convergence. In response to this problem, based on previous research, we use the idea of clustering [22] to adaptively divide the particle swarm into different populations and guide the populations by applying different update strategies. This enhances the diversity of particles and helps particles jump out of a local optimum. On the basis of the above considerations, we use the adaptive particle swarm optimization (APSO) RF model for data classification to obtain the high accuracy prediction.

The rest of the paper is organized as follows: Section 2 presents the related methods processing problem of unbalanced data. Section 3 explains the related work on the methods used. Section 4 introduces the principle of the APSO-RF model. Section 5 describes the experimental setup. Section 6 reports the experimental analysis results. Finally, Section 7 concludes the paper and discusses the future work.

2. Related Work

Unbalanced data is processed from two aspects: data level and algorithm level. For the algorithm level, traditional machine learning methods such as Logistic regression (LR), Support Vector Machine (SVM), and Decision Tree (DT) have been applied to unbalanced problems in different fields.

LR has found wide acceptance as a model for describing the dependence of a binary response variable on a vector of explanatory variables. The maximum likelihood estimation method is used to fit LR, as it is a nonlinear least square estimation problem. Sampling is used to enhance the model's effects of responding to the unbalanced problem. SVM tries to find the decision boundary between various classes without actually worrying about the number of instances available for a class. SVM is suitable for high-dimensional problems and works with a small number of observations as well. DT is widely used in the field of unbalanced problem. By splitting the nodes, the DT forms a tree decision structure to predict instances. It is suitable for handling samples with missing attributes and massive data. The growth of branches without reasonable limits easily traps overfitting. Farquad and Bose [23] proposed a strategy of data balancing for handling imbalanced distribution in data. The proposed approach first employs SVM as a preprocessor, and the actual target values of training data are then replaced by the predictions of trained SVM. Later, this modified training data is used to train techniques such as multilayer perceptron (MLP), LR, and RF. The result observed that the proposed approach balances the data effectively. Wang et al. [24] developed models where minority oversampling technology is combined with machine learning methods to identify the unbalanced medical data. LR, SVM, and RF were used to build risk identification models. The result showed that the model combined with machine learning methods effectively improved the discrimination efficacy of adverse outcomes in heart failure patients.

Neural network (NN) has been widely applied in the field of assessment including target threat, transformer fault diagnosis, and detection of malicious code. The performance of NN on imbalanced classification is more accurate by selecting specific evaluation indexes and adjusting the weight to match the loss value of the two types of samples during the calculation of the loss function. Wang [25] et al. proposed a wavelet mother function selection algorithm with minimum mean squared error and then construct MWFWNN network using the above algorithm. Firstly, it needs to establish a wavelet function library; secondly, a wavelet neural network is constructed with each wavelet mother function in the library and wavelet function parameters, and the network weights are updated according to the relevant modifying formula. Yi et al. [26] presented a self-adaptive probabilistic neural network. In the network, Spread can be self-adaptively adjusted and selected, and then the best-selected Spread is used to guide the self-adaptive probabilistic neural network train and test. In [27], a self-adaptive extreme learning machine (SaELM) was present. Its self-adaptive learning algorithm can select the best neuron number in the hidden layer to form the neural networks without adjusting any parameters in the training process. Cui et al. [28] used the convolutional neural network (CNN) to extract the features of the malware images automatically and utilized a bat algorithm to address the data imbalance among different malware families. The experimental results demonstrated that the model achieved good accuracy and speed as compared with other malware detection models. Wang et al. [29] established the Elman-AdaBoost model for target threat assessment, which used Elman neural network as weak predictor trained to predict sample output repeatedly. The results show that Elman-AdaBoost model and algorithm have good prediction ability.

As traditional single classifiers, NN, SVM, and DT commonly have an insufficient amount of training data, and their hypothesis space is small, so that they are easy to obtain the local optimal value. By combining several classifiers, ensemble learning can improve the generalization performance by improving the learning effect of the algorithm on unbalanced data. As a classical ensemble learning model, RF can use bootstrap sampling, that is, random select sample and random feature selection of base learner, to effectively balance data set errors, which is beneficial in reducing overfitting in case of imbalanced classification. In [30, 31], the RF optimized by PSO algorithm was applied to the particular task and achieved better prediction accuracy.

The wide application of learning-based intelligent optimization algorithm has offered thought on the improvement of PSO. Wang et al. [32] proposed a novel hybrid metaheuristic optimization approach by adding differential evolution (DE) mutation operator to the accelerated particle swarm optimization (APSO) algorithm to solve numerical optimization problems. Wang et al. [33] improved the performance of the krill herd (KH) algorithm by proposing a series of chaotic particle-swarm krill herd (CPKH) algorithms that are used for solving optimization tasks within limited time requirements. Wang et al. [34] purposed a new hybrid algorithm, that is, annealing krill quantum particle swarm optimization (AKQPSO) algorithm based on the annealing krill herd algorithm (AKH) and quantum particle

swarm optimization algorithm (QPSO). In combination with favorable performance in the exploitation of QPSO and good performance in the exploration of AKH, AKQPSO increases the diversity of population individuals and shows better performance in both exploitation and exploration. Mirjalili and Wang [35] proposed a binary version of hybrid PSOGSA called BPSOGSA to solve problems. PSOGSA combined with PSO and gravitational search algorithm (GSA) has binary parameters problem. The experimental results confirm the better performance of BPSOGSA in terms of avoiding local minima and convergence rate. In [36], a novel hybrid KH–QPSO combined with Krill herd (KH) and quantum behaved particle swarm optimization (QPSO) was presented for benchmark and engineering optimization. QPSO is intended for enhancing the ability of the local search and increasing the individual diversity in the population. The experimental results showed that KH–QPSO is more efficient for solving standard test problems and engineering optimization problems. Zou et al. [37] presented improved PSO (IPSO) to solve the infinitive impulse response (IIR) system identification problem. The population initialization step makes use of golden ratio to segment solution space, then all particles using different inertia weights in velocity updating step, and uses the normal distribution to disturb the global best particle. The three operations guarantee high-quality solutions, strong global search capacity, and fast convergence rate and avoid low diversity, excessive local search, and premature stagnation. These properties of IPSO make it much better suited for IIR system identification problems.

Based on the advantages of RF in reasonably addressing the imbalanced data, its mechanisms to prevent overfitting, and the advantages of PSO, we build an RF model that is optimized by an improved APSO to realize high-precision model. APSO sets the objective function as the fitness of the particle and directs the particle to simultaneously optimize multiple hyperparameters and obtains the hyperparameter value that makes the objective function lower. Its reasonable hyperparameter setting enables RF to fully exert its effect and improve the accurate performance.

3. Materials and Methods

In this section, we introduce the related works about techniques of RF and PSO.

3.1. Decision Tree. Classification and Regression Tree (CART) is an inductive learning algorithm for a single classification regressor, which is composed of root nodes, leaf nodes, and non-leaf nodes. The decision tree generates a path from the root node to the leaf node through regression analysis on the training set and analyzes the path rules. Classify or predict new instance according to path rules.

CART is based on information entropy and uses the Gini index minimum principal index to split the node. The input space of the training set $\mathbf{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is divided into regions, and each sample is recursively divided into the corresponding region, and a determined output value is obtained. The steps of the algorithm are as follows:

(1) Assuming that the feature of the independent variable is j, the value of this feature is s. Assuming that the value s divides the space of feature j into two regions, the formula is as follows:

$$R_1(j,s) = \left\{x|x^{(j)} \le s\right\}, R_2(j,s) = \left\{x|x^{(j)} > s\right\}. \quad (1)$$

(2) Traverse and calculate the loss function of each segmentation point in turn, and select the segmentation point with the smallest loss function.

$$\text{loss} = \min_{j,s}\left[\min_{c_1}\sum_{x_i\in R_1(j,s)}(y_i - c_2)^2 + \min_{c_2}\sum_{x_i\in R_2(j,s)}(y_i - c_2)^2\right]. \quad (2)$$

(3) Among them, c_1 and c_2 are the output average values in the intervals R_1 and R_2, respectively.

(4) Calculate the point of division, and proceed in sequence until the division can no longer be continued.

(5) Divide the input space into M parts D_k to generate the final decision tree as

$$f(x) = \sum_{m=1}^{M} c_m I(x \in R_m). \quad (3)$$

In the classification task, CART builds the branch according to the Gini index minimization criterion. Assuming that the probability of the sample point belongs to the Kth class is p_k, the Gini index of the probability distribution is defined as

$$\text{Gini}(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2. \quad (4)$$

For the binary classification task, if the probability that the sample point belongs to the first class is p, then the Gini index of the probability distribution is

$$\text{Gini}(p) = 2p(1 - p). \quad (5)$$

For a given sample set D, its Gini index is

$$\text{Gini}(\mathbf{D}) = 1 - \sum_{k=1}^{K}\left(\frac{|C_k|}{|\mathbf{D}|}\right)^2. \quad (6)$$

C_k is the subset of class k, and K is the number of classes.

If the sample set D is divided into R1 and R2 according to feature j taking certain possible value s, $\mathbf{R}_1 = \{(x, y) \in \mathbf{D}| x^{(j)} = s\}$, $\mathbf{R}_2 = \mathbf{D} - \mathbf{D}_1$, the Gini index of set D is defined as

$$\text{Gini}(\mathbf{D}, A) = \frac{|\mathbf{R}_1|}{|\mathbf{D}|}\text{Gini}(\mathbf{R}_1) + \frac{|\mathbf{R}_2|}{|\mathbf{D}|}\text{Gini}(\mathbf{R}_2). \quad (7)$$

Gini (D) represents the uncertainty of set D, and Gini (D, s) represents the uncertainty of set D after being segmented by $j = s$. The pseudocode of this algorithm is shown in Figure 1.

Algorithm 1: CART for Classification.

Input: training data set D
Output: decision tree T

1 The algorithm starts from the root node and recursively establishes tree with the training set;
2 For the training data set of nodes D, calculate the Gini index of features from D. For each feature of current node and their each possible value, according to $j = s$ the data set D divided into R_1 and R_2;
3 In all possible features j and all possible segmentation point s, select the feature with minimum Gini index and its corresponding cut points as the optimal features and optimal cut points respectly. According to the optimal feature and the optimal cut point, two children are generated from the current node, and the training data set is allocated to the two child nodes;
4 Recursively run step 2 and step 3 on two child nodes until Gini index of sample set less than the threshold value;

Figure 1: Pseudocode of Decision Tree.

3.2. Random Forest. RF is composed of multiple decision trees combined into a strong classifier on the basis of bagging (see Figure 2). It uses Bootstrap to randomly sample m instances with replacement on the training set and selects random features for each decision tree. Build m decision tree models from these m samples. Finally, the results are obtained by voting through these m models. The specific algorithm steps are as follows:

The training set D is input. Use Bootstrap sampling to form K training subsets. Randomly extract m features from the original features. Perform training on the training subset, make the optimal segmentation of the randomly selected m features, and obtain K decision tree prediction results. Voting is based on K prediction results to get the prediction result with the highest number of votes. The pseudocode of this algorithm is shown in Figure 3.

3.3. PSO. The PSO algorithm simulates a bird in a flock of birds by designing a massless particle. This particle has only two attributes: speed and position. Speed represents the speed at which it moves, and position represents its spatial position. Each particle finds the optimal solution in the individual search space, stores it as the current individual extreme value, finds the current global optimal solution according to the individual extreme values of all current particles, and adjusts its speed and position for the entire particle swarm. The traditional PSO algorithm is described as follows:

Suppose there is a population of m particles in the d-dimensional search space. Suppose, at time T, population particle information:

$$\text{Position } X_i = [x_i^1, x_i^2, \ldots, x_i^d]$$
$$\text{Speed } V_i = [v_i^1, v_i^2, \ldots, v_i^d]$$
$$\text{Personal best position } p_i^t = [p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t]$$
$$\text{Global optimal position } p_g = [p_g^1, p_g^2, \ldots, p_g^d]$$

Then, the speed and position information of the particles is updated at time T + 1 by the following formula:

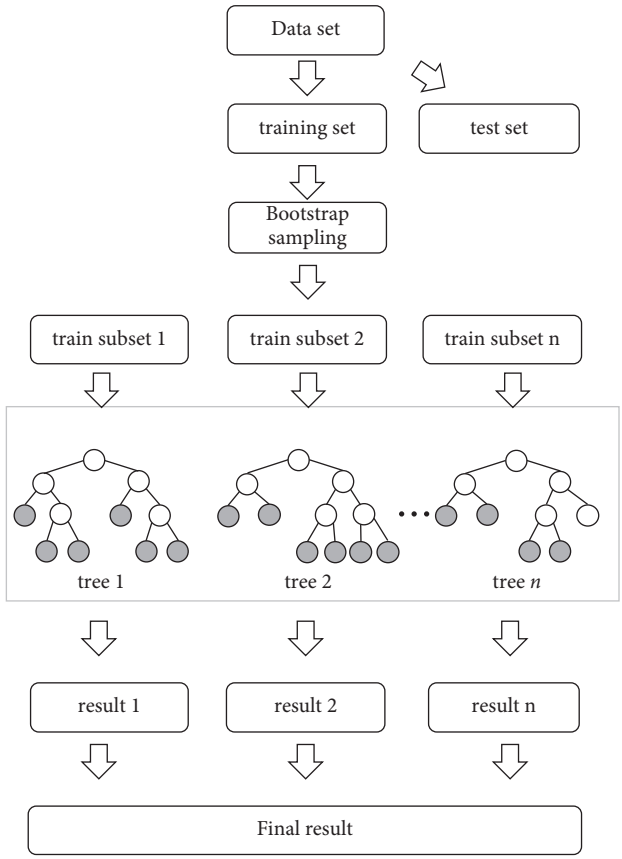

Figure 2: Random forest.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1^t \left(p_i^t - x_i^t\right) + c_2 r_2^t \left(p_g^t - x_i^t\right),$$
$$x_i^{t+1} = x_i^t + v_i^{t+1}. \tag{8}$$

Among them, the inertia weight ω maintains an effective balance between global exploration and local exploration, c_1 and c_2 are the learning factors, respectively, responsible for adjusting the step length in the exploration direction to the

Algorithm 2: Random Forest for Classification.

for $n = 1; k \le K; k = k + 1$ **do**
  Draw a bootstrap sample Z of size M from the training data;
  Grow a random-forest tree T_k to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size m_{min} is reached;
    i. Select m variables at random from the p variables;
    ii. Pick the best variable/split-point among the m;
    iii. Split the node into two daughter nodes.
Output the ensemble of trees $\{T_k\}_1^K$;
To make a prediction at a new point x:
  Let $f_k(x)$ be the class prediction of the *bth* random-forest tree. Then
  $f_{RF}(x) = majority\ vote\{f_k(x)\}_1^K$.

Figure 3: Pseudocode of random forest.

optimal position of the population and the exploration direction to the global optimal position, and r_1 and r_2 are random numbers on the uniform distribution function. In order to avoid blind search of particles, their speed and position are usually limited to $[-V_{\max}, V_{\max}]$, $[-X_{\max}, X_{\max}]$.

4. Adaptive Particle Swarm Optimization Algorithm Ensemble Model

In this section, we introduce the structure of the model APSO-RF in detail. First, PSO improved by adaptive learning strategies is shown. In the process of searching, the group is adaptively divided into subgroups according to the particle distribution. In each subgroup, we use two different learning strategies to guide the search directions of two different types of particles. Then, the optimization model building process is introduced. By applying APSO to optimize the selected hyperparameters, the classification model was established.

4.1. Adaptive Particle Swarm Optimization Algorithm. Relevant studies have shown that the diversity of the population is the key to avoiding the premature convergence of PSO; the core guiding principle of the algorithm is clustering. According to the distribution of each particle, the fast search clustering method [38] is adopted to perform the adaptive division of the population into several subgroups. This method can automatically discover the data set samples' class cluster center. The basic principle is that the center of the class cluster has two basic features: the first is that it is surrounded by points with lower local density, and the second is that it has a greater distance from points with a higher local density. Therefore, for a population of N particles $S = \{x_i\}_{i=1}^H$, the two properties ρ_i and δ_i are defined for each particle. ρ_i, the distance between the local density of the particle and a higher local density of particles, is defined as follows:

$$\rho_i = \sum_{j \ne i} \exp\left(-\left(\frac{\mathbf{d}_{\mathbf{ij}}^2}{d_c}\right)\right), \tag{9}$$

where d_{ij} is the Euclidean distance of particles between x_i and x_j and d_c is the truncation distance. The truncation distance is $d_c = d_{R*M}$, where R represents the proportion, and M indicates that the matrix $\mathbf{d}_{\mathbf{ij}}$ contains $M = 1/2N(N-1)$ values, where N represents the number of particles. It can be seen that d_c is the distance corresponding to the $R * M$th value of $\mathbf{d}_{\mathbf{ij}}$. (10) gives the expression of the distance δ_i, representing the minimum distance from particle i to other particles that have a higher ρ_i:

$$\delta_i = \min_{j:\rho_j > \rho_i}\left(\mathbf{d}_{\mathbf{ij}}\right). \tag{10}$$

For the maximum local density ρ of the sample, $\delta_i = \max_j \mathbf{d}_{\mathbf{ij}}$.

According to Equation (9), if the density of particle x_i is the maximum, δ_i is much larger than the distance δ of its nearest particles. Therefore, the center of the subgroup consists of particles that have an unusually large distance δ and a relatively high density as well. In other words, the particles with larger ρ and δ values are selected as the center of the cluster.

According to the above idea from [38], the formula $\gamma_i = \rho_i * \delta_i$ is used to filter out particles that may become cluster centers. We arrange the γ_i values in descending order and then use the truncation distance to filter out the cluster centers from the order.

Because the γ value of the top particle is more likely to increase exponentially than those of the other particles, it is distinguished from the γ value of the next particle. Referring to [38], R is set to be between 0.1 and 0.2. Through a parameter sensitivity analysis, we found that the value of the distribution parameter has no effect on the performance of the particle swarm algorithm. The default value in this article is 0.2. The cluster center is obtained by dividing by the truncation distance after placing the other particles x_j in subgroups, where the denser ρ is larger than ρ of x_j and γ is the closest to γ of x_j.

The particles of each subgroup are divided into ordinary particles, and local optimal particles based on the result of the division of subgroups. Under the primary guidance of the optimal particles, the ordinary particles exert their local search ability, and the updated formula is given as

$$x_i^d = \omega x_i^d + c_1 \mathrm{rand}_1^d\left(\mathrm{pbest}_i^d - x_i^d\right) + c_2 \mathrm{rand}_2^d\left(\mathrm{cgbest}_c^d - x_i^d\right), \tag{11}$$

where ω is the inertia weight, c_1 and c_2 are the learning factors, rand_1^d and rand_2^d are uniformly distributed random numbers in the interval $[0, 1]$, pbest_i^d is the best position of particles, and cgbest_c^d is the current best position of particle in the subgroup c. To enhance the exchange of information between subgroups, the local optimal particles are mainly updated by integrating the information of each subgroup. The update formula is as follows (see (12)), where C is the number of subgroups.

$$x_i^d = \omega x_i^d + c_1 \mathrm{rand}_1^d\left(\mathrm{pbest}_i^d - x_i^d\right) + c_2 \mathrm{rand}_2^d\left(\frac{1}{C}\sum_{c=1} \mathrm{cgbest}_c^d - x_i^d\right). \tag{12}$$

Ordinary particles search for local optimality, but more importantly, they are used as the medium for information exchange between subgroups to modify the direction of population search and further improve the population diversity. In the same subgroup, unlike a learning strategy that

causes too many particles to be gathered locally, the learning strategy integrates the information of the locally optimal particles from different subgroups to obtain more information and help avoid local optima.

In addition, learning too much information may lead to the direction of the update being too fuzzy, which may counteract the convergence of particles.

Considering that the local optimal particles have the maximum probability of finding the optimal solution in the subgroup, valuable guidance for the optimal solution is provided by their information. Therefore, the cgbest_c^d of each subgroup uses the average information to guide the local optimal particle update (see (12)). The transmission of the optimized information in the subgroups can be improved by this approach, the population diversity can be further increased, and particles can be prevented from falling into local optima.

4.2. APSO-RF. In order to make the model structure of RF match the data features more accurately and get the classification prediction results accurately, we use adaptive particle swarm optimization to control the hyperparameters of the model structure and build the APSO-RF model. By adaptively dividing the population, the update strategy guides the particle information update to avoid the particles from falling into the local optimum, thereby overcoming the shortcomings of traditional particle swarms (see Figure 4). The main steps of the model are as follows.

First, the hyperparameters in the RF model are taken as the optimization target, and the position information of each particle is randomly initialized in the set hyperparameter value space.

Second, the particles are divided into adaptive populations. This step is realized by calculating the local density of the particles and the distance to the higher local density particles. According to the value determined by the particle position, the hyperparameters of the RF model are assigned, and the verification data is brought into the model for prediction, and the loss function value of the model on the verification data set is used as the particle fitness value.

Based on loss function, $\text{logistic} = \log(1 + \exp(-\text{yp}))$ to set the fitness value, where p is the predictive value and y represents the true value. According to the fitness value of each particle, the subgroup is divided into various types of particles. Use the update strategy to update the information of different types of particles. When the termination condition is reached, the optimal value in the current parameter space is obtained. Finally, the RF model is constructed with the optimal value of the hyperparameter. Detailed steps are expressed in the pseudocode as shown in Figure 5.

4.3. Cross Validation. Cross validation can make full use of limited data to find appropriate model parameters to prevent overfitting. The main steps of K-fold cross validation are as follows: the initial sampling is divided into K subsamples, a separate subsample is used as the data of the validation model, and other K-1 samples are used for training. Cross validation is repeated K times, each subsample is verified once, the average K times resulted, and, finally, a single estimate is obtained. The advantage of the method is that the randomly generated subsamples are repeatedly used for training and verification. In the experiment, we used the most common 10-fold cross validation.

4.4. Data Preprocessing. Although the tree-based algorithm is not affected by scaling, feature normalization can greatly improve the accuracy of classifiers. The training set is described as $\mathbf{D} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X} = \{x_1, x_2, \ldots, x_m\}$ represents an m-dimensional eigenspace, and $\mathbf{Y} = \{0, 1\}$ represents the target value. If x is a certain feature, it is done by $0 - 1$ scaling as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{13}$$

where x' expresses the standardized value.

5. Experiment Setting

5.1. Data Set. The experimental data of this study is an unbalanced data set obtained in the keel data mining platform (see Table 1). All the imbalanced data sets are available with imbalance ratio between 1.5 and 9. The specific details of the data set are shown in the table. IR represents the class imbalance ratio.

5.2. Setting Range of Hyperparameters. According to the previous RF parameter optimization research, we put a group of hyperparameters as optimization targets and set their search space. The setting range is shown in Table 2.

5.3. Measure. To compare the results of the evaluation model, we use the evaluation criteria based on confusion matrix (see Table 3). True Positive (TP) is the number of samples that are predicted to be positive class; True negation (TN) is the number of actual negative samples and predicted negative samples; false positive (FP) is the number of actual negative samples and predicted positive samples; false negative (FN) is the number of actual positive samples and predicted negative samples.

Both F1-measure and Roc Area are comprehensive measures of the ability to deal with unbalanced data sets. The formulas are as follows.

The average accuracy (ACC):

$$\frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}. \tag{14}$$

The F1-measure takes into account both precision and recall of classification models. It is the harmonic average of these two indicators, and it ranges from 0 to 1. ROC is a graph to judge the accuracy of the prediction. If the graph area is close to 1, it is 100% correct.

$$F2 = \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}, \tag{15}$$

where precision is the proportion of positive samples in positive cases, and it is defined as

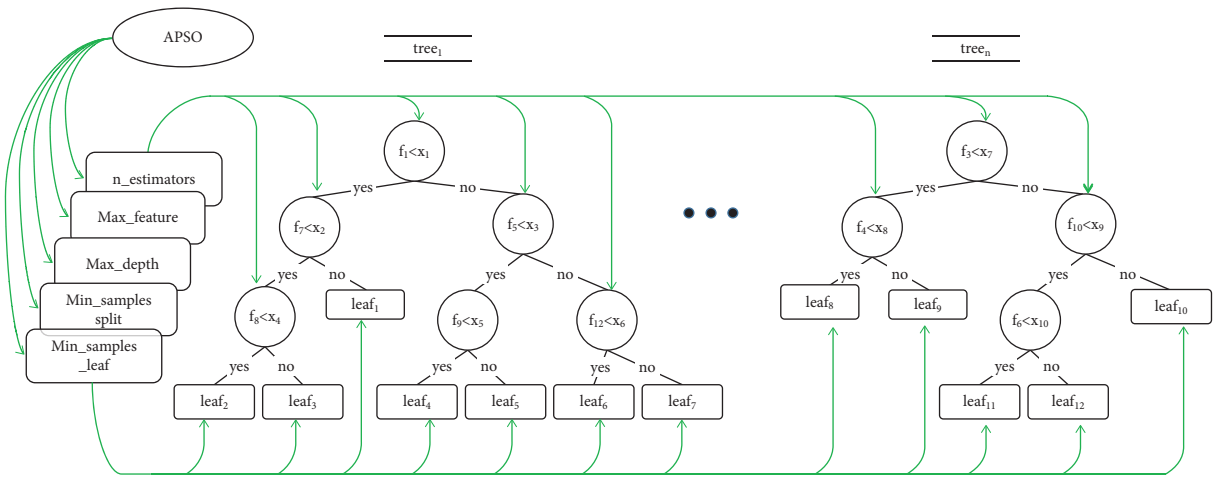

Figure 4: APSO-RF.

Algorithm 3: RF optimizated by APSO

Input: $DatasetD = \{(x, y_1), (x_2, y_2), \ldots (x_m, y_m)\}$

Output: Optimal value of hyper-parameter of RF

Initializes particles $x_i = (x_i^1, x_i^2, \ldots, x_i^D)$ with position $X_i = [x_i^1, x_i^2, \ldots, x_i^D]$ and velocity $V_i = [v_i^1, v_i^2, \ldots, v_i^D]$

for $i = 1; i \le N; i = i + 1$ **do**

  Compute the local density δ_i, and the distance ρ_i;

  Choose particles with high δ_i and relatively high ρ_i as centers according to $\gamma_i = \rho_i * \delta_i$;

  Assign remaining particles and get C subgroups;

  Initialize RF with instance nodes set I on train data, the hyper-parameter ← current optimal value;

  for $t = 1; t \le T; t = t + 1$ **do**

    The t_th time randomly sample the training set, and the sampling set D_m containing m samples was obtained ;

    On D_m train the mth decision tree model $G_m(x)$, select k sample features randomly, select the best segmentation attribute as node based on Gini to establish trees;

  Update particle state ($pBest_i$, $gBest$) refer to the loss function

  for $c = 1; c \le C; c = c + 1$ **do**

    if *particle is local optimal* **then**

      $v_i^d = \omega v_i^d + c_1 rand_1^d (pbest_i^d - x_i^d) + c_2 rand_2^d (\frac{1}{C} \sum_{c=1} cgbest_c^d - x_i^d)$

      $x_i^d = x_i^d + v_i^d$

    else

      $v_i^d = \omega v_i^d + c_1 rand_1^d (pbest_i^d - x_i^d) + c_2 rand_2^d (cgbest_c^d - x_i^d)$

      $x_i^d = x_i^d + v_i^d$

Figure 5: Pseudocode of APSO-RF.

Table 1: The Keel data set.

Data set	Attributes	Examples	IR
ecoli-3	7	336	8.6
glass-1	9	214	1.82
new-thyroid-1	5	215	5.14
page-blocks-0	10	5472	8.79
vehicle-1	18	846	2.9
wisconsin	9	683	1.86
yeast-1	8	1484	2.46

Table 2: Range of hyperparameters.

Hyperparameter	APSO
N_estimators	(50–200)
Max_features	(12–16)
Max_depth	350,400,450
Min_samples_split	(2, 3)
Min_samples_leaf	(1, 5)

Table 3: Confusion matrix.

Actual value		0	1	
	0	TP	FN	TP + FN
Predicted value	1	FP	TN	FP + TN
		TP + FP	FN + TN	TP + FP + FN + TN

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{16}$$

And, recall is the proportion of predicted positive cases in the total positive cases; it is defined as

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{17}$$

5.4. Baseline Model. In order to analyze and verify the performance of the proposed model for unbalanced data classification research, we selected several commonly used machine learning classification models for comparison.

5.4.1. DT. The DT is a process for classifying instances based on features, where each internal node represents a judgement on an attribute, each branch represents the output of a judgement result, and each leaf node represents a classification result. The algorithm loops all splits and selects the best-partitioned subtree based on the error rate and the cost of misclassification.

5.4.2. LR. The statistical technique of logistic regression is usually used to solve binary classification problems. Regression analysis is used to describe the relationship between the independent variable x and the dependent variable y and to predict the dependent variable y. LR adds a logistic function on the basis of regression.

5.4.3. MLP. It refers to neural principles, where each neuron can be regarded as a learning unit. The MLP is constructed on the basis of many neurons, which are composed of an input layer, hidden laver, and output layer. These neurons take certain characteristics as input and obtain output according to their own model. The weight assigned to each attribute varies according to its relative importance, and the weight is adjusted iteratively to make the predicted output closer to the actual target.

Table 4: Parameters of all algorithms in the experiments.

Algorithm	Parameters
DT	Min_samples_leaf = 6 Max_depth = 8 Min_samples_split = 2 Gamma = 0.0 Max_leaf_nodes = 4
LR	No parameters specified
MLP	Epoch = 1000 Learning_rate = 0.01 Hidden_units = 5
SVM	Kernel: RBF C = 32 Gamma = 0.1
RF	N_estimators = 100 Max_features = 12 Max_depth = 400 Min_samples_split = 2 Min_samples_leaf = 1

5.4.4. SVM. By mapping the feature vector of an instance to a point in space, the purpose of the SVM is to draw a line to best distinguish the two types of points. The SVM finds the hyperplane that separates the data. To best distinguish the data, the sum of the distances from the closest points on both sides of the hyperplane is required to be as large as possible.

The hyperparameter settings of the above baseline models are set according to the corresponding references in the article involved in the related work [30, 31]. The settings of the hyperparameter are shown in Table 4.

6. Results

We verify the effectiveness of the model on seven data sets (see Table 5).

On the data set ecoli-3,the RF model performs better than other types of models, and most of the indicators surpass other models.

RF have obtained good results, which shows that the ensemble model can pay more attention to the learning unbalanced data sets. Moreover, APSO-RF model reached the highest value on the F1-measure, 93.8%, which is 0.8% higher than NN.

On the data set glass-1, the results of APSO-RF are satisfactory for its all-evaluation criteria are better than those of other algorithms. Compared with the SVM, our model has improved ACC and F1-measure by 7.3% and 5.8%, respectively. The model with hyperparameters setting optimized by APSO has improved in all indicators, especially in ACC and F1-measure, compared to RF in these two indicators, 1.7% and 0.7%, respectively. This shows that the model can still deal with the problem of data imbalance well, while ensuring the level of overall accuracy.

On the new-thyroid-1 data set, the LR model performs better than RF. RF does not optimize the hyperparameter settings, which makes it insufficient to learn samples. RF performs better than LR on F1-measure,

Table 5: Results of the measured performance of models.

Data	Model	ACC	Precision	Recall	F1-measure	ROC area
ecoli-3	LR	0.922	0.917	0.922	0.910	0.910
	MLP	0.932	0.933	0.934	0.933	0.932
	DT	0.922	0.915	0.922	0.917	0.824
	SVM	0.896	0.802	0.896	0.846	0.502
	RF	0.934	0.929	0.934	0.932	0.936
	APSO-RF	0.938	0.931	0.942	0.941	0.939
glass-1	LR	0.648	0.621	0.648	0.618	0.681
	MLP	0.662	0.668	0.662	0.665	0.676
	DT	0.775	0.770	0.775	0.771	0.749
	SVM	0.793	0.794	0.793	0.784	0.743
	RF	0.836	0.837	0.836	0.824	0.896
	APSO-RF	0.851	0.841	0.838	0.830	0.902
new-thyroid-1	LR	0.986	0.986	0.986	0.986	0.997
	MLP	0.981	0.981	0.980	0.981	0.997
	DT	0.981	0.981	0.981	0.981	0.972
	SVM	0.879	0.894	0.879	0.847	0.629
	RF	0.972	0.972	0.972	0.971	0.998
	APSO-RF	0.988	0.987	0.986	0.982	0.998
page-blocks-0	LR	0.951	0.947	0.950	0.947	0.941
	MLP	0.968	0.967	0.968	0.967	0.978
	DT	0.986	0.986	0.986	0.986	0.991
	SVM	0.994	0.992	0.990	0.994	0.977
	RF	0.996	0.995	0.992	0.996	0.993
	APSO-RF	0.997	0.997	0.994	0.998	0.994
vehicle-1	LR	0.786	0.781	0.786	0.783	0.937
	MLP	0.842	0.838	0.840	0.839	0.918
	DT	0.717	0.714	0.717	0.716	0.830
	SVM	0.492	0.242	0.492	0.325	0.502
	RF	0.831	0.803	0.812	0.821	0.933
	APSO-RF	0.852	0.840	0.832	0.842	0.994
wisconsin	LR	0.965	0.965	0.965	0.965	0.992
	MLP	0.963	0.963	0.963	0.963	0.992
	DT	0.959	0.959	0.959	0.959	0.957
	SVM	0.960	0.964	0.960	0.961	0.968
	RF	0.969	0.969	0.969	0.969	0.993
	APSO-RF	0.971	0.975	0.974	0.975	0.944
veast-1	LR	0.757	0.740	0.757	0.728	0.790
	MLP	0.769	0.756	0.769	0.757	0.796
	DT	0.760	0.745	0.760	0.746	0.726
	SVM	0.721	0.713	0.721	0.626	0.526
	RF	0.778	0.767	0.778	0.769	0.806
	APSO-RF	0.782	0.774	0.782	0.770	0.821

indicating that RF uses the bagging method. It has good generalization ability. The key to adopting this method is to deal with imbalances to obtain effective classifiers while ensuring the diversity of base classifiers; the model APSO-RF optimized by hyperparameters has reached ACC and other indicators to the top; it shows that the improved particle swarm can help the model build a branch structure suitable for the data set, by selecting reasonable hyperparameter settings.

Most models on page-block-0 performed well, and on the evaluation indicators, APSO-RF algorithm was better than other algorithms. The model's performance in F1-measure ranks in the forefront, indicating that our model is superior to other algorithms in the classification performance of unbalanced data.

On the Wisconsin data set, the APSO-RF is better 0.2% than RF at ACC; APSO-RF is 0.6% higher in ACC than the third-highest model LR model, and the model has the best recall rate, indicating that the model can distinguish more positive categories.

On the veast-1, our model has achieved the best performance in all indicators, and it also performs well in the prediction accuracy and regression rate. A high accuracy rate means that the positive examples in the sample are more accurately predicted.

Positive examples in the sample are more accurately predicted. It shows that our proposed algorithm is superior to other algorithms in the classification performance of positive classes.

Overall, RF has better average performance than other models, which shows that this model can reduce the model

error effectively and achieve more accurate unbiased estimation with the help of integrated classification strategy. Specifically, traditional classification algorithms usually use classification accuracy as the evaluation criteria and aim to maximize the average accuracy. In order to maximize accuracy, they often sacrifice the performance of the minority class, while the RF uses an appropriate induction algorithm to benefit the minority Class classification learning. APSO-RF is improved obviously compared with RF in all index, which shows that hyperparameters can match the fitness value better, and its tree structure is more suitable for nonbalanced data, so the precision of the model is higher. The algorithm can improve the ability of positive classification obviously without losing the ability of global classification, because APSO is optimizing the hyperparameter reasonably. As a result, the tree structure that is more suitable for unbalanced data set is not built, and the performance is limited. Adaptive particle swarm optimization uses adaptive group division and different updating strategies to guide particles learning, which helps maintain the diversity of the population and avoid the model falling into local optimum early.

7. Conclusions

Unbalanced data classification is a big challenge in the field of data mining. RF, as an ensemble learning method, is usually used to solve the problem of unbalanced data classification. This paper proposes a particle swarm optimization strategy based on adaptive partitioning, which uses the good global and local search performance of the optimization strategy to optimize the hyperparameters of the RF and optimizes the misclassification of samples in the imbalanced data classification problem. The proposed model is verified on six nonequilibrium data sets and gets good prediction results. The result demonstrates that the model has excellent generalization ability and the ability to deal with nonequilibrium data sets.

Besides the PSO algorithm, we also study other swarm-based metaheuristic search methods and carry out in-depth research such as monarch butterfly optimization (MBO) [39], earthworm optimization algorithm (EWA) [40], elephant herding optimization (EHO) [41], moth search (MS) algorithm [42], slime mould algorithm (SMA) [43], and Harris hawks optimization (HHO) [44].

The MBO is proposed by idealizing the migration of monarch butterflies, and the positions of the monarch butterflies are updated in two ways. Firstly, the offspring is generated (position updating) by the migration operator that can be adjusted by the migration ratio. Secondly, tune the positions of other butterflies by butterfly adjusting operator. The EWA method is inspired by the two kinds of reproduction of the earthworms that generate only one offspring by themselves or generates one or more than one offspring at one time. The method successfully realized the above functions by nine improved crossover operators. EHO idealizes the behavior of elephant herding into clan updating operator and separating operator for solving global optimization tasks. MS corresponds to two features, the phototaxis and Lévy flights of the moths to the development and exploration of metaheuristic optimization methods. SMA uses adaptive weights to simulate the process of producing positive and negative feedback of the propagation wave of slime mould to form the optimal path for connecting food. The main inspiration of HHO is the cooperative behavior and chasing style of Harris hawks in nature. In this strategy, several hawks cooperatively pounce a prey from different directions. The algorithm works mathematically mimics chasing patterns based on the dynamic nature of scenarios and escaping patterns of the prey.

The above research result provides us with more thoughts for improving the swarm search strategy. We plan to use two kinds of algorithms to make them cooperate and optimize in a relatively simple way and study how to make them cooperate more efficiently through other methods.

Data Availability

Publicly available datasets were analyzed in this study. These data can be found at https://sci2s.ugr.es/keel/imbalanced.php.

Conflicts of Interest

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

Acknowledgments

This research was funded by the National Natural Science Foundation of China under Grant numbers 61972227, 61873117, and U1609218; in part by the Natural Science Foundation of Shandong Province under Grant numbers ZR201808160102 and ZR2019MF051; in part by the Primary Research and Development Plan of Shandong Province under Grant numbers GG201710090122, 2017GGX10109, 2018GGX101013, 2020CXGC010110; in part by the Key Research and Development Project of Shandong Province, under Grant numbers 2020cxgc010110 and ZR2020KF015; in part by the Independent Training and Innovation Team of Shandong Province under Grant number 2020gxrc016.and in part by the Fostering Project of Dominant Discipline and Talent Team of Shandong Province Higher Education Institutions.

References

[1] H. He and Y. Fan, "A novel hybrid ensemble model based on tree-based method and deep learning method for default prediction," *Expert Systems with Applications*, vol. 176, no. 4, Article ID 114899, 2021.

[2] H. Zhang, J.-L. Li, X.-M. Liu, and C. Dong, "Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection," *Future Generation Computer Systems*, vol. 122, pp. 130–143, 2021.

[3] S. B. Abkenar, E. Mahdipour, S. M. Jameii, and M. H. Kashani, "A hybrid classification method for Twitter spam detection based on differential evolution and random forest," *Concurrency and Computation: Practice and Experience*, no. 7, 2021.

[4] H. Li, A. Feng, B. Lin et al., "A novel method for credit scoring based on feature transformation and ensemble model," *PeerJ Computer Science*, vol. 7, p. e579, 2021.

[5] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification OF imbalanced data: a review," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 4, pp. 687–719, 2009.

[6] M. Janicka, M. Lango, and J. Stefanowski, "Using information on class interrelations to improve classification of multiclass imbalanced data: a new resampling algorithm," *International Journal of Applied Mathematics and Computer Science*, vol. 29, no. 4, pp. 769–781, 2019.

[7] W. Pei, B. Xue, L. Shang, and M. Zhang, "Genetic programming for development of cost-sensitive classifiers for binary high-dimensional unbalanced classification," *Applied Soft Computing*, vol. 101, Article ID 106989, 2021.

[8] 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.

[9] T. G. Dietterich, "Ensemble methods in machine learning," *Proc International Workshop on Multiple Classifier Systems*, p. 1857, 2000.

[10] M. J. Kearns and L. G. Valiant, "Cryptographic limitations on learning Boolean formulae and finite automata," *Machine Learning: From Theory to Applications*, vol. 661, pp. 29–49, 1993.

[11] L. Breiman, "Random forest," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] A. Alhudhaif, "A novel multi-class imbalanced EEG signals classification based on the adaptive synthetic sampling (ADASYN) approach," *Peer Journal Computer Science*, vol. 7, p. e523, 2021.

[13] W. Li, G.-G. Wang, and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3781–3799, 2021.

[14] J. Li, H. Lei, A. H. Alavi, and G.-G. Wang, "Elephant herding optimization: variants, hybrids, and applications," *Mathematics*, vol. 8, no. 9, p. 1415, 2020.

[15] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and D. Gong, "A comprehensive review of krill herd algorithm: variants, hybrids and applications," *Artificial Intelligence Review*, vol. 51, no. 1, pp. 119–148, 2019.

[16] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: a comprehensive review," *Expert Systems with Applications*, vol. 168, Article ID 114418, 2021.

[17] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: theory, analysis, and application," *Swarm and Evolutionary Computation*, vol. 63, Article ID 100868, 2021.

[18] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Mhs95 Sixth International Symposium on Micro Machine & Human Science IEEE*, pp. 39–43, Nagoya, Japan, October 2002.

[19] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Z. Gao, "A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–14, 2021.

[20] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, p. 1, 2021.

[21] Y. Zhang, D.-W. Gong, and Z. Ding, "A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch," *Information Sciences*, vol. 192, pp. 213–227, 2012.

[22] Y. Qiang, W. N. Chen, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, pp. 578–594, 2018.

[23] M. A. H. Farquad and I. Bose, "Preprocessing unbalanced data using support vector machine," *Decision Support System*, vol. 53, no. 1, pp. 226–233, 2012.

[24] K. Wang, J. Tian, C. Zheng et al., "Improving risk identification of adverse outcomes in chronic heart failure using SMOTE + ENN and machine learning," *Risk Management and Healthcare Policy*, vol. 14, pp. 2453–2463, 2021.

[25] G. Wang, L. Guo, and D. Hong, "Wavelet neural network using multiple wavelet functions in target threat assessment," *The Scientific World Journal*, vol. 2013, Article ID 632437, 2013.

[26] J.-H. Yi, J. Wang, and G.-G. Wang, "Improved probabilistic neural networks with self-adaptive strategies for transformer fault diagnosis problem," *Advances in Mechanical Engineering*, vol. 8, no. 1, pp. 1–13, 2016.

[27] G.-G. Wang, M. Lu, Y.-Q. Dong, and X.-J. Zhao, "Self-adaptive extreme learning machine," *Neural Computing & Applications*, vol. 27, no. 2, pp. 291–303, 2015.

[28] Z. Cui, F. Xue, X. Cai, Y. Cao, G.-G. Wang, and J. Chen, "Detection of malicious code variants based on deep learning," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3187–3196, 2018.

[29] G.-G. Wang, L.-H. Duan, H. Duan, L. Liu, and H.-Q. Wang, "Target threat assessment model and algorithm based on Elman_Adaboost Strong predictor," *Acta Electronica Sinica*, vol. 40, no. 5, pp. 901–906, 2012.

[30] Z. Chatrsimab, A. A. Alesheikh, B. Voosoghi, S. Behzadi, and M. Modiri, "Development of a land subsidence forecasting model using small baseline subset-differential synthetic aperture radar interferometry and particle swarm optimization-random forest (case study: tehran-karaj-shahriyar aquifer, Iran)," *Doklady Earth Sciences*, vol. 494, no. 1, pp. 718–725, 2020.

[31] Y. W. Wan and B. Zhu, "Abnormal patterns recognition in bivariate autocorrelated process using optimized random forest and multi-feature extraction - ScienceDirect," *ISA Transactions*, vol. 109, pp. 102–112, 2020.

[32] G.-G. Wang, A. Hossein Gandomi, X.-S. Yang, and A. Hossein Alavi, "A novel improved accelerated particle swarm optimization algorithm for global numerical optimization," *Engineering Computations*, vol. 31, no. 7, pp. 1198–1220, 2014.

[33] G.-G. Wang, A. Hossein Gandomi, and A. Hossein Alavi, "A chaotic particle-swarm krill herd algorithm for global numerical optimization," *Kybernetes*, vol. 42, no. 6, pp. 962–978, 2013.

[34] C.-L. Wei and G.-G. Wang, "Hybrid annealing krill herd and quantum-behaved particle swarm optimization," *Mathematics*, vol. 8, 2020.

[35] S. Mirjalili, G. G. Wang, and L. S. Coelho, "Binary optimization using hybrid particle swarm optimization and gravitational search algorithm," *Neural Comput & Application*, vol. 25, 2014.

[36] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and S. Deb, "A hybrid method based on krill herd and quantum-behaved

particle swarm optimization," *Neural Computing & Applications*, vol. 27, no. 4, pp. 989–1006, 2016.

[37] D.-X. Zou, S. Deb, and G.-G. Wang, "Solving IIR system identification by a variant of particle swarm optimization," *Neural Computing & Applications*, vol. 30, no. 3, pp. 685–698, 2018.

[38] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[39] G.-G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing & Applications*, vol. 31, no. 7, pp. 1995–2014, 2015.

[40] G. G. Wang, S. Deb, and L. D. S. Coelho, "Earthworm optimisation algorithm: a bio-inspired metaheuristic algorithm for global optimisation problems," *International Journal of Bio-Inspired Computation*, vol. 12, no. 1, p. 1, 2018.

[41] G. G. Wang, S. Deb, and L. Coelho, "Elephant Herding Optimization," in *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI 2015)*, Bali, Indonesia, December 2015.

[42] G. G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, 2018.

[43] 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.

[44] A. Asghar 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.