Research Article

Semi-Supervised Ensemble Classifier with Improved Sparrow Search Algorithm and Its Application in Pulmonary Nodule Detection

Jiangnan Zhang [ID], Kewen Xia [ID], Ziping He, Zhixian Yin, and Sijie Wang

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

Correspondence should be addressed to Kewen Xia; kwxia@hebut.edu.cn

The Adaptive Boosting (AdaBoost) classifier is a widely used ensemble learning framework, and it can get good classification results on general datasets. However, it is challenging to apply the AdaBoost classifier directly to pulmonary nodule detection of labeled and unlabeled lung CT images since there are still some drawbacks to ensemble learning method. Therefore, to solve the labeled and unlabeled data classification problem, the semi-supervised AdaBoost classifier using an improved sparrow search algorithm (AdaBoost-ISSA-S4VM) was established. Firstly, AdaBoost classifier is used to construct a strong semi-supervised classifier using several weak classifiers S4VM (AdaBoost-S4VM). Next, in order to solve the accuracy problem of AdaBoost-S4VM, sparrow search algorithm (SSA) is introduced in the AdaBoost classifier and S4VM. Then, sine cosine algorithm and new labor cooperation structure are adopted to increase the global optimal solution and convergence performance of sparrow search algorithm, respectively. Furthermore, based on the improved sparrow search algorithm and adaptive boosting classifier, the AdaBoost-S4VM classifier is improved. Finally, the effective improved AdaBoost-ISSA-S4VM classification model was developed for actual pulmonary nodule detection based on the publicly available LIDC-IDRI database. The experimental results have proved that the established AdaBoost-ISSA-S4VM classification model has good performance on labeled and unlabeled lung CT images.

1. Introduction

Pulmonary nodule detection belongs to the category of classification. The pulmonary nodule detection based on lung CT images is the key to diagnose lung cancer for doctors. In the field of actual lung CT image recognition, classification accuracy and acquisition of lung CT image labels are crucial issues. To solve the classification accuracy problem, ensemble learning is introduced. After 30 years of development, ensemble learning has been applied in many fields of machine learning [1, 2] and is considered to be one of the effective ways to improve classification accuracy problem. In 1996, Breiman proposed the Bagging algorithm [3] which is similar to the Boosting algorithm [4]. This algorithm is one of the many algorithm families in the field of ensemble learning. But, the Boosting algorithm is difficult to apply in practical problems, because that it must know the generalization lower bound of the "weak" learning algorithm. To solve this problem, Freund and Schapire proposed the famous adaptive boosting (AdaBoost) algorithm in 1997 [5]. Compared with the Boosting algorithm, this algorithm has stronger robustness and applicability and further promotes the development of ensemble learning. In many researches, the combination of AdaBoost classifier is optimized by the method of "weak" classifier, such as support vector machine (SVM) method [6] and long short-term memory (LSTM) network [7].

Although AdaBoost classifier has been successfully applied in many fields with its competitive accuracy, the AdaBoost classifier in classification problem of missing part of lung CT image labels show the main weakness: when dealing with labeled and unlabeled lung CT images, the classification cannot be classified well alone. To solve this problem, a hybrid ensemble learning approach is proposed

for pulmonary nodule detection combining AdaBoost algorithm and safe semi-supervised support vector machine (S4VM) [8]. In addition, its performance is also mainly affected by the key parameters in the model. The corresponding weight of each weak classifier β needs to be optimized in AdaBoost-S4VM. And the S4VM optimization involves two main hyper-parameters: regularization trading off the complexity parameter C_1 and the empirical error on label and unlabeled data parameter C_2. The parameter of S4VM is usually optimized by the 10-fold cross-validation method that cannot adapt to the data automatically and set the parameter range difficultly. And, it is easy to fall into the local optimum. In order to overcome the above shortcomings, many researches have proposed the use of intelligent optimization algorithms to optimize the key parameters of the S4VM model. These algorithms include quasi-Newton algorithms [9], cuckoo search algorithm (CS) [10], and beetle antennae search (BAS) [11]. For the advantages of sparrow search algorithm (SSA) [12], such as simple principle, strong mining capacity, and few adjustable parameters, the improved SSA that improves the classification performance of AdaBoost-S4VM is used to optimize the key parameters of AdaBoost-S4VM and S4VM in this research. As well as, hybrid strategy which is one of the main research directions to improve the performance of swarm intelligence algorithms has become a research hotspot in machine learning. Rasoulizadeh et al. [13] modified local RBF-generated finite difference method (RBF-FD) based on local stencil nodes which has a sparsity system to overcame the dense and ill-condition. Rashidinia et al. [14] also has proposed the two meshless collocation methods based on radial basis function-generated finite difference (RBF-FD) and global RBF(GRBF) methods, and the simulation results have shown that the proposed approach was viable and effective. Can et al. [15] modified the idea of the interpolation by radial basis function, and the obtained results show that the proposed method is able to provide valid and accurate results and outperform other counterparts.

In this study, in view of the shortcomings of the AdaBoost algorithm, the S4VM weak classifier is introduced. Then, SSA is used to optimize the parameters of AdaBoost-S4VM. But, it has disadvantages such as easy to fall into local optimum and poor performance in solving complex optimization problems. After that, because the sine cosine algorithm (SCA) [16] has the characteristics of achieving high search and avoiding local optimization, we first introduced the SCA algorithm to improve the global search capability of the SSA algorithm. Additionally, in order to enhance the convergence ability of the SSA algorithm, the labor cooperation structure of the sparrow in the SSA algorithm is redefined. Finally, based on the new labor cooperation structure and SCA algorithm, the improved cooperative sparrow search algorithm based on sine cosine algorithm (SCA-CSSA) is proposed. The SCA-CSSA algorithm is used to optimize the weight of AdaBoost-S4VM and the key parameters of S4VM to improve the accuracy of the AdaBoost-S4VM model for semi-supervised lung CT classification. And, the improved semi-supervised AdaBoost classification model using an improved sparrow search algorithm (AdaBoost-ISSA-S4VM) was established. In order to

verify the effectiveness of the proposed AdaBoost-ISSA-S4VM model, first it compared with several hybrid algorithms and popular algorithms on CEC2017 functions and 12 benchmark functions, including unimodal and multimodal functions. In addition, in order to evaluate the effectiveness of the AdaBoost-ISSA-S4VM model, it is also compared with the supervised classifiers and semi-supervised classifiers. Experimental results show that the improved machine learning model proposed has better stability and higher prediction accuracy.

2. The Basic Method

2.1. Adaptive Boosting (AdaBoost). The AdaBoost classifier algorithm is implemented by changing the data distribution. It determines the weight of each sample based on whether the classification of each sample in each training dataset is correct and the accuracy of the last overall classification accuracy. It sends the new dataset with modified weights to the lower classifier for training, and so on. Finally, according to the calculated corresponding weights, we will get the final desired "strong classifier" named AdaBoost-S4VM by stacking a series of "weak classifiers." The strong learning algorithm is defined as follows:

$$G(x) = \text{sign}\left(\sum_{m=1}^{M} \beta_m g_m(x)\right), \tag{1}$$

where $g_m(x)$ is the basic weak classifier, β_m $(m = 1, \ldots, M)$ is the corresponding weight of each weak classifier, and M is the total number of basic weak classifiers.

The calculation method of the weight coefficient β_m $(m = 1, \ldots, M)$ of $g_m(x)$ is

$$\beta_m = \frac{1}{2}\log\frac{1 - e_m}{e_m}, \tag{2}$$

where e_m is the error rate of calculating g_m on the training set, that is, $e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^{N} w_{mi}I(G_m(x_i) \neq y_i)$.

Since we explore the discrimination of features by training weak classifiers and organize the AdaBoost classifiers in a cascade way, we ask for simple weak classifiers, with which the target of the cascade-AdaBoost classifier can be easily controlled. Thus, simple threshold classifiers are chosen as weak classifiers as follows:

$$g_m(x) = \begin{cases} 1, & T_{\text{lower}} \leq x \leq T_{\text{upper}}, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where T_{lower} and T_{upper} are thresholds for the weak classifier g_m, which can be obtained by using a semi-exhaustive search technique.

2.2. Safe Semi-Supervised Support Vector Machines (S4VM). The core design idea of S4VM is that it optimizes the classification of unlabeled sample data when there are many different situations that meet the larger "interval" dividing line. So that the performance improvement relative to the support vector machine that only uses labeled samples is

maximized in the worst case. And, the objective function $h(f, \hat{y})$ that S4VM needs to optimize is as follows:

$$h(f, \hat{y}) = \frac{\|f\|_H}{2} + C_1 \sum_{i=1}^{l} l(y_i, f(x_i)) + C_2 \sum_{j=1}^{u} l(\hat{y}_j, f(\hat{x}_j)). \tag{4}$$

Its goal is to find multiple large-margin low-density separators $\{f_t\}_{t=1}^{T}$ and the corresponding label assignments $\{\hat{y}_t\}_{t=1}^{T}$ such that the following functional is minimized:

$$\min_{\{f_t, \hat{y}_t \in B\}_{t=1}^{T}} \sum_{t=1}^{T} h(f_t, \hat{y}_t) + M\Omega(\{\hat{y}_t\}_{t=1}^{T}), \tag{5}$$

where T is the number of separators, Ω is a quantity of penalty about the diversity of separators, and M is a large constant enforcing large diversity.

And, $\Omega(\{\hat{y}_t\}_{t=1}^{T})$ is as sum of pairwise terms. Here it is defined as $\Omega(\{\hat{y}_t\}_{t=1}^{T}) = \sum_{1 \le t \ne \tilde{t} \le T} I((\hat{y}_t' \hat{y}_{\tilde{t}}/u) \ge 1 - \varepsilon)$, where I is the identity function and $\varepsilon \in [0, 1]$ is a constant, but note that other penalty quantities are also applicable. Without loss of generality, suppose that f is a linear model, and it is defined as $f(x) = w' \phi(x) + b$ Where $\phi(x)$ is a feature mapping induced by the kernel k. Then, the optimization problem to be solved can be expressed as follows:

$$\min_{\{w_t, b_t, \hat{y}_t \in B\}_{t=1}^{T}} \sum_{t=1}^{T} \left(\frac{1}{2} \|w_t\|^2 + C_1 \sum_{i=1}^{l} \xi_i + C_2 \sum_{j=1}^{u} \hat{\xi}_j \right) + M \sum_{1 \le t \ne \tilde{t} \le T} I\left(\frac{\hat{y}_t' \hat{y}_{\tilde{t}}}{u} \ge 1 - \varepsilon \right),$$

$$\text{s.t} \quad \begin{aligned} y_i \left(w_t' \phi(x_i) + b_t \right) \ge 1 - \xi_i, \quad & \xi_i > 0, \\ \hat{y}_{t,j} \left(w_t' \phi(\hat{x}_j) + b_t \right) \ge 1 - \hat{\xi}_j, \quad & \hat{\xi}_j > 0, \end{aligned} \quad \forall i = 1, \dots, l, \forall j = 1, \dots, u, \forall t = 1, \dots, T, \tag{6}$$

where $\hat{y}_{t,j}$ refers to the jth entry of $\hat{y}_t$. Formula (6) is nonconvex, and in the following, we will present two solutions. It is evident that this can also be implemented by other solutions, especially those based on efficient S3VMs.

2.3. Sparrow Search Algorithm (SSA).

Sparrow search algorithm (SSA) was originally proposed by Xue et al. The algorithm imitates the unique predation method of sparrows in nature to solve the optimization problem. In SSA, the position of the sparrow in the population is the candidate solution for a given optimization problem. According to the mathematical model of SSA, the behavior of the sparrows is mainly divided into three divisions of labor: producers, scroungers and sparrows at the edge of the group.

According to the rules of producers and once the sparrow detects the predator, the producers can search for food in a broad range of the places than that of the scroungers. The location of the producer is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\frac{-i}{\alpha \cdot \text{iter}_{\max}} \right), & R_2 < \text{ST}, \\ X_{i,j}^{t} + Q \cdot L, & R_2 \ge \text{ST}, \end{cases} \tag{7}$$

where t indicates the current iteration, $j = 1, 2, \dots, d$. $X_{i,j}^{t+1}$ represents the value of the jth dimension of the ith sparrow at iteration t. $\text{iter}_{\max}$ is a constant with the largest number of iterations, $\alpha \in (0, 1]$ is a random number, R_2 ($R_2 \in [0, 1]$) and ST (ST $\in [0.5, 1.0]$) represent the alarm value and the safety threshold, respectively. Q is a random number which

obeys normal distribution, and L shows a matrix of $1 \times d$ for which each element inside is 1.

According to rules of producers and the scroungers, the position update formula for the scrounger is described as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}}^{t} - X_{i,j}^{t}}{i^2} \right), & i \ge \frac{n}{2}, \\ X_P^{t+1} + \left| X_{i,j}^{t} - X_P^{t+1} \right| \cdot A^+ \cdot L, & \text{otherwise}, \end{cases} \tag{8}$$

where X_P is the optimal position occupied by the producer, X_{worst} denotes the current global worst location, A represents a matrix of $1 \times d$ for which each element inside is randomly assigned 1 or -1, and $A^+ = A^T (AA^T)^{-1}$. When $i > n/2$, it suggests that the ith scrounger with the worse fitness value is most likely to be starving.

According to rule of the sparrows at the edge of the group, the mathematical model of the sparrows at the edge of the group can be expressed as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \cdot \left| X_{i,j}^{t} - X_{\text{best}}^{t} \right|, & f_i > f_g, \\ X_{i,j}^{t} + K \cdot \left(\frac{\left| X_{i,j}^{t} - X_{\text{worst}}^{t} \right|}{(f_i - f_w) + \varepsilon} \right), & f_i = f_g, \end{cases} \tag{9}$$

where X_{best} is the current global optimal location. β, as the step size control parameter, is a normal distribution of random numbers with a mean value of 0 and a variance of 1, and $K \in [0, 1]$ is a random number. Here, f_i is the fitness value of the present sparrow, f_g and f_w are the current

global best and worst fitness values, respectively, and ε is the smallest constant so as to avoid zero-division-error.

3. The Proposed Method

The sparrow search algorithm and the principle of AdaBoost classifier have been clarified, and the basic S4VM has also been discussed. However, the pulmonary nodule detection classifying process based on lung CT images is complex and challenging.

Although the AdaBoost classifier is novel and superior, there are still some shortcomings when utilized to the pulmonary nodule detection classifying problem based on lung CT images. Thus, two improving strategies called "S4VM algorithm" and "parameters optimization" or "weight optimization" are introduced to the original Ada-Boost classifier and AdaBoost-S4VM, respectively, to help them jump out of local optima.

There are also some shortcomings in SSA to optimize them, so a sine cosine algorithm is used as hybrid algorithm to help it jump out of local optima. And, a new division of labor structure is introduced to the original SSA to help it converge to the global optimal solution faster and more stably. In this section, the proposed SCA-CSSA algorithm, AdaBoost-S4VM algorithm, and AdaBoost-ISSA-S4VM algorithm will be discussed in detail.

3.1. The Proposed AdaBoost-S4VM Model (AdaBoost-S4VM). The AdaBoost classifier algorithm is implemented by changing the data distribution. It determines the weight of each sample based on whether the classification of each sample in each training dataset is correct and the accuracy of the last overall classification accuracy. It sends the new dataset with modified weights to the lower classifier for training and so on. Finally, according to the calculated corresponding weights, we will get the final desired "strong classifier" named AdaBoost-S4VM by stacking a series of "weak classifiers." The strong learning algorithm is defined as follows:

$$G(x) = \text{sign}\left(\sum_{m=1}^{M} \beta_m g_m(x)\right), \qquad (10)$$

where $g_m(x)$ is the basic weak classifier, β_m $(m = 1, \ldots, M)$ is the corresponding weight of each weak classifier and it is defined as $\sum_{m=1}^{M} \beta_m = 1$, $0 \le \beta_m \le 1$, and M is the total number of basic weak classifiers.

According to the characteristic of formulas (6) and (10), we can get the improved AdaBoost-S4VM formula combined with "weak" classifier S4VM as follows:

$$G(x) = \sum_{m=1}^{M} \beta_m \left\{ \min_{\{w_t, b_t, \widehat{y}_t \in B\}_{t=1}^{T}} \sum_{t=1}^{T} \left(\frac{1}{2}\|w_t\|^2 + C_1 \sum_{i=1}^{l} \xi_i + C_2 \sum_{j=1}^{u} \widehat{\xi}_j\right) + M \sum_{1 \le t \ne \widetilde{t} \le T} I\left(\frac{\widehat{y}_t' \widehat{y}_{\widetilde{t}}}{u} \ge 1 - \varepsilon\right) \right\}, \qquad (11)$$

where β_m $(m = 1, \ldots, M)$ is the corresponding weight of each S4VM classifier and it is defined as $\sum_{m=1}^{M} \beta_m = 1$, $0 \le \beta_m \le 1$ and M is the total number of basic S4VM classifiers.

3.2. The Proposed Sparrow Search Algorithm (SCA-CSSA). It can be seen from Section 3.1 that, in order to obtain the optimal classification effect and efficiency, parameters C_1, C_2, and weight β_m $(m = 1, \ldots, M)$ need to be optimized. But, due to the disadvantages of being easy to fall into the local optimum and slow convergence speed of the original SSA, it is difficult to guarantee the quality of the obtained solution. In response to these problems, SCA algorithm and a new labor cooperation structure is used to improve the global and local search capability of SSA algorithm.

The sine cosine algorithm (SCA) position can be written mathematically as follows:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \cdot \sin(r_2) \cdot |r_3 P_i^t - X_i^t|, & r_4 < 0.5, \\ X_i^t + r_1 \cdot \cos(r_2) \cdot |r_3 P_i^t - X_i^t|, & r_4 \ge 0.5, \end{cases} \qquad (12)$$

where X_i^t is the position of the current solution in i-th dimension at t-th iteration, r_2/r_3 are random numbers, P_i is position of the destination point in i-th dimension, $||$ indicates the absolute value, and r_4 is a random number in $[0, 1]$.

In order to balance exploration and exploitation, the range of sine and cosine in formula (12) is changed adaptively using the following equation:

$$r_1 = a - t\frac{a}{T}, \qquad (13)$$

where t is the current iteration, T is the maximum number of iterations, and a is a constant.

To increase the search speed and jump out of local optimization, the SCA algorithm is introduced in SSA. Because SCA has the characteristics of increasing the search speed and jumping out of local optimization, it can well avoid the problem of premature sparrows. The following formula is the update SSA formula that combines the SCA algorithm:

$$X_{i,j}^{\text{SSA}} = X_{i,j}^{t+1} \oplus \text{SCA}(\alpha), \qquad (14)$$

$$X_i^{t+1} = \begin{cases} X_i^t, & \text{fitness}\left(X_i^{\text{SSA}}\right) > \text{fitness}\left(X_i^t\right), \\ X_i^{\text{SSA}}, & \text{fitness}\left(X_i^{\text{SSA}}\right) \le \text{fitness}\left(X_i^t\right). \end{cases} \qquad (15)$$

Then, in the improved SSA algorithm, a new labor cooperation structure is first used to converge to the global optimal solution faster and more stably. In the new labor cooperation structure, three divisions of sparrows are also

defined: producer, scrounger, and sparrows at the edge of the group. Since the producer and scrounger determine the location range and convergence performance of the group, they share their locations to achieve cooperation. Then, cooperation could make both producer and scrounger great, thereby achieving the effect of improving convergence. The location of the producer is remarked as follows:

$$X_1 = X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\dfrac{-i}{\alpha \cdot \mathrm{iter}_{\max}}\right), & R_2 < ST, \\ \\ X_{i,j}^t + Q \cdot L, & R_2 \geq ST. \end{cases} \quad (16)$$

The position remarked formula for the scrounger is described as follows:

$$X_2 = X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{\mathrm{worst}}^t - X_{i,j}^t}{i^2}\right), & i \geq \dfrac{n}{2}, \\ \\ X_P^{t+1} + \left| X_{i,j}^t - X_P^{t+1} \right| \cdot A^+ \cdot L, & \text{otherwise,} \end{cases} \quad (17)$$

This process can be expressed by the following formula:

$$X_{i,j}^{\mathrm{cooperative}} = \frac{w_1 X_1 + w_2 X_2}{2}, \quad (18)$$

$$w_1, w_2 = \begin{cases} w_1 = 2, w_2 = 0, & \mathrm{fitness}(X_1) = \min_{\mathrm{best}}, \\ w_1 = 0, w_2 = 2, & \mathrm{fitness}(X_2) = \min_{\mathrm{best}}, \\ w_1 = 1, w_2 = 1, & \mathrm{fitness}(X_1) \neq \min_{\mathrm{best}} \& \mathrm{fitness}(X_2) \neq \min_{\mathrm{best}}, \end{cases} \quad (19)$$

where $\min_{\mathrm{best}}$ refers to the global optimal minimum solution.

$$X_i^t = \begin{cases} X_i^{\mathrm{cooperative}}, & \mathrm{fitness}\left(X_i^{\mathrm{cooperative}}\right) > \mathrm{fitness}\left(X_i^t\right), \\ X_i^t, & \mathrm{fitness}\left(X_i^{\mathrm{cooperative}}\right) \leq \mathrm{fitness}\left(X_i^t\right). \end{cases} \quad (20)$$

The pseudocode of the whole SCA-CSSA is given below in Algorithm 1.

3.3. The Proposed AdaBoost-S4VM Model Improved by the Improved Sparrow Search Algorithm (AdaBoost-ISSA-S4VM).

After getting the improved SCA-CSSA, it can be seen from Section 3.1 that, in order to obtain the optimal classification effect and efficiency, parameters C_1 and C_2 and weight β_m ($m = 1, \ldots, M$) need to be optimized.

Finally, the AdaBoost-S4VM parameter is optimized using SCA-CSSA. And, the pseudocode of AdaBoost-ISSA-S4VM is shown in Algorithm 2.

4. Experimental Studies

In this section, in order to evaluate the performance of the proposed SCA-CSSA and AdaBoost-ISSA-S4VM model, a series of experiments on test functions and CT images are used in this paper. All experiments in this paper are implemented using the following: MATLAB R2014b; Win 10 (64 bit); Inter (R) Core (TM) i5-10210M CPU @1.60 GHz 2.11 GHz.

4.1. Function Optimization Experiment.
This section presents the evaluation of SCA-CSSA using a series of experiments on benchmark functions [17] and CEC2017 test functions [18]. To obtain fair results, all the experiments were conducted under the same conditions. The number of the population size is set as 30 in these algorithms. And, each algorithm runs 30 times independently for each function.

4.1.1. Benchmark Functions and CEC 2017 Test Functions.
When investigating the effective and universal performance of SCA-CSSA compared with several hybrid algorithms and popular algorithms, 12 benchmark functions and CEC2017 test functions are applied. In order to test the effectiveness of the proposed SCA-CSSA, 12 benchmark functions are adopted, all of which have an optimal value of 0. The benchmark functions and their searching ranges are shown in Table 1. In this test suite, $f_1 - f_7$ are unimodal functions. These unimodal functions are usually used to test and investigate whether the proposed algorithm has a good convergence performance. Then, $f_8 - f_{12}$ are multimodal functions. These multimodal functions are used to test the global search capability of the proposed algorithm. The smaller the fitness value of functions, the better the algorithm performs. Furthermore, in order to better verify the comprehensive performance of SCA-CSSA in a more comprehensively manner, another 30 complex CEC2017 test functions are used. The CEC2017 test functions are simply described in Table 2.

4.1.2. Parameter Settings.
In order to verify the effectiveness and generalization of the proposed SCA-CSSA, the improved SCA-CSSA is compared with several hybrid algorithms and popular algorithms. These algorithms are SSA, SCA, SCA_SSA, and SCA_CSSA. Another 4 popular intelligence algorithms, such as particle swarm optimization (PSO) [19], bird swarm algorithm (BSA) [20], crow search algorithm (CSA) [21], whale optimization algorithm (WOA) [22], grasshopper optimization algorithm (GOA) [23], are used to compare with SCA-CSSA. These algorithms represented state-of-the-art can be used to better verify the performance of SCA-CSSA, in a more comprehensively manner. For fair comparison, the number of populations of all algorithms is set to 30, respectively, and other parameters of all algorithms are set according to their original papers. The initial controlling parameters of all algorithms are shown in Table 3.

4.1.3. Comparison on Benchmark Functions with Hybrid Algorithms and Popular Algorithms.
According to the Section 2.2, the basic SSA method has been improved by two strategies. To investigate the effectiveness of SCA_CSSA, it has been compared with several popular algorithms and hybrid algorithms, such as PSO, BSA, CSA, WOA, GOA,

Input: the maximum iterations: iter_max; the number of producers: P_Num; the number of sparrows who perceive the danger: SD_Num; the number of sparrows: pop; dynamic parameter: a;
Output: global optimal position: X_{best}; fitness of global optimal position: f_g;
Begin:
(1) Initialize a population of sparrows pop and define its relevant parameters.
(2) **while** ($t <$ iter_max)
(3) Rank the fitness values and find the current best individual and the current worst individual.
(4) $R_2 = \text{rand}(1)$
(5)   **for** $i = 1$: pop
(6)   Using formula (16), update the sparrow's location;
(7)   **end for**
(8)   **for** $i = 1$: pop
(9)   Using formula (17), update the sparrow's location;
(10)   **end for**
    / * the new division of labor structure scheme * /
(11) Using formulas (18) and (19), update the producer and scrounger's cooperative location;
(12) If the new location is better than before, update it use formula (20);
(13)   **for** $l = 1$: SD_Num
(14)   Using formula (9) update the sparrow's location;
(15)   **end for**
(16) Get the current new location;
(17)   If the new location is better than before, update it;
    / * sine cosine algorithm scheme * /
(18) Using formula (14), update the SCA sparrow's location;
(19)   If the new location is better than before, update it use formula (15);
(20) $t = t + 1$
(21) **end while**
(22) **return** X_{best}, f_g.

ALGORITHM 1: The framework of the SCA-CSSA.

Input: weak classifier type: S4VM; train data set, train label set, test data set, test label set; the maximum iterations: M; kernel: RBF; parameters of S4VM: weight for the hinge loss of labeled instance C_1, weight for the hinge loss of unlabeled instance C_2, and the sampling times for each trial sampleTime.
Output: prediction label of test data set
(1) set the weights of the training data set $D_1 = (w_{11}, \ldots, w_{1i}, \ldots, w_{1N})$, $w_{1i} = 1/N$;
(2) **for** $m = 1$: M
(3)   **If** there are misclassification points
    / * Parameter selection based on SCA-CSSA * /
(4)   According to SCA-CSSA, find the optimal hyper-parameters (C_1, C_2) of weak classifier S4VM;
    / * **Weight of AdaBoost** selection based on SCA-CSSA * /
(5)   According to SCA-CSSA, find the optimal weight β_m ($m = 1, \ldots, M$) of weak classifier S4VM;
(6)   Using the weight distribution β_m, calculate the m_{th} weak classifier G_m;
(7)   Update the weight distribution of the training set $w_{m+1,i}$;
(8) $m = m + 1$;
(9) **else**
(10)   jump out of the loop;
(11)   **end**
(12) **end for**
(13) According to formula (11), m groups of weak classifiers are linearly combined, and the final classifier is output;
(14) Use the final classifier to predict the training set classification.

ALGORITHM 2: AdaBoost-ISSA-S4VM classification model algorithm.

SSA, SCA, and SCA-SSA, on 12 benchmark functions. Compared with SCA-CSSA, new labor cooperation structure is not used in SCA-SSA. In this experiment, the dimension's size of these functions is 10. Dim = 10 is the typical dimensions for the benchmark functions. The number of function evaluations (FEs) is 1000. We selected two different function evaluations (FEs), such as FEs = 1000 and FEs = 10,000.

TABLE 1: 18 benchmark functions.

Name	Test function	Range
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
Schwefel P2.22	$f_2(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|$	$[-10, 10]^D$
Schwefel P1.2	$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j}^{i} x_j\right)^2$	$[-100, 100]^D$
Schwefel 2.21	$f_4(x) = \max_i \{|x_i|, 1 \le i \le 30\}$	$[-100, 100]^D$
Generalized Rosenbrock	$f_5(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-10, 10]^D$
Step	$f_6(x) = \sum_{i=1}^{D} (|x_i + 0.5|)^2$	$[-100, 100]^D$
Noise	$f_7(x) = \sum_{i=1}^{D} i x_i^4 + \text{rand}[0, 1)$	$[-1.28, 1.28]^D$
Generalized Rastrigin	$f_8(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$
Ackley	$f_9 = 20 + e - 20\exp\left(-0.2\sqrt{(1/D)\sum_{i=1}^{D} x_i^2}\right) - \exp\left((1/D)\sum_{i=1}^{D}\cos 2\pi x_i\right)$	$[-32, 32]^D$
Generalized Griewank	$f_{10}(x) = \sum_{i=1}^{D}(x_i^2/4000) - \prod_{i=1}^{D}\cos(x_i/\sqrt{i}) + 1$	$[-600, 600]^D$
Generalized Penalized 1	$f_{11}(x) = (\pi/D)\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, & x_i < -a, \end{cases}$ $y_i = 1 + (1/4)x_i$	$[-50, 50]^D$
Generalized Penalized 2	$f_{12}(x) = (1/10)\left\{10\sin^2(3\pi x_1) + \sum_{i=1}^{D-1}(x_i - 1)^2[1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2[1 + \sin^2(2\pi x_D)]\right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \le x_i \le a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$	$[-50, 50]^D$

TABLE 2: Summary of the CEC'17 test functions.

Type	No.	Functions	$Fi^* = Fi(x^*)$
Unimodal functions	F_1	Shifted and Rotated Bent Cigar function	100
	F_2	Shifted and Rotated sum of different Power function*	200
	F_3	Shifted and Rotated Zakharov function	300
Simple multimodal functions	F_4	Shifted and Rotated Rosenbrock's function	400
	F_5	Shifted and Rotated Rastrigin's function	500
	F_6	Shifted and Rotated Expanded Scaffer's F_6 function	600
	F_7	Shifted and Rotated Lunacek Bi_Rastrigin function	700
	F_8	Shifted and Rotated noncontinuous Rastrigin's function	800
	F_9	Shifted and Rotated Levy function	900
	F_{10}	Shifted and Rotated Schwefel's function	1000
Hybrid functions	F_{11}	Hybrid function 1 ($N = 3$)	1100
	F_{12}	Hybrid function 2 ($N = 3$)	1200
	F_{13}	Hybrid function 3 ($N = 3$)	1300
	F_{14}	Hybrid function 4 ($N = 4$)	1400
	F_{15}	Hybrid function 5 ($N = 4$)	1500
	F_{16}	Hybrid function 6 ($N = 4$)	1600
	F_{17}	Hybrid function 6 ($N = 5$)	1700
	F_{18}	Hybrid function 6 ($N = 5$)	1800
	F_{19}	Hybrid function 6 ($N = 5$)	1900
	F_{20}	Hybrid function 6 ($N = 6$)	2000
Composition functions	F_{21}	Composition function 1 ($N = 3$)	2100
	F_{22}	Composition function 2 ($N = 3$)	2200
	F_{23}	Composition function 3 ($N = 4$)	2300
	F_{24}	Composition function 4 ($N = 4$)	2400
	F_{25}	Composition function 5 ($N = 5$)	2500
	F_{26}	Composition function 6 ($N = 5$)	2600
	F_{27}	Composition function 7 ($N = 6$)	2700
	F_{28}	Composition function 8 ($N = 6$)	2800
	F_{29}	Composition function 9 ($N = 3$)	2900
	F_{30}	Composition function 10 ($N = 3$)	3000
Search range		$[-100, 100]^D$	

TABLE 3: Initial values for the controlling parameters of algorithms.

Algorithm	Parameter	Value
PSO	Cognitive and social constants: c_1, c_2	2, 2
	Inertial weight: w	Linearly decreases from 0.9 to 0.2
	Flight interval: FQ	10
BSA	Cognitive accelerated and social accelerated coefficients: C, S	1.5,1.5
	Direct effect: a_1, a_2	Single point (probability = 1)
CSA	Awareness probability: AP	0.1
WOA	Probability switch (p)	0.4
GOA	Decreasing coefficient: c	Linearly decreases from 1 to 0.00004
SCA	Movement direction between the solution and destination: r_1	Decreases from 2 to 0
	Movement direction towards or outwards the destination: r_2	$[0, 2\pi]$
	A random weight for the destination: r_3	$[0, 2]$
	Switch between the sine and cosine components: r_4	$[0, 1]$
SSA	Producers account for percent of the total population size: $P\_percent$	0.2
SSA-SCA	Movement direction between the solution and destination: r_1	Decreases from 2 to 0
	Movement direction towards or outwards the destination: r_2	$[0, 2\pi]$
	A random weight for the destination: r_3	$[0, 2]$
	Switch between the sine and cosine components: r_4	$[0, 1]$
SCA-CSSA	Movement direction between the solution and destination: r_1	Decreases from 2 to 0
	Movement direction towards or outwards the destination: r_2	$[0, 2\pi]$
	A random weight for the destination: r_3	$[0, 2]$
	Switch between the sine and cosine components: r_4	$[0, 1]$

The fitness value curves of a run of several algorithms on about eight different functions when FEs = 10,000 are shown in Figures 1 and 2, where the horizontal axis represents the number of iterations and the vertical axis represents the fitness value. We can obviously see that the convergence speeds of several different algorithms. The maximum value (Max), the minimum value (Min), the mean value (Mean), and the variance (Var) obtained by several benchmark algorithms are shown in Tables 4 and 5, where the best results are marked in bold. Table 4 shows the performance of the several algorithms on unimodal functions when FEs = 1000, and Table 5 shows the performance of the several algorithms on multimodal functions when FEs = 1000.

(1) Unimodal Functions. The evolution curves of these algorithms on 3 unimodal functions f_1, f_3, and f_5 are given in Figure 1. It can be detected from the figure that the curve of SCA-CSSA descends fastest in the number of iterations that are far less than 10,000 times. For f_1 and f_3 case, SCA-CSSA has the fastest convergence speed compared with other algorithms. But, on functions f_1 and f_3, the original CSA and GOA got the worst solution because it is trapped in the local optimum prematurely. For function f_5, these algorithms did not find the value 0. However, SCA-CSSA continues to decline and the convergence speed of it is significantly faster than other algorithms in the early stage; the solution eventually found is the best. Overall, owing to enhance the diversity of population, SCA-CSSA has a relatively excellent convergence speed when FEs = 10,000.

From the numerical testing results on 7 unimodal functions in Table 4, we can see that SCA-CSSA can find the minimum value on f_1, f_2, f_3, f_4, f_5, and f_7. And, SCA-CSSA can find the optimal solution for all unimodal functions and get the minimum value of 0 on f_1, f_2, f_3, and f_4. It illustrates that the SCA-CSSA has best performance on unimodal functions compared to the other algorithms when FEs = 1000. Moreover, SCA-CSSA has the best maximum value (Max), the minimum value (Min), the mean value (Mean), and the standard deviation (Std) on f_1, f_3, f_5, and f_7. Obviously, the SCA-CSSA has a relatively good convergence speed. In summary, compared with these popular algorithms and hybrid algorithms, SCA-CSSA is a competitive algorithm for solving several functions and has the best performance on the most test benchmark functions.

(2) Multimodal Functions. The evolution curves of these algorithms on 3 multimodal functions f_8, f_9, and f_{10} when FEs = 10,000 are depicted in Figure 2. We can see that SCA-CSSA can find the optimal solution in the same iteration. For f_8 and f_{10} cases, SCA-CSSA continues to decline and got the best value 0. But, the original PSO and GOA get parallel straight lines because of their poor global convergence ability on these 3 functions. For function f_9, although SCA-CSSA is also trapped the local optimum, it finds the minimum value compared to other algorithms. Obviously, the convergence speed of the SCA-CSSA is significantly faster than other algorithms in the early stage, and the solution eventually found is the best. In general, owing to enhance the diversity

of population, SCA-CSSA has a relatively balanced global search capability when FEs = 10,000.

From the numerical testing results on 5 multimodal functions in Table 5, we can see that SCA-CSSA can find the optimal solution for all multimodal functions and get the minimum value of 0 on f_8 and f_{10}. The SCA-CSSA has relatively well performance on multimodal functions compared to the other algorithms. Moreover, SCA-CSSA has the best maximum value (Max), the minimum value (Min), the mean value (Mean), and the standard deviation (Std) on f_9, f_{10}, f_{11}, and f_{12}. Obviously, the SCA-CSSA has a relatively well global search capability. The main reason is that SCA-CSSA has a stronger global exploration capability based on the SCA method. In summary, the SCA-CSSA has a superior global search capability on most multimodal functions when FEs = 1000.

4.1.4. Comparison on CEC2017 Test Functions with Hybrid Algorithms and Popular Algorithms. In order to further verify the universality of the proposed SCA_CSSA algorithm, it has been compared with PSO, BSA, WOA, GOA, SSA, SCA, and SCA_SSA on the latest CEC2017 test functions. In this experiment, the dimension's size (Dim) is set to 10. The number of function evaluations (FEs) is 10,000. Experimental comparisons included the maximum value (Max), the minimum value (Min), the mean value (Mean), and the standard deviation (Std) and are given in Tables 6 and 7, where the best results are marked in bold.

SCA-CSSA gets the minimum value on F_3, F_4, F_6, F_{10}, F_{11}, F_{12}, F_{13}, and F_{14} in Table 6 and on F_{16}, F_{17}, F_{18}, F_{19}, F_{20}, F_{22}, F_{23}, F_{24}, F_{25}, F_{27}, F_{28}, F_{29}, and F_{30} in Table 7. According to the results, we can observe that SCA-CSSA does well on 21 CEC2017 test functions. Further, SCA-CSSA has the best maximum value (Max), the minimum value (Min), the mean value (Mean), and the standard deviation (Std) on F_3, F_4, F_6, F_{10}, F_{11}, F_{12}, F_{13}, F_{14}, and F_{15} in Table 6 and on F_{17}, F_{18}, F_{20}, F_{22}, F_{23}, F_{24}, and F_{25} in Table 7. Therefore, SCA-CSSA can not only find the optimal solution but also has stability on 16 CEC2017 test functions. In summary, it can be observed that SCA-CSSA obtains optimal value. It can be concluded that SCA-CSSA has better global search ability and better robustness on these test suites.

4.2. Application to Practical Pulmonary Nodule Detection Classification Problem Based on Lung CT Images. In this section, in order to evaluate the performance of SCA-CSSA in optimizing real-world optimization problem, the proposed AdaBoost-ISSA-S4VM model is used for Pulmonary Nodule Detection Classification. The CT images from LIDC/ IDRI database were used for the AdaBoost-ISSA-S4VM classification. In order to obtain fair results, all the implementations, such as SVM [24], S4VM, AdaBoost-SVM, and AdaBoost-S4VM, are conducted under the same conditions. The experimental environment for all experiments in this section is the same as in Section 4.1. And, each algorithm runs 30 times independently for each classification model. Population size and maximum generation are set to 30 and 100, respectively.

(a)

(b)

(c)
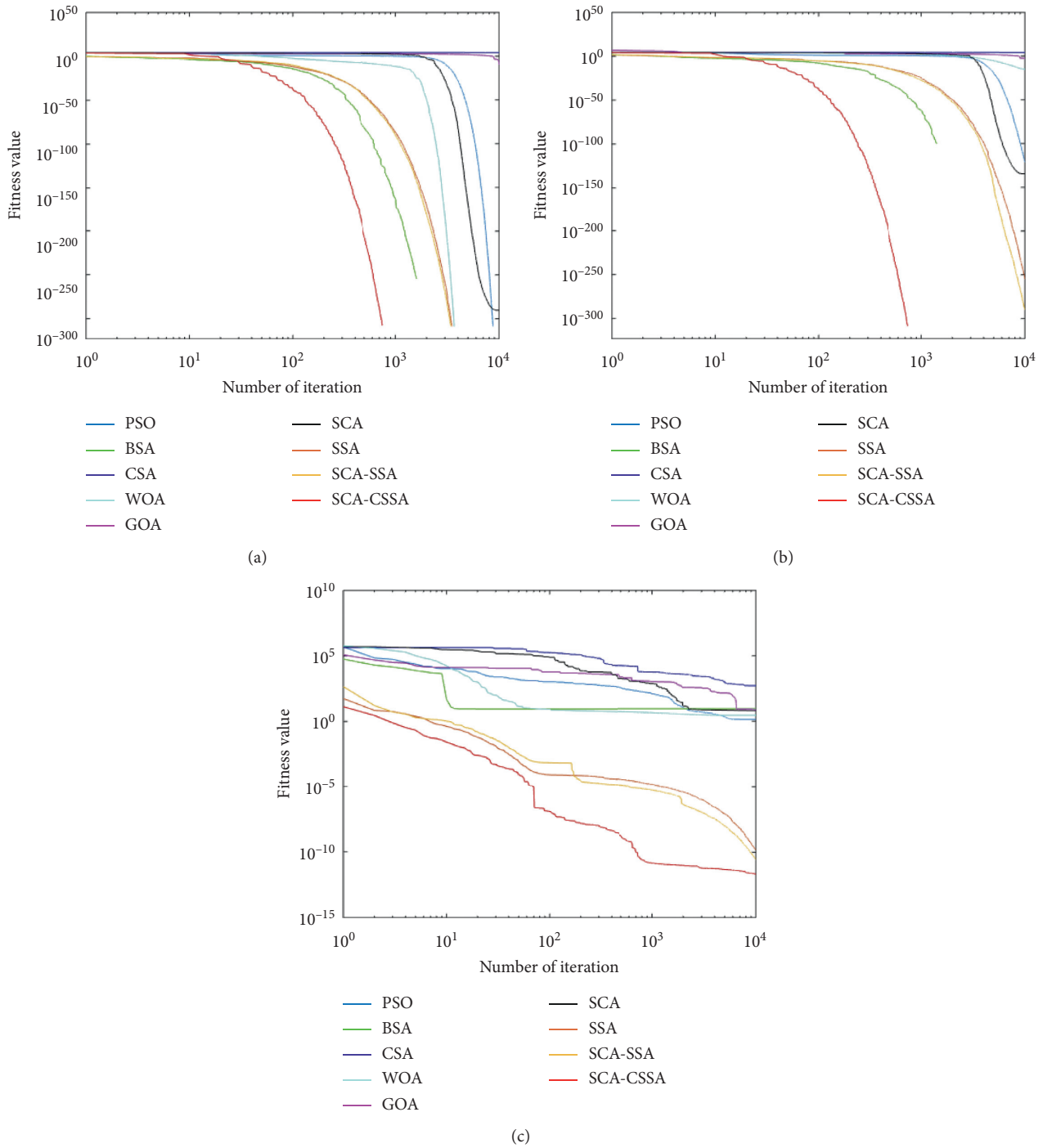
FIGURE 1: Fitness value curves of 9 algorithms on (a) f_1; (b) f_3; (c) f_5 (FEs = 10,000).

4.2.1. Design of Pulmonary Nodule Detection Classification System.

In order to identify and classify the lung nodules and non-nodules, the processing module includes the pre-processing of DICOM image, the extraction of the lung parenchyma and lung nodule, the interception of the ROI (region of interest) image, the acquisition of ROI feature vectors, and the dimensionality reduction and classification of the vectors. Block diagram of the pulmonary nodule detection classification system based on the improved AdaBoost-ISSA-S4VM is shown in Figure 3.

(a)

(b)

(c)
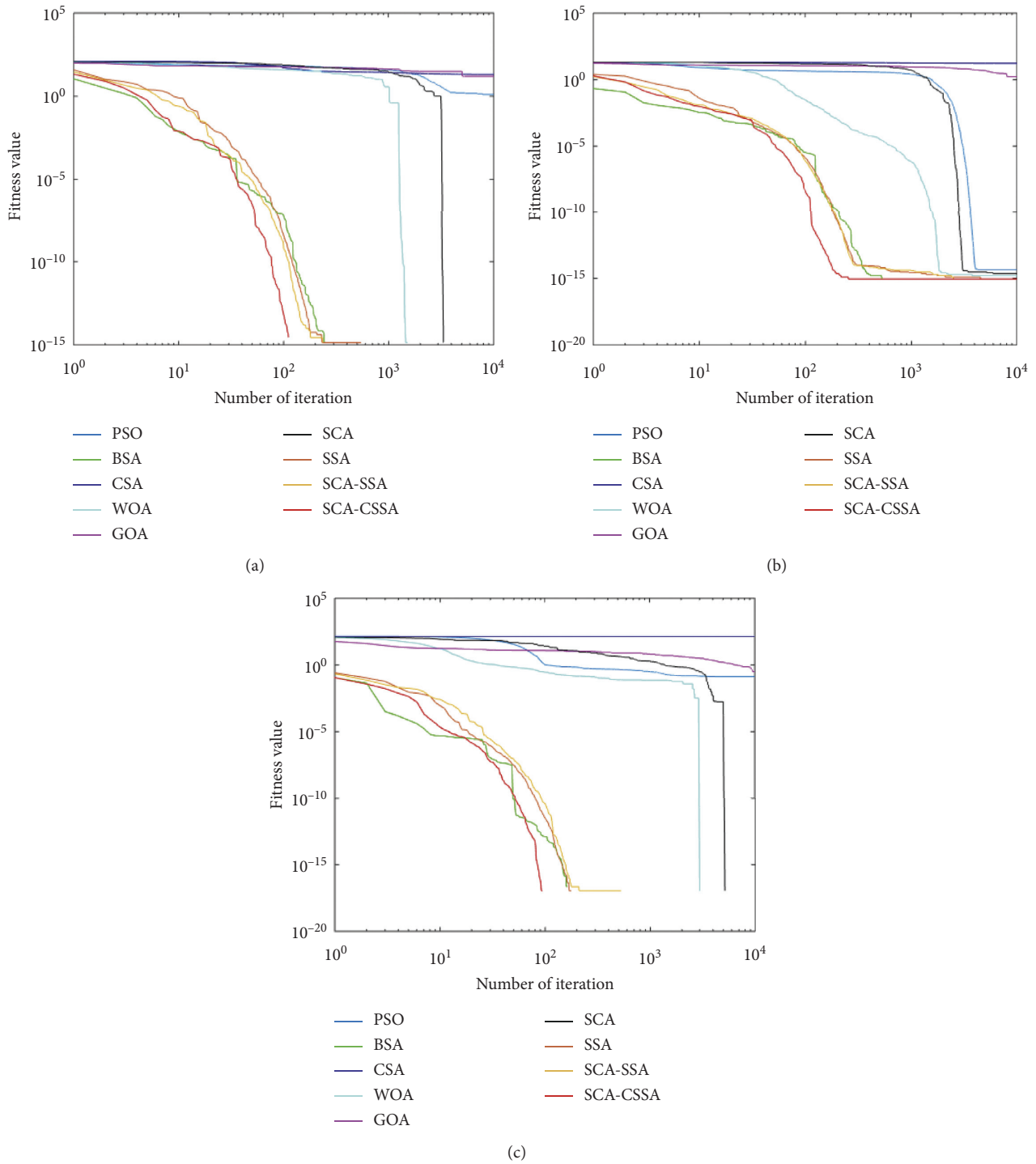
FIGURE 2: Fitness value curves of 9 algorithms on (a) f_8; (b) f_9; (c) f_{10} (FEs = 10,000).

The pulmonary nodule detection classification can be simplified by the four steps (Figure 4):

Step 1 (*Image Selection*). The selection of the CT images of lung is solitary lung nodule. At the same time, the datasets should be closely related to lung cancer sample analysis. The dataset should be relatively independent.

The dataset is randomly divided into two parts of training and testing samples.

Step 2 (*Picture Preprocessing*). Read the CT images of lung first, as shown in Figure 4(a), and then use RPCA method which is improved by weighted nonconvex regularization for image denoising [25]. After

Table 4: Comparison on 7 unimodal functions (FEs = 1000).

Fun.	Term	PSO	BSA	CSA	WOA	GOA	SCA	SSA	SCA-SSA	SCA-CSSA
f_1	Max	$1.56E+04$	$7.19E+03$	$1.57E+04$	$1.55E+04$	$7.02E+03$	$1.38E+04$	2.46	$6.16E-01$	**$3.43E-01$**
	Min	$2.44E-43$	**0**	$1.56E+04$	$2.00E-156$	$4.03E-07$	$1.30E-30$	$4.04E-127$	$4.51E-113$	**0**
	Mean	$9.95E+01$	$2.83E+01$	$1.56E+04$	$7.63E+01$	$2.92E+02$	$6.15E+02$	$1.43E-03$	$1.14E-03$	**$3.83E-04$**
	Std	$1.00E+03$	$3.41E+02$	$6.09E+01$	$8.22E+02$	$4.76E+02$	$2.12E+03$	$5.49E-02$	$2.47E-02$	**$1.10E-02$**
f_2	Max	$1.43E+03$	$2.24E+01$	$1.37E+03$	$2.77E+03$	$9.84E+01$	$1.91E+02$	**2.04**	3.12	3.81
	Min	$7.63E-23$	$2.66E-217$	8.35	$2.08E-108$	$2.52E-04$	$7.42E-22$	$9.86E-71$	$9.92E-66$	**0**
	Mean	3.00	$1.24E-01$	$1.07E+02$	3.12	7.03	1.20	**$2.63E-03$**	$6.45E-03$	$5.03E-03$
	Std	$5.28E+01$	1.31	$2.73E+02$	$8.84E+01$	$1.21E+01$	8.60	**$6.05E-02$**	$1.14E-01$	$1.26E-01$
f_3	Max	$2.02E+04$	$7.30E+03$	$2.03E+04$	$1.58E+04$	$7.94E+03$	$1.81E+04$	$4.48E+01$	$3.65E+01$	**$1.93E+01$**
	Min	$1.88E-12$	0	$2.02E+04$	$2.06E+01$	$6.16E-01$	$7.40E-10$	$4.46E-59$	$4.52E-50$	**0**
	Mean	$1.50E+02$	$2.86E+01$	$2.03E+04$	$1.75E+03$	$4.67E+02$	$1.76E+03$	$1.45E-02$	$4.81E-02$	**$4.56E-02$**
	Std	$1.39E+03$	$3.49E+02$	9.35	$2.78E+03$	$6.95E+02$	$4.12E+03$	1.42	1.21	**$4.55E-01$**
f_4	Max	$6.96E+01$	$4.79E+01$	$6.78E+01$	$6.29E+01$	$4.30E+01$	$7.08E+01$	**$3.05E-01$**	$3.14E-01$	$3.29E-01$
	Min	$6.79E-10$	$1.61E-219$	$6.62E+01$	1.56	$1.74E-01$	$4.48E-10$	$4.04E-47$	$1.85E-55$	**0**
	Mean	1.01	$2.72E-01$	$6.76E+01$	6.92	6.86	8.08	**$4.06E-04$**	$7.89E-04$	$6.08E-04$
	Std	5.55	2.90	$4.82E-01$	$1.07E+01$	6.22	$1.78E+01$	**$8.80E-03$**	$1.25E-02$	$1.19E-02$
f_5	Max	$4.81E+05$	$1.29E+05$	$4.58E+05$	$4.70E+05$	$7.32E+04$	$5.88E+05$	$2.58E+01$	$2.29E+01$	**7.73**
	Min	4.49	8.97	$7.11E+03$	5.92	6.02	7.04	$4.00E-07$	$7.63E-07$	**$7.80E-08$**
	Mean	$1.05E+03$	$2.57E+02$	$6.40E+04$	$1.36E+03$	$9.72E+02$	$1.94E+04$	$2.56E-02$	$6.11E-02$	**$1.45E-02$**
	Std	$1.57E+04$	$4.53E+03$	$1.15E+05$	$2.12E+04$	$3.08E+03$	$8.55E+04$	$6.04E-01$	$9.25E-01$	**$2.84E-01$**
f_6	Max	$1.52E+04$	$6.85E+03$	$1.67E+04$	$1.54E+04$	$6.89E+03$	$1.51E+04$	**1.99**	2.30	3.44
	Min	$3.08E-33$	1.82	$1.62E+04$	$9.85E-05$	$5.73E-07$	$2.34E-01$	$1.54E-33$	**$6.16E-34$**	$8.63E-33$
	Mean	$9.82E+01$	$2.89E+01$	$1.64E+04$	$7.14E+01$	$3.02E++02$	$5.30E+02$	**$1.35E-03$**	$3.84E-03$	$3.73E-03$
	Std	$1.01E+03$	$3.22E+02$	$1.44E+02$	$8.20E+02$	$4.90E+02$	$1.79E+03$	**$4.49E-02$**	$8.17E-02$	$1.11E-01$
f_7	Max	4.76	1.94	7.13	7.64	1.05	5.91	1.24	$8.73E-01$	**1.28**
	Min	$3.80E-03$	$1.11E-04$	$5.08E-01$	$9.47E-04$	$3.00E-03$	$1.64E-03$	$3.91E-04$	$4.64E-04$	**$9.87E-05$**
	Mean	$3.59E-01$	$6.41E-03$	$6.47E-01$	$4.59E-02$	$1.16E-02$	$2.37E-01$	$2.06E-03$	$2.41E-03$	**$1.93E-03$**
	Std	$6.53E-01$	$8.07E-02$	$5.89E-01$	$4.07E-01$	$4.77E-02$	$7.68E-01$	$4.16E-02$	$2.93E-02$	**$1.96E-02$**

Table 5: Comparison on 6 multimodal functions (FEs = 1000).

Fun.	Term	PSO	BSA	CSA	WOA	GOA	SCA	SSA	SCA-SSA	SCA-CSSA
f_8	Max	$1.23E+02$	$1.05E+02$	$1.18E+02$	$1.12E+02$	$1.11E+02$	$1.12E+02$	**$3.34E+01$**	$5.18E++01$	$3.85E+01$
	Min	3.36	**0**	$3.15E+01$	**0**	$5.87E+01$	$6.42E-11$	**0**	**0**	**0**
	Mean	$2.36E+01$	$7.33E-01$	$4.20E+01$	5.86	$7.39E+01$	$1.72E+01$	$4.73E-02$	$7.62E-02$	**$4.12E-02$**
	Std	$2.36E+01$	$5.56E+00$	$1.76E+01$	$1.41E+01$	$1.57E+01$	$2.82E+01$	1.21	1.74	**1.10**
f_9	Max	$1.99E+01$	$1.79E+01$	$1.91E+01$	$1.96E+01$	$1.77E+01$	$2.00E+01$	2.74	2.59	**1.07**
	Min	$4.88E-15$	**$8.88E-16$**	$1.79E+01$	$1.60E-15$	$5.78E-01$	$8.73E-14$	**$8.88E-16$**	**$8.88E-16$**	**$8.88E-16$**
	Mean	1.21	$1.27E-01$	$1.84E+01$	$2.95E-01$	5.60	4.07	$3.01E-03$	$6.38E-03$	**$1.17E-03$**
	Std	2.16	1.33	$4.25E-01$	1.81	4.09	6.82	$8.74E-02$	$1.06E-01$	**$2.94E-02$**
f_{10}	Max	$1.56E+02$	$6.94E+01$	$1.57E+02$	$1.45E+02$	$5.48E+01$	$1.30E+02$	$3.20E-01$	$4.10E-01$	**$2.10E-01$**
	Min	$1.21E-01$	**0**	$1.57E+02$	$4.85E-02$	$2.95E-01$	$3.04E-02$	**0**	**0**	**0**
	Mean	5.78	$2.99E-01$	$1.57E+02$	$7.52E-01$	3.42	7.12	$3.52E-04$	$7.56E-04$	**$1.91E-04$**
	Std	$2.32E+01$	3.37	$2.46E-12$	7.26	3.74	$2.07E+01$	$1.03E-02$	$1.58E-02$	**$5.36E-03$**
f_{11}	Max	$9.94E+07$	$1.39E+07$	$5.90E+07$	$8.76E+07$	$1.02E+07$	$6.01E+07$	1.27	1.44	**$7.33E-01$**
	Min	$4.80E-32$	$4.69E-01$	$4.78E+07$	$6.21E-03$	$4.35E-05$	$8.11E-02$	$4.75E-32$	$4.76E-32$	**$4.73E-32$**
	Mean	$1.71E+05$	$1.96E+04$	$4.78E+07$	$1.90E+05$	$1.54E+04$	$5.44E+06$	$1.32E-03$	$2.03E-03$	**$3.13E-04$**
	Std	$3.54E+06$	$4.59E+05$	$4.49E+06$	$3.29E+06$	$3.64E+05$	$1.42E+07$	$4.03E-02$	$4.90E-02$	**$1.47E-02$**
f_{12}	Max	$1.64E+08$	$1.37E+07$	$2.63E+08$	$1.23E+08$	$3.17E+07$	$1.30E+08$	$2.95E-01$	$1.49E-01$	**$3.58E-01$**
	Min	$3.72E-32$	$9.57E-01$	$2.63E+08$	$2.71E-04$	$1.06E-02$	$3.06E-01$	$1.57E-32$	**$1.35E-32$**	**$1.35E-32$**
	Mean	$3.58E+05$	$2.19E+04$	$2.63E+08$	$3.88E+05$	$5.97E+04$	$5.15E+06$	$4.25E-04$	$5.95E-04$	**$3.39E-04$**
	Std	$6.47E+06$	$4.64E+05$	$2.55E-06$	$5.51E+06$	$1.12E+06$	$2.30E+07$	$1.02E-02$	$7.52E-03$	**$8.64E-03$**

enhancing the contrast ratio of the images through binarization processing, this study uses optimal threshold segmentation (OTSU) method to sharpen the image, as shown in Figure 4(b).

Step 3 (Lung Parenchyma Extraction). In order to narrow the range of the lung parenchyma and reduce the difficulty of detection, thus improving the accuracy of detection, we fill the lung parenchyma, as shown in

TABLE 6: Comparison of numerical testing results on CEC2017 test sets (F_1–F_{15}, FEs = 10,000).

Fun	Term	PSO	BSA	WOA	GOA	SCA	SSA	SCA-SSA	SCA-CSSA
F_1	Max	$1.92E+10$	$1.44E+10$	$2.70E+10$	$1.26E+10$	$1.95E+10$	$2.18E+10$	$2.00E+10$	**$1.38E+10$**
	Min	**$4.65E+02$**	$1.74E+09$	$3.78E+04$	$3.69E+03$	$4.27E+08$	$4.56E+03$	$7.30E+08$	$2.09E+03$
	Mean	$1.87E+07$	$2.31E+09$	$3.05E+07$	$1.76E+08$	$1.22E+09$	$8.64E+06$	$2.24E+09$	**$3.61E+06$**
	Std	$4.71E+08$	$9.39E+08$	$5.99E+08$	$3.16E+08$	$1.10E+09$	$3.14E+08$	$2.46E+09$	**$1.67E+08$**
F_3	Max	$1.35E+06$	$7.41E+04$	$1.96E+07$	$6.27E+04$	$1.95E+05$	$8.72E+04$	$6.65E+04$	**$4.30E+04$**
	Min	**$3.00E+02$**	$8.88E+03$	$3.60E+02$	**$3.00E+02$**	$7.16E+02$	**$3.00E+02$**	$5.85E+03$	**$3.00E+02$**
	Mean	$6.09E+02$	$1.09E+04$	$2.79E+03$	$8.61E+02$	$3.98E+03$	$3.89E+02$	$1.03E+04$	**$3.50E+02$**
	Std	$1.68E+04$	$2.30E+03$	$1.98E+05$	$1.91E+03$	$7.16E+03$	$1.44E+03$	$4.55E+03$	**$7.74E+02$**
F_4	Max	$3.58E+03$	$2.32E+03$	$3.47E+03$	$1.69E+03$	$3.76E+03$	$1.87E+03$	$2.51E+03$	**$1.19E+03$**
	Min	$4.01E+02$	$6.05E+02$	$4.21E+02$	$4.05E+02$	$4.24E+02$	**$4.00E+02$**	$4.82E+02$	**$4.00E+02$**
	Mean	$4.04E+02$	$6.37E+02$	$4.26E+02$	$4.16E+02$	$4.63E+02$	$4.02E+02$	$6.70E+02$	**$4.02E+02$**
	Std	$6.72E+01$	$6.56E+01$	$5.12E+01$	$2.33E+01$	$9.46E+01$	$2.42E+01$	$2.13E+02$	**$1.20E+01$**
F_5	Max	$6.56E+02$	$6.37E+02$	$6.73E+02$	**$6.23E+02$**	$6.62E+02$	$6.51E+02$	$6.42E+02$	**$6.23E+02$**
	Min	$5.39E+02$	$5.63E+02$	$5.49E+02$	**$5.20E+02$**	$5.38E+02$	$5.30E+02$	$5.42E+02$	$5.28E+02$
	Mean	$5.40E+02$	$5.66E+02$	$5.51E+02$	$5.31E+02$	$5.53E+02$	$5.31E+02$	$5.51E+02$	**$5.28E+02$**
	Std	3.66	3.33	5.30	$1.07E+01$	$1.43E+01$	3.43	$1.26E+01$	**3.16**
F_6	Max	$6.98E+02$	$6.71E+02$	$7.16E+02$	$6.72E+02$	$7.02E+02$	$6.95E+02$	$6.87E+02$	**$6.61E+02$**
	Min	$6.07E+02$	$6.42E+02$	$6.29E+02$	$6.07E+02$	$6.14E+02$	$6.08E+02$	$6.21E+02$	**$6.02E+02$**
	Mean	$6.08E+02$	$6.42E+02$	$6.31E+02$	$6.18E+02$	$6.24E+02$	$6.09E+02$	$6.30E+02$	**$6.02E+02$**
	Std	3.57	1.26	4.21	$1.13E+01$	$1.17E+01$	2.47	$1.08E+01$	**1.72**
F_7	Max	$1.21E+03$	$1.00E+03$	$1.19E+03$	$1.02E+03$	$1.21E+03$	$9.05E+02$	**$8.97E+02$**	$9.02E+02$
	Min	**$7.20E+02$**	$7.90E+02$	$7.74E+02$	$7.30E+02$	$7.64E+02$	$7.67E+02$	$7.86E+02$	$7.42E+02$
	Mean	$7.26E+02$	$7.93E+02$	$7.75E+02$	$7.51E+02$	$7.89E+02$	$7.71E+02$	$8.07E+02$	$7.44E+02$
	Std	$1.43E+01$	5.43	8.82	$1.87E+01$	$3.72E+01$	8.58	$2.39E+01$	**5.10**
F_8	Max	$9.56E+02$	$9.10E+02$	$9.51E+02$	$9.29E+02$	$9.45E+02$	$9.29E+02$	$9.24E+02$	**$8.99E+02$**
	Min	$8.19E+02$	$8.46E+02$	$8.55E+02$	**$8.18E+02$**	$8.32E+02$	$8.31E+02$	$8.50E+02$	$8.21E+02$
	Mean	**$8.20E+02$**	$8.47E+02$	$8.56E+02$	$8.31E+02$	$8.48E+02$	$8.31E+02$	$8.59E+02$	$8.22E+02$
	Std	3.85	2.19	3.73	$1.31E+01$	$1.84E+01$	2.37	$1.13E+01$	**1.74**
F_9	Max	$5.51E+03$	$4.09E+03$	$6.41E+03$	$3.10E+03$	$5.50E+03$	$3.61E+03$	$3.37E+03$	**$2.78E+03$**
	Min	**$9.00E+02$**	$1.51E+03$	$1.36E+03$	**$9.00E+02$**	$9.56E+02$	$1.39E+03$	$1.28E+03$	$9.33E+02$
	Mean	**$9.07E+02$**	$1.54E+03$	$1.44E+03$	$9.44E+02$	$1.18E+03$	$1.41E+03$	$1.50E+03$	$9.43E+02$
	Std	$1.01E+02$	$5.76E+01$	$1.80E+02$	$7.16E+01$	$4.02E+02$	$6.87E+01$	$2.96E+02$	**$4.25E+01$**
F_{10}	Max	$3.92E+03$	**$3.52E+03$**	$3.67E+03$	$3.64E+03$	$3.68E+03$	$3.61E+03$	$3.66E+03$	**$3.52E+03$**
	Min	$1.94E+03$	$2.29E+03$	$1.65E+03$	$1.84E+03$	$2.02E+03$	$1.92E+03$	$2.14E+03$	**$1.61E+03$**
	Mean	$1.97E+03$	$2.31E+03$	$1.70E+03$	$2.11E+03$	$2.14E+03$	$1.98E+03$	$2.29E+03$	**$1.67E+03$**
	Std	**$1.04E+02$**	$4.89E+01$	$1.55E+02$	$1.80E+02$	$1.56E+02$	$9.65E+01$	$2.20E+02$	$1.09E+02$
F_{11}	Max	$6.40E+04$	$8.54E+03$	$4.26E+04$	$1.42E+04$	$2.10E+04$	$1.05E+04$	$1.61E+04$	**$5.46E+03$**
	Min	$1.14E+03$	$1.46E+03$	$1.16E+03$	**$1.12E+03$**	$1.15E+03$	$1.19E+03$	$1.37E+03$	**$1.12E+03$**
	Mean	$1.18E+03$	$1.54E+03$	$1.19E+03$	$1.17E+03$	$1.33E+03$	$1.19E+03$	$1.75E+03$	**$1.12E+03$**
	Std	$1.27E+03$	$2.19E+02$	$5.79E+02$	$2.21E+02$	$4.96E+02$	$1.48E+02$	$7.15E+02$	**$5.63E+01$**
F_{12}	Max	$3.57E+09$	$2.19E+09$	$3.71E+09$	$2.62E+09$	$1.99E+09$	$1.22E+09$	$1.83E+09$	**$5.84E+08$**
	Min	$1.06E+04$	$8.94E+06$	$5.78E+05$	$1.39E+05$	$4.13E+06$	**$8.20E+03$**	$1.54E+07$	$1.02E+04$
	Mean	$2.36E+06$	$1.39E+07$	$2.02E+06$	$6.15E+06$	$3.05E+07$	$3.57E+05$	$6.33E+07$	**$1.14E+05$**
	Std	$7.49E+07$	$4.66E+07$	$4.27E+07$	$3.18E+07$	$6.96E+07$	$1.56E+07$	$1.22E+08$	**$6.35E+06$**
F_{13}	Max	$3.85E+08$	$1.29E+08$	$4.56E+08$	$1.07E+08$	$3.85E+08$	$1.46E+08$	$2.27E+08$	**$4.59E+07$**
	Min	$7.89E+03$	$3.19E+03$	$2.03E+04$	$3.28E+03$	$1.58E+04$	$3.59E+03$	$2.42E+04$	**$2.03E+03$**
	Mean	$1.30E+05$	$2.06E+04$	$1.32E+05$	$3.21E+04$	$2.30E+05$	$2.71E+04$	$5.00E+05$	**$1.03E+04$**
	Std	$5.79E+06$	$1.33E+06$	$5.77E+06$	$1.12E+06$	$5.38E+06$	$1.56E+06$	$5.50E+06$	**$4.71E+05$**
F_{14}	Max	$1.98E+07$	$6.27E+05$	$1.21E+07$	$8.62E+05$	$5.10E+06$	$1.01E+06$	$7.43E+06$	**$3.07E+05$**
	Min	**$1.45E+03$**	$1.69E+03$	$1.51E+03$	$1.44E+03$	$1.52E+03$	$1.46E+03$	$1.53E+03$	**$1.45E+03$**
	Mean	$5.16E+03$	$1.88E+03$	$3.18E+03$	$1.86E+03$	$3.35E+03$	$1.65E+03$	$3.52E+03$	**$1.54E+03$**
	Std	$2.40E+05$	$7.24E+03$	$1.31E+05$	$8.97E+03$	$7.50E+04$	$1.11E+04$	$7.81E+04$	**$3.44E+03$**
F_{15}	Max	$4.34E+07$	$3.44E+06$	$7.65E+07$	**$7.29E+05$**	$1.34E+07$	$1.25E+07$	$8.62E+06$	$7.62E+05$
	Min	**$1.53E+03$**	$4.84E+03$	$2.33E+03$	$1.60E+03$	$1.76E+03$	$1.59E+03$	$1.98E+03$	**$1.53E+03$**
	Mean	$1.24E+04$	$7.19E+03$	$1.12E+04$	$3.00E+03$	$6.23E+03$	$3.35E+03$	$8.97E+03$	**$1.71E+03$**
	Std	$5.97E+05$	$3.73E+04$	$7.66E+05$	$1.62E+04$	$1.50E+05$	$1.29E+05$	$9.35E+04$	**$8.23E+03$**

Table 7: Comparison of numerical testing results on CEC2017 test sets (F_{16}–F_{30}, FEs = 10,000).

| Fun	Term	PSO	BSA	WOA	GOA	SCA	SSA	SCA-SSA	SCA-CSSA
F_{16}	Max	2.81E + 03	2.62E + 03	2.88E + 03	**2.50E + 03**	2.77E + 03	2.73E + 03	2.68E + 03	2.58E + 03
	Min	1.82E + 03	1.93E + 03	1.76E + 03	1.99E + 03	1.65E + 03	1.84E + 03	1.72E + 03	**1.73E + 03**
	Mean	1.82E + 03	2.00E + 03	1.78E + 03	2.01E + 03	**1.75E + 03**	1.84E + 03	1.83E + 03	1.78E + 03
	Std	2.81E + 01	4.95E + 01	4.40E + 01	5.57E + 01	1.00E + 02	**2.38E + 01**	1.28E + 02	5.95E + 01
F_{17}	Max	2.56E + 03	2.21E + 03	2.48E + 03	2.45E + 03	2.42E + 03	2.33E + 03	2.37E + 03	**2.20E + 03**
	Min	1.77E + 03	1.86E + 03	1.77E + 03	1.88E + 03	1.76E + 03	1.78E + 03	1.77E + 03	**1.75E + 03**
	Mean	1.78E + 03	1.86E + 03	1.77E + 03	1.90E + 03	1.79E + 03	1.78E + 03	1.80E + 03	**1.75E + 03**
	Std	1.87E + 01	**8.12**	2.14E + 01	3.85E + 01	4.73E + 01	1.32E + 01	5.46E + 01	9.96
F_{18}	Max	1.56E + 09	2.95E + 08	7.81E + 08	1.68E + 08	2.77E + 08	4.24E + 08	3.51E + 08	**5.92E + 07**
	Min	6.65E + 03	3.59E + 03	1.53E + 04	6.77E + 03	3.02E + 04	1.92E + 03	5.25E + 04	**1.85E + 03**
	Mean	9.05E + 05	1.37E + 05	1.29E + 05	8.75E + 04	4.10E + 05	7.57E + 04	1.40E + 07	**1.17E + 04**
	Std	3.20E + 07	4.34E + 06	8.45E + 06	2.00E + 06	5.17E + 06	4.73E + 06	1.86E + 07	**6.34E + 05**
F_{19}	Max	8.24E + 07	**7.44E + 06**	8.09E + 07	4.07E + 07	3.12E + 07	2.34E + 07	2.51E + 07	1.72E + 07
	Min	1.91E + 03	2.54E + 03	1.30E + 04	1.96E + 03	2.11E + 03	1.94E + 03	3.14E + 03	**1.92E + 03**
	Mean	3.09E + 04	6.41E + 03	4.19E + 04	9.89E + 03	2.42E + 04	7.92E + 03	3.52E + 04	**4.13E + 03**
	Std	1.40E + 06	**8.05E + 04**	1.06E + 06	4.52E + 05	6.37E + 05	3.20E + 05	3.77E + 05	1.77E + 05
F_{20}	Max	2.57E + 03	2.41E + 03	2.64E + 03	2.66E + 03	2.50E + 03	2.58E + 03	2.51E + 03	**2.46E + 03**
	Min	2.10E + 03	2.16E + 03	2.11E + 03	2.16E + 03	2.07E + 03	2.09E + 03	2.08E + 03	**2.03E + 03**
	Mean	2.11E + 03	2.17E + 03	2.12E + 03	2.19E + 03	2.10E + 03	2.09E + 03	2.14E + 03	**2.04E + 03**
	Std	1.44E + 01	**8.34**	2.26E + 01	3.93E + 01	5.07E + 01	1.14E + 01	5.83E + 01	2.46E + 01
F_{21}	Max	2.44E + 03	**2.41E + 03**	2.44E + 03	2.41E + 03	2.44E + 03	2.42E + 03	2.43E + 03	**2.41E + 03**
	Min	2.32E + 03	2.33E + 03	2.28E + 03	2.32E + 03	**2.21E + 03**	2.32E + 03	2.31E + 03	2.29E + 03
	Mean	2.32E + 03	2.34E + 03	2.28E + 03	2.33E + 03	**2.23E + 03**	2.32E + 03	2.33E + 03	2.30E + 03
	Std	3.88E + 00	1.15E + 01	1.55E + 01	2.46E + 01	4.73E + 01	3.09E + 00	3.02E + 01	**1.73E + 01**
F_{22}	Max	4.04E + 03	3.48E + 03	4.12E + 03	4.85E + 03	4.20E + 03	4.16E + 03	4.27E + 03	**3.14E + 03**
	Min	**2.30E + 03**	2.57E + 03	2.31E + 03	**2.30E + 03**	2.32E + 03	**2.30E + 03**	2.44E + 03	**2.30E + 03**
	Mean	2.31E + 03	2.60E + 03	2.32E + 03	2.33E + 03	2.40E + 03	**2.30E + 03**	2.55E + 03	**2.30E + 03**
	Std	4.53E + 01	2.72E + 01	4.79E + 01	5.02E + 01	1.23E + 02	2.97E + 01	1.97E + 02	**1.06E + 01**
F_{23}	Max	2.91E + 03	2.84E + 03	2.93E + 03	2.83E + 03	2.81E + 03	2.79E + 03	2.81E + 03	**2.76E + 03**
	Min	2.70E + 03	2.74E + 03	2.64E + 03	2.61E + 03	2.64E + 03	2.63E + 03	2.65E + 03	**2.61E + 03**
	Mean	2.70E + 03	2.75E + 03	2.65E + 03	2.62E + 03	2.65E + 03	2.63E + 03	2.66E + 03	**2.62E + 03**
	Std	8.24	6.67	1.03E + 01	2.92E + 01	2.83E + 01	3.35	2.07E + 01	**2.19**
F_{24}	Max	3.04E + 03	3.00E + 03	3.05E + 03	2.93E + 03	2.94E + 03	**2.91E + 03**	2.97E + 03	2.92E + 03
	Min	2.82E + 03	2.86E + 03	2.77E + 03	2.74E + 03	2.73E + 03	2.77E + 03	2.78E + 03	**2.72E + 03**
	Mean	2.82E + 03	2.87E + 03	2.77E + 03	2.76E + 03	2.76E + 03	2.77E + 03	2.79E + 03	**2.74E + 03**
	Std	8.28E + 00	7.49E + 00	7.41E + 00	2.96E + 01	4.61E + 01	4.22E + 00	1.68E + 01	**1.64E + 01**
F_{25}	Max	4.98E + 03	3.96E + 03	4.80E + 03	3.65E + 03	4.61E + 03	4.30E + 03	4.52E + 03	**3.57E + 03**
	Min	2.92E + 03	3.09E + 03	2.94E + 03	2.95E + 03	2.94E + 03	2.93E + 03	2.95E + 03	**2.89E + 03**
	Mean	2.92E + 03	3.10E + 03	2.95E + 03	2.96E + 03	2.99E + 03	2.93E + 03	3.07E + 03	**2.90E + 03**
	Std	4.65E + 01	2.22E + 01	4.20E + 01	3.31E + 01	6.75E + 01	2.01E + 01	2.23E + 02	**1.13E + 01**
F_{26}	Max	4.67E + 03	4.65E + 03	5.17E + 03	**4.06E + 03**	4.90E + 03	5.29E + 03	5.03E + 03	4.46E + 03
	Min	3.14E + 03	3.73E + 03	2.99E + 03	**2.90E + 03**	3.05E + 03	3.43E + 03	3.24E + 03	3.10E + 03
	Mean	3.16E + 03	3.76E + 03	3.04E + 03	**2.95E + 03**	3.10E + 03	3.43E + 03	3.56E + 03	3.14E + 03
	Std	7.01E + 01	3.82E + 01	1.43E + 02	**6.14E + 01**	1.14E + 02	3.62E + 01	3.55E + 02	6.78E + 01
F_{27}	Max	3.49E + 03	3.29E + 03	3.50E + 03	**3.18E + 03**	3.33E + 03	3.34E + 03	3.31E + 03	3.27E + 03
	Min	3.12E + 03	3.18E + 03	3.13E + 03	3.10E + 03	3.10E + 03	3.10E + 03	3.10E + 03	**3.09E + 03**
	Mean	3.12E + 03	3.18E + 03	3.13E + 03	3.10E + 03	3.10E + 03	3.10E + 03	3.12E + 03	**3.09E + 03**
	Std	1.19E + 01	3.40	1.13E + 01	3.11E + 01	3.14E + 01	**3.34**	2.11E + 01	6.92
F_{28}	Max	4.14E + 03	3.94E + 03	4.42E + 03	4.23E + 03	4.03E + 03	4.05E + 03	**3.44E + 03**	3.49E + 03
	Min	3.17E + 03	3.47E + 03	3.43E + 03	3.38E + 03	3.22E + 03	3.33E + 03	3.29E + 03	**3.21E + 03**
	Mean	3.18E + 03	3.54E + 03	3.43E + 03	3.39E + 03	3.25E + 03	3.33E + 03	3.29E + 03	**3.24E + 03**
	Std	3.97E + 01	5.20E + 01	2.38E + 01	3.67E + 01	6.14E + 01	1.33E + 01	**6.27**	1.77E + 01
F_{29}	Max	4.25E + 03	3.78E + 03	4.01E + 03	**3.58E + 03**	3.90E + 03	3.77E + 03	3.89E + 03	3.80E + 03
	Min	3.21E + 03	3.36E + 03	3.28E + 03	3.28E + 03	3.20E + 03	3.26E + 03	3.26E + 03	**3.20E + 03**
	Mean	3.21E + 03	3.36E + 03	3.29E + 03	3.30E + 03	3.24E + 03	3.26E + 03	3.31E + 03	**3.23E + 03**
	Std	2.61E + 01	**1.12E + 01**	2.32E + 01	4.00E + 01	5.83E + 01	1.26E + 01	7.90E + 01	2.47E + 01
F_{30}	Max	2.96E + 08	6.29E + 07	2.70E + 08	**1.54E + 07**	8.63E + 07	7.88E + 07	7.97E + 07	3.54E + 07
	Min	5.44E + 03	2.53E + 06	4.05E + 05	9.20E + 03	3.95E + 05	2.13E + 05	1.05E + 04	**5.19E + 03**
	Mean	2.58E + 05	2.95E + 06	7.44E + 05	9.12E + 04	7.63E + 05	2.52E + 05	8.56E + 05	1.82E + 05
	Std	6.56E + 06	1.58E + 06	3.18E + 06	**6.69E + 05**	1.64E + 06	1.22E + 06	3.10E + 06	7.56E + 05
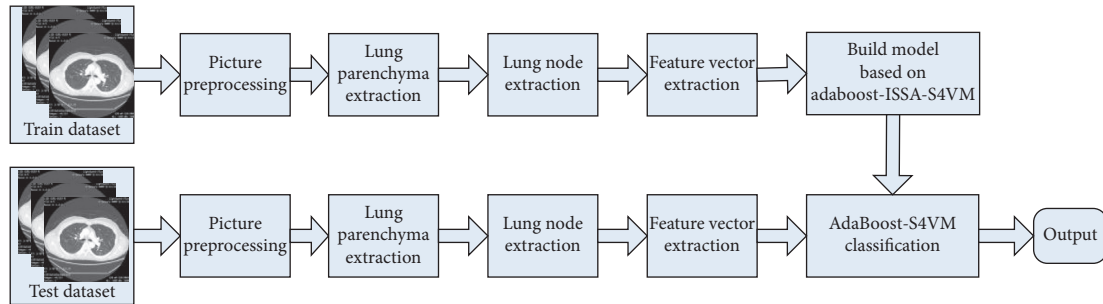
FIGURE 3: Block diagram of the oil layer classification system based on AdaBoost-ISSA-S4VM.
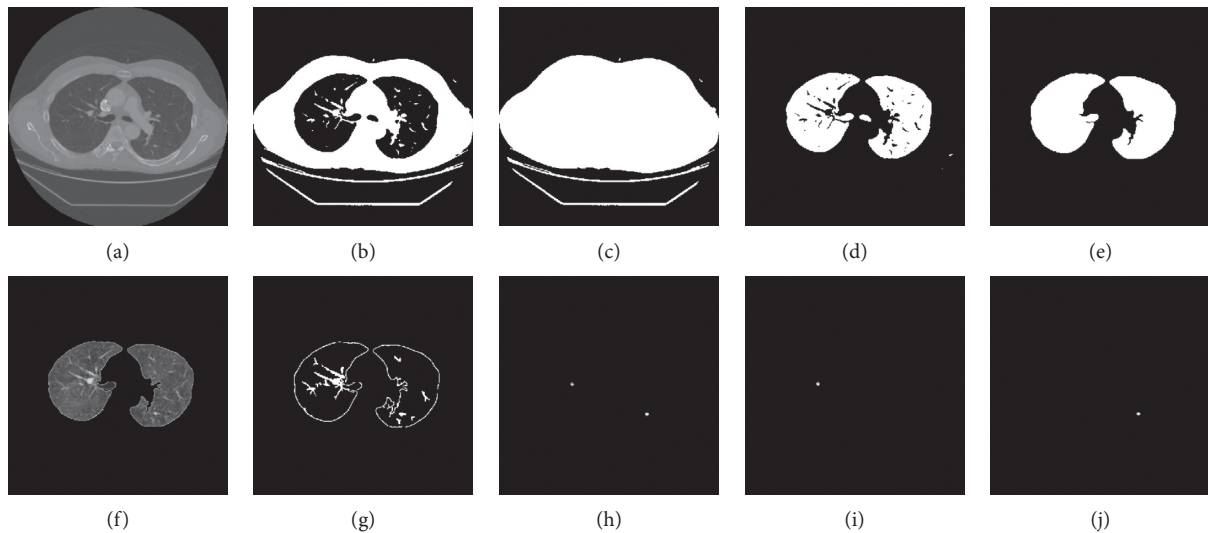

FIGURE 4: Some steps in the extraction process of lung nodules.

Figure 4(c). Then, do XOR to the figure; the area of lung parenchyma is obtained as shown in Figure 4(d). After deleting external and small area in lung parenchyma, the morphological method is used to repair the edge of the image, as shown in Figure 4(e). Finally, the lung parenchyma template and the image after pretreatment are multiplied to obtain the required lung parenchyma, as shown in Figure 4(f).

Step 4 (ROI Region Extraction). The optimal threshold segmentation method is used again in order to extract the pathological part. After eliminating the linear structures, the small area is removed by removing smaller connected components, as shown in Figure 4(g). Finally, we can remove the false positives by the dot filter method which can remove the linear structure effectively and get the ROI regions as shown in Figure 4(h). The ROI regions include lung nodules and non-nodules as shown in Figures 4(i) and 4(j).

Step 5 (Feature Vector Extraction). In order to avoid the influence of the particularity, heterogeneity, texture, and complexity of lung nodules on the selection of feature vectors, we introduce the Curvelet transform with rigorous mathematical theory based on the conventional feature extraction methods [24] to supplement the feature vectors.

Step 6 (Classifier Training and Feature Classification). In the AdaBoost-ISSA-S4VM classifier, input the actual feature vector of lung node dataset after feature vector extraction, use AdaBoost-ISSA-S4VM classification algorithm to train, and finally get the AdaBoost-ISSA-S4VM classification model. The train dataset is identified by the trained Ada-Boost-ISSA-S4VM classification model and the classification results are obtained as output.

4.2.2. Practical Application. In Section 2.3, the performance of the proposed ISSA is simulated and analyzed on benchmark functions. In order to test the application effect of the improved AdaBoost-ISSA-S4VM classification model, the CT images of lung from LIDC/IDRI database is selected for experiments. According to the description of the XML annotation file of the case nodule information in the database, the solid solitary lung nodule was analyzed. ROI region extraction on the DICOM image is performed before feature vector extraction, as shown in Figure 4. Figure 4 shows some partial steps of lung nodule extraction, where the (a) is the original CT image, the (f) is a lung parenchyma after being processed, and the (i) and (j) are the suspected lesion areas. Finally, 200 RIO regions were
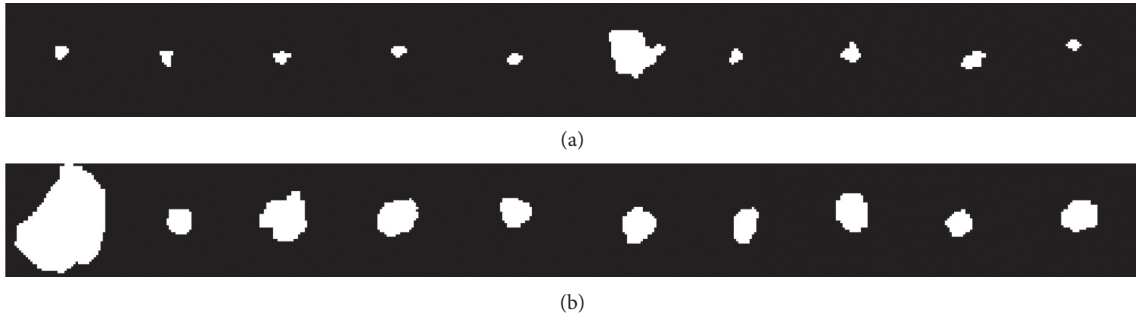
(a)

(b)

FIGURE 5: Some of lung nodules and non-nodules obtained in the experiment. (a) Non-nodules; (b) lung nodules.

TABLE 8: Feature vector extraction results of ROI regions.

Sample	1	2	3	4	5	...	713	714	715
1	$4.5925E+02$	1.0071	$7.9167E-01$	$4.5332E-01$	5.4959	...	$1.3085E-05$	$1.3085E-05$	$1.3085E-05$
2	$7.4675E+02$	$5.3339E-01$	$8.7500E-01$	$1.6086E-02$	4.2682	...	$2.6276E-05$	$2.6276E-05$	$2.6276E-05$
3	$5.2038E+02$	$5.9813E-01$	$9.6667E-01$	$2.9988E-01$	3.6474	...	$1.1133E-06$	$1.1133E-06$	$1.1133E-06$
4	$3.8388E+02$	$8.3633E-01$	$6.2963E-01$	$5.6769E-01$	1.7350	...	$1.5003E-06$	$1.5003E-06$	$1.5003E-06$
5	$6.8213E+02$	$6.7671E-01$	$7.7778E-01$	$2.8724E-01$	2.1450	...	$1.1633E-05$	$1.1633E-05$	$1.1633E-05$
6	$6.0013E+02$	$4.5361E-01$	$5.5102E-01$	$1.0952E-02$	1.9821	...	$1.6688E-05$	$1.6688E-05$	$1.6688E-05$
7	$6.8838E+02$	$8.6916E-01$	$6.1111E-01$	$1.3839E-01$	2.6320	...	$2.2705E-05$	$2.2705E-05$	$2.2705E-05$
...	...	...	...	...	...	...	...	...	...
123	$2.2238E+03$	$7.5330E-01$	$5.6944E-01$	$3.5254E-01$	1.2325	...	$1.1612E-04$	$1.1612E-04$	$1.1612E-04$
124	$8.4838E+02$	$5.8792E-01$	$9.4872E-01$	$1.2225E-01$	5.1242	...	$2.1470E-05$	$2.1470E-05$	$2.1470E-05$
125	$1.8721E+03$	$6.4757E-01$	$8.3051E-01$	$5.4866E-02$	3.6554	...	$1.7342E-04$	$1.7342E-04$	$1.7342E-04$

extracted in this experiment, including 80 lung nodules and 195 non-nodules. After group them randomly, 125 of them are as the training set and 200 are as the test set for supervised learning model. And, 125 of them are as the labeled set and 200 are as the unlabeled set for semi-supervised learning model. Figure 5 shows some lung nodules and non-nodules gained from the experiment.

Feature vector extraction on the ROI regions datasets is performed before the training of AdaBoost-ISSA-S4VM classification model on the training dataset, as shown in Table 8. 715 feature vector parameters are extracted, including 12 morphological feature parameters, 10 gray-scale feature parameters, 7 texture feature parameters, and 686 Curvelet transform coefficients. Then, these feature vectors are normalized to prevent features with large dynamic range from affecting the characteristic of features with small one, as shown in Table 9.

The CT image after preprocessing, extracting lung parenchyma and lung nodes, and extracting the feature of the lung nodes is used to train the pulmonary nodule detection classification model. In order to measure the performance of the AdaBoost-ISSA-S4VM classification model, we compare the improved classification model with several popular pulmonary nodule detection classification models. These classification models are the SVM model [26], standard S4VM model, AdaBoost-SVM model, AdaBoost-S4VM model, and AdaBoost-ISSA-S4VM model. In order to evaluate the performance of the recognition model, the following performance indicators are selected in this paper. The formula for evaluating the classification indexes is shown in Table 10.

ACC is used to evaluate the accuracy of each classification model. SEN and SPE are used to refer the ability to detect the true positive and true negative, respectively. FPR and FNR are, respectively, the misdiagnosis rate and missed diagnosis rate. Table 11 records the performance indicator data of each classification model, and the best results are marked in bold. The larger the SEN and SPE are, the better the classification model performs. On the contrary, the smaller the FPR and FNR are, the better the classification model performs.

From the performance indicators data of each classification model in Table 11, we can see that the classification accuracy of AdaBoost-ISSA-S4VM classification model can be comparable to or even better than supervised classifiers such as SVM. First of all, it can be seen that the classification accuracy of the S4VM classifier is quite poor and far lower than the SVM. The reason is that the SVM is a supervised classifier whose input dataset is labeled, while S4VM is a semi-supervised classifier whose input dataset contains unlabeled dataset, which will reduce the accuracy of the classifier. Secondly, the S4VM classifier optimized by ensemble learning is better than SVM combined with ensemble learning. Then, the classification accuracy of the established AdaBoost-ISSA-S4VM, which is the S4VM classifier optimized by ensemble learning and SCA-CSSA and can get 94.22% on labeled and unlabeled lung CT images which is much higher than the original supervised classifier on labeled samples. At the same time, the false positive rate and false negative rate of the established AdaBoost-ISSA-S4VM can get 0.1234 and 0.0146 on these

TABLE 9: The normalized feature vector extraction results of Nodule.

Sample	1	2	3	4	5	...	713	714	715
1	0.0391	0.7129	0.6227	0.1787	5.4959	...	0.0093	0.0093	0.0093
2	0.0797	0.3685	0.7736	0.0049	4.2682	...	0.0187	0.0187	0.0187
3	0.0477	0.4155	0.9396	0.1177	3.6474	...	0.0008	0.0008	0.0008
4	0.0285	0.5887	0.3293	0.2242	1.7350	...	0.0011	0.0011	0.0011
5	0.0706	0.4727	0.5976	0.1127	2.1450	...	0.0083	0.0083	0.0083
6	0.0590	0.3105	0.1870	0.0029	1.9821	...	0.0119	0.0119	0.0119
7	0.0714	0.6126	0.2958	0.0535	2.6320	...	0.0162	0.0162	0.0162
...	...	...	...	...	...	...	...	...	...
123	0.2881	0.5283	0.2203	0.1387	1.2325	...	0.0827	0.0827	0.0827
124	0.0940	0.4081	0.9071	0.0471	5.1242	...	0.0153	0.0153	0.0153
125	0.2384	0.4515	0.6931	0.0203	3.6554	...	0.1235	0.1235	0.1235

TABLE 10: The calculation formulas of accuracy, sensitivity and specificity.

Evaluation index	Calculation formulas
Accuracy (ACC)	$(TP + TN)/(TP + FP + TN + FN)$
Sensitivity (SEN)	$TP/(TP + FN)$
Specificity (SPE)	$TN/(FP + TN)$
False positive rate (FPR)	$FP/(FP + TN)$
False negative rate (FNR)	$FN/(TP + FN)$

where TP stands for true positive, TN for true negative, FP for false positive, and FN for false negative.

TABLE 11: Performance of various classification model.

Classification model	ACC (%)	SEN	SPE	FPR	FNR
SVM	77.39	0.9833	0.4557	0.5443	0.0167
S4VM	68.64	0.5483	0.8962	0.1038	0.4517
AdaBoost-SVM	75.28	0.9883	0.3949	0.6051	0.0117
AdaBoost-S4VM	81.57	0.7695	0.8861	0.1139	0.2305
AdaBoost-ISSA-S4VM	94.22	0.9855	0.8766	0.1234	0.0146

lung nodule images. The false positive rate and false negative rate of AdaBoost-ISSA-S4VM also performs well. Based on above results of data, the proposed classification model is better than the traditional supervised classifiers such as SVM model on lung nodule classification.

5. Conclusion

In summary, the improved semi-supervised ensemble classifier (AdaBoost-ISSA-S4VM) is proposed by combining AdaBoost classifier, semi-supervised SVM, and improved sparrow search algorithm for semi-supervised problem. The proposed algorithm is employed in lung CT images for pulmonary nodule detection, and a detailed performance comparison and analysis are presented based on the publicly available LIDC-IDRI database. Better experimental results are obtained with the improved algorithm compared to that with the SVM, S4VM, AdaBoost-SVM, and AdaBoost-S4VM algorithms. In particular, the proposed AdaBoost-ISSA-S4VM is able to improve 21% more accuracy than standard SVM and 26% more accuracy than S4VM. This study demonstrates that the established AdaBoost-ISSA-S4VM classifier can solve the problem of pulmonary nodules detection of labeled and unlabeled lung CT images. In other words, the proposed AdaBoost-ISSA-S4VM classifier has the potential for improving the performance of the lung CT image classification by labeled and unlabeled lung CT images with a high detection probability of being cancers at its early stage.

Although the proposed AdaBoost-ISSA-S4VM has been proven to be effective in solving general optimization problems, AdaBoost-ISSA-S4VM has some shortcomings that warrant further investigation. And, in AdaBoost-ISSA-S4VM, due to the improvement strategies, AdaBoost-ISSA-S4VM has needed more time than the classical S4VM and most of supervised classifiers. Therefore, deploying the proposed algorithm to increase recognition efficiency is a worthwhile direction. In the future research work, the method presented in this paper can also be extended to solving discrete optimization problems and multiobjective optimization problems. Furthermore, applying the proposed AdaBoost-ISSA-S4VM model to other fields such as financial prediction and biomedical science diagnosis is also an interesting future work.

Data Availability

All data included in this study are available upon request by contact with the corresponding author. All the lung CT images for pulmonary nodule detection in this study can be found in the free publicly available LIDC/IDRI Database

created by the National Cancer Institute and the Foundation for the National Institutes of Health.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] F. R. Wang, Z. Chen, and G. B. Song, "Smart crawfish: a concept of underwater multi-bolt looseness identification using entropy-enhanced active sensing and ensemble learning," *Mechanical Systems and Signal Processing*, vol. 149, pp. 1–11, 2021.

[2] C. Niyomugabo, H.-R. Choi, and T. Y. Kim, "A modified Adaboost algorithm to reduce false positives in face detection," *Mathematical Problems in Engineering*, vol. 2016, Article ID 5289413, 6 pages, 2016.

[3] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[4] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithm: bagging, boosting, and variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

[5] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory (EuroCOLT)*, pp. 23–37, Berlin, Heidelberg, March 1995.

[6] X. Hu, Y. Cheng, D. Ding, and D. Chu, "Axis-guided vessel segmentation using a self-constructing cascade-AdaBoost-SVM classifier," *BioMed Research International*, vol. 2018, pp. 1–12, 2018.

[7] Y. Wu and J. Gao, "AdaBoost-based long short-term memory ensemble learning approach for financial time series forecasting," *Current Science*, vol. 115, no. 1, pp. 159–165, 2018.

[8] Y. F. Li and Z. H. Zhou, "Towards making unlabeled data never hurt," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 1, pp. 175–188, 2015.

[9] W. S. Wang, J. Q. Xi, A. Chong, and L. Li, "Driving style classification using a semisupervised support vector machine," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 650–660, 2017.

[10] Z. He, K. Xia, W. Niu, N. Aslam, and J. Hou, "Semisupervised SVM based on cuckoo search algorithm and its application," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–13, 2018.

[11] Z. Wang, L. Yao, and Y. Cai, "Mahalanobis semi-supervised mapping and beetle antennae search based support vector machine for wind turbine rolling bearings fault diagnosis," *Renewable Energy*, vol. 155, pp. 1312–1327, 2020.

[12] J. K. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science and Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[13] M. N. Rasoulizadeh and J. Rashidinia, "Numerical solution for the Kawahara equation using local RBF-FD meshless method," *Journal of King Saud University-Science*, vol. 32, no. 4, pp. 2277–2283, 2020.

[14] J. Rashidinia and M. N. Rasoulizadeh, "Numerical methods based on radial basis function-generated finite difference (RBF-FD) for solution of GKdVB equation," *Wave Motion*, vol. 90, pp. 152–167, 2019.

[15] N. Can, O. Nikan, M. N. Rasoulizadeh, H. Jafari, and Y. Gasimov, "Numerical computation of the time non-linear fractional generalized equal width model arising in shallow water channel," *Thermal Science*, vol. 24, no. 1, pp. 49–58, 2020.

[16] S. Mirjalili, "SCA: a Sine Cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.

[17] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.

[18] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, and B. Y. Qu, "Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization," *Report of the Computational Intelligence Laboratory*, Technical Report, Nanyang Technological University, Singapore and Jordan University of Science and Technology, Jordan and Zhengzhou University, Zhengzhou, China, 2016.

[19] K. James and E. Russell, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference Neural Networks*, pp. 1942–1948, Perth, Australia, November 1995.

[20] X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, "A new bio-inspired optimisation algorithm: bird Swarm algorithm," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 4, pp. 673–687, 2016.

[21] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm: crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, 2016.

[22] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[23] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in Engineering Software*, vol. 105, pp. 30–47, 2017.

[24] Q. Zhou, K. W. Xia, P. Wu, and H. Wang, "Lung nodule classification using curvelet transform, LDA algorithm and BAT-SVM algorithm," *Pattern Recognition and Image Analysis*, vol. 27, no. 4, pp. 855–862, 2017.

[25] S. Wang, K. Xia, L. Wang, J. Zhang, and H. Yang, "Improved RPCA method via non-convex regularisation for image denoising," *IET Signal Processing*, vol. 14, no. 5, pp. 269–277, 2020.

[26] P. Wu, K. Xia, and H. Yu, "Correlation coefficient based supervised locally linear embedding for pulmonary nodule recognition," *Computer Methods and Programs in Biomedicine*, vol. 136, pp. 97–106, 2016.