Research Article

Boosted Sine Cosine Algorithm with Application to Medical Diagnosis

Xiaojia Ye ⓘ,[1] Zhennao Cai ⓘ,[2] Chenglang Lu ⓘ,[3,4] Huiling Chen ⓘ,[2,4] and Zhifang Pan ⓘ[3,4]

[1]*Shanghai Lixin University of Accounting and Finance, Shanghai 201209, China*
[2]*College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou 325035, China*
[3]*College of Modern Information Technology, Zhejiang Institute of Mechanical and Electrical Engineering, Hangzhou, Zhejiang 310051, China*
[4]*The First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, China*

Correspondence should be addressed to Chenglang Lu; luchenglang@zime.edu.cn, Huiling Chen; chenhuiling.jlu@gmail.com, and Zhifang Pan; panzhifang@wmu.edu.cn

The sine cosine algorithm (SCA) was proposed for solving optimization tasks, of which the way to obtain the optimal solution is mainly through the continuous iteration of the sine and cosine update formulas. However, SCA also faces low population diversity and stagnation of locally optimal solutions. Hence, we try to eliminate these problems by proposing an enhanced version of SCA, named ESCA_PSO. ESCA_PSO is proposed based on hybrid SCA and particle swarm optimization (PSO) by incorporating multiple mutation strategies into the original SCA_PSO. To validate the effect of ESCA_PSO in handling global optimization problems, ESCA_PSO was compared with quality algorithms on various types of benchmark functions. In addition, the proposed ESCA_PSO was employed to tune the best parameters of support vector machines for dealing with medical diagnosis tasks. The results prove the efficiency of the proposed algorithms in solving optimization problems.

1. Introduction

1.1. Motivation. Many problems in real life can be summarized as global optimization problems. When it comes to increasingly complex optimization problems, traditional methods are generally unsatisfactory [1]. Therefore, many scholars began to explore new solutions. The metaheuristic algorithm (MA) is developed for obtaining and grasping information to effectively find approximately optimal solutions through learning strategies. MA has been applied to many scenarios owing to its effective optimization capability [2]. For example, MA has found the great potential in wind speed prediction [3], scheduling problem [4], parameter optimization [5], PID optimization control [6], gate resource allocation [7], fault diagnosis of rolling bearings [8], cloud workflow scheduling [9], energy vehicle dispatch [10], combination optimization problems [11], traveling salesman problem [12], object tracking [13], neural network training [14], and multiattribute decision making [15].

In 2016, a new swarm intelligence algorithm named sine cosine algorithm (SCA) [16] was proposed. SCA searches the solution based on the sine function and cosine function. It owns strong global searchability, which can significantly increase the global optimal solution through enough iterations. However, SCA also is faced with some problems, for instance, slow convergence speed, low convergence accuracy, and easily falling into local optimum. To overcome the problems existing in SCA, a hybrid SCA and PSO algorithm (SCA_PSO) was put forward by Nenavath et al. [17], which aims to solve optimization problems and target tracking. The search mechanism of the PSO algorithm is added to the traditional SCA to guide the search for potential candidate solutions. It should be noted that though the SCA_PSO has achieved promising results on object tracking when dealing with complex problems, it is still easy to skip the true optimal solution and lead to premature convergence.

According to "No Free Lunch" [18], we have introduced the differential evolution algorithm (DE) and combined

```
Input:    MaxFEs,dim,ub,lb
Output:   best_fitness
While (t ≤ MaxFEs)
      For each search agent
            Calculate fitness value
            If fitness value <SCA_P_best in the iteration
                  Move the current value to <SCA_P_best matrix
            End if
            If Fitness value <SCA_G_best
                  Set current value as <SCA_G_best
            End if
      End for
      For each search agent
            Update r_1, r_2, r_3, r_4
            If r_4 < 0.5
               Update X using Eq.(1)
            Else
               Update X using Eq.(2)
            End if
            Check and correct the new positions based on ub, lb
      End for
      For each particle
            Initialize particle with PSO_P_best
            Set PSO_G_best as SCA_G_best
      End for
      While (t ≤ MaxFEs)
            For each particle
                  Calculate fitness value
                        If fitness value <PSO_P_best in history
                              Set current value as new PSO_P_best
                        End if
            End for
      Choose the particle with the best fitness value of all the particle as the PSO_G_best
      For each particle
            v_i^{t+1} = w^t + C_1 * rand * (SCA_P_{best}^t − x_i^t) + C_2 * rand * (SCA_G_{best}^t − x_i^t)
            x_i^{t+1} = x_i^{t+1} + v_i^{t+1}
      End for
End while
best_fitness = PSO_G_best
Return best_fitness
```

PSEUDOCODE 1: The pseudocode of the SCA_PSO.

mutation strategies into the original SCA_PSO to strengthen its capability of local search and reduce the occurrence of premature convergence. The proposed ESCA_PSO was validated on a benchmark test which includes various types of functions. Experimental results demonstrated that ESCA_PSO was significantly better than SCA_PSO and other competitive counterparts. In addition, the ESCA_PSO was also used to construct an optimal support vector machine model (SVM) to deal with the medical diagnosis problems in an effective manner. In general, ESCA_PSO has improved the performance of SCA_PSO in a significant manner.

1.2. Literature Review. SCA has been widely studied and applied in many fields because of its simple implementation and relatively excellent performance in realizing complex problems. For example, SCA was employed to tackle the

scheduling problem in [19]. A multiobjective SCA was proposed to solve engineering optimization problems in [20] and forecast the wind speed in [21]. SCA was employed to predict the time series by constructing an optimal support vector regression model in [22]. In [23], SCA was utilized to optimize the parameters of fuzzy k-nearest neighbor to build an optimized classifier for predicting the intention of students for a postgraduate entrance examination. In [24], SCA was applied to optimize the SVM's parameters and the boosted classifier was trained to predict students' entrepreneurial intentions.

In addition to applications, scholars have also proposed many improved SCA variants. Issa et al. [25] proposed a new idea of SCA, that is, the combination of SCA and PSO. This algorithm was used to solve the problem of pairwise local alignment to look for the longest continuous

```
Input:    noP,MaxFEs, ub, lb, dim, fobj.
Output:   SCA_gbest, cg_curve.
While (t≤MaxFEs)
    For each search agent
        Calculate fitness agent
        If fitness value < SCA_P_best in this iteration
            Move the current value to SCA_P_best matrix
        End if
        If Fitness value < SCA_G_best
            Set current value as SCA_G_best
        End if
    End for
    For each search agent
        Update r_1, r_2, r_3, r_4
        If r_4 < 0.5
            Update X using Eq.(1)
        Else
            Update X using Eq.(2)
        End if
        Check and correct the new positions based on ub, lb
    End for
    For each particle
        Initialize particle with PSO_P_best
        Set PSO_G_best as SCA_G_best
    End for
    Perform DE strategy
    While (t PSO ≤ PSOMaxFEs)
        For each particle
            Calculate fitness value
                If fitness value <PSO_P_best in history
                    Set current value as new PSO_P_best
                End if
        End for
    Choose the particle with the best fitness value of all the particle as the PSO_G_best
    For each particle
        Update v(k) with Eq. (9)
        Update v(k) with Eq. (10)
    End for
    End while
    Update X_m_Levy using Eq. (14)
    Update X_m_gaus using Eq. (12)
    Update X_m_cauchy using Eq. (13)
End while
Return SCAgbest and cg_curve
```

PSEUDOCODE 2: Pseudocode of the ESCA_PSO.

substring between two biological sequences, which has shown good performance in accuracy and calculation time. Nenavath and Jatoth [26] proposed to select the DE algorithm to merge into SCA and applied it to target tracking. Abd Elaziz et al. [27] mutated SCA (OBSCA) with opposition-based learning mechanisms, which has effectively boosted SCA's search efficiency and expanded its search scope. Qu et al. [28] improved the SCA by adding three strategies. Kumar et al. [29] tried to mix SCA with Cauchy and Gaussian distributions, which were named CGSCA. Simulations showed that the single-sensor tracking scheme based on CGSCA had better results in terms of tracking time and tracking efficiency. Long et al. [30] combined SCA with inertia weight based on a position updating equation and a nonlinear conversion parameter strategy to ameliorate SCA's ability in solving high-dimensional problems. Tu et al. [24] proposed to adopt the chaotic local search enhanced SCA for training an optimal support vector machine to predict students' entrepreneurial intentions. Turgut [31] proposed that mixing SCA with a backtracking search algorithm (BSA) was an efficient way to realize the shell and tube evaporator's optimal design. With the proposed optimization method, the optimal values of the total design cost of the heat exchanger and the total heat transfer
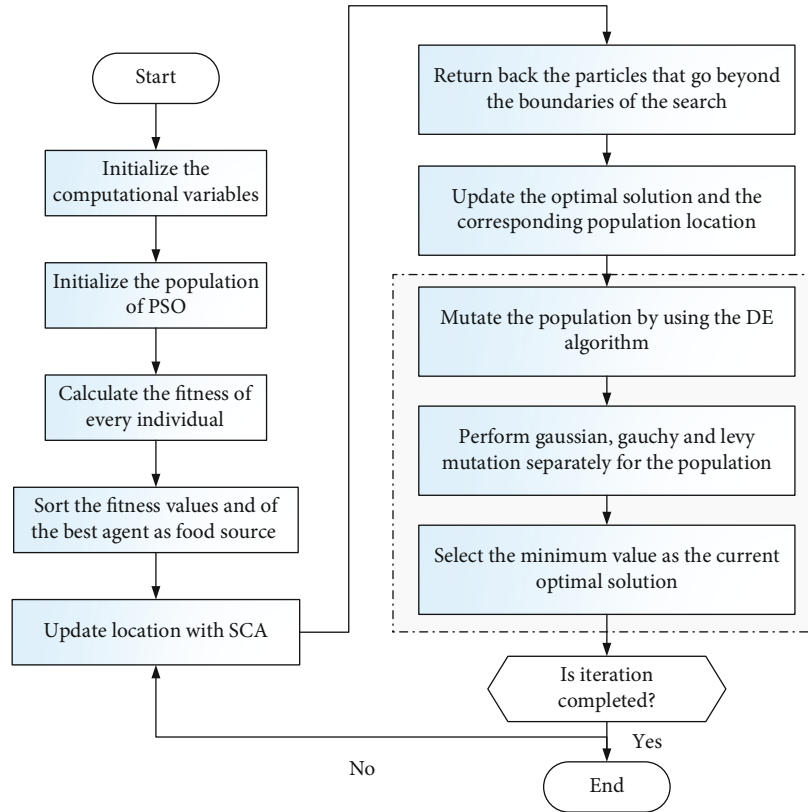
Figure 1: Flowchart of ESCA_PSO.

coefficient were better than the results of the other optimizers in the literature. Guo et al. [32] introduced the Riesz fractional derivative and the OBL strategy into SCA and applied the method to deal with the engineering problems. Gupta and Deep [33] proposed to add a leading guidance mechanism and simulated quenching algorithm together to SCA and applied it to train multilayer perceptron. Gupta and Deep [34] utilized the OBL strategy and the self-adaptive component to enhance the SCA. And this improved algorithm's efficacy was confirmed by lots of benchmark problems and engineering problems. Tawhid and Savsani [20] proposed a novel and effective multiobjective version of SCA (MO-SCA). The difference between MO-SCA and SCA is mainly reflected in two aspects, one is to improve the nondominated level by adding an elite nondominated sorting strategy, and the other is to maintain the diversity of optimal solutions by adding crowded distance method. At present, machine learning methods are still a research hotspot [35–42]. However, the hyperparameters of the model have a crucial impact on its performance. Therefore, combining the improved version of SCA with machine learning methods to obtain the best hyperparameter combination model is also a novel research angle.

1.3. Contribution and Paper Organization. The main contributions of this study are as follows:

(a) An improved SCA named ESCA_PSO is proposed

(b) The proposed ESCA_PSO has achieved superior performance to other peers on function optimization and machine learning tasks

(c) Successfully applied ESCA_PSO to SVM parameter optimization problem

(d) Successfully use ESCA_PSO-SVM in the field of medical diagnosis

The rest part of this paper is assigned as follows: The introduction of SCA and SCA-PSO is arranged in Section 2. Our proposed ESCA_PSO is presented in Section 3. The details information on experimental results and discussions are described in Section 4. Finally, in Section 5, conclusions and future works are summarized.

2. An Overview of SCA_PSO Algorithm

2.1. Standard SCA. In recent years, many new intelligent algorithms have been proposed to solve practical problems, such as hunger games search (HGS) [43], weighted mean of vectors (INFO) [44], Harris hawks optimization (HHO) [45], slime mould algorithm (SMA) [46], and Runge Kutta optimizer (RUN) [47]. These algorithms have shown great potential in various fields such as engineering, medicine, energy, finance, and education. In 2016, Mirjalili [16] put forward a novel swarm intelligence algorithm called SCA for global optimization tasks. Similar to other metaheuristics, it looks for the

TABLE 1: Description of 30 benchmark functions.

ID	Function equation	Range	Optimum value
Unimodal functions			
F1	Rotated high conditioned elliptic function	[-100, 100]	$f_1\{X_{\min}\} = 100$
F2	Rotated bent cigar function	[-100, 100]	$f_2\{X_{\min}\} = 200$
F3	Rotated discus function	[-100, 100]	$f_3\{X_{\min}\} = 300$
Simple multimodal functions			
F4	Shifted and rotated Rosenbrock's function	[-100, 100]	$f_4\{X_{\min}\} = 400$
F5	Shifted and rotated Ackley's function	[-100, 100]	$f_5\{X_{\min}\} = 500$
F6	Shifted and rotated Weierstrass function	[-100, 100]]	$f_6\{X_{\min}\} = 600$
F7	Shifted and rotated Griewank's function	[-100, 100]	$f_7\{X_{\min}\} = 700$
F8	Shifted Rastrigin's function	[-100, 100]	$f_8\{X_{\min}\} = 800$
F9	Shifted and rotated Rastrigin's function	[-100, 100]	$f_9\{X_{\min}\} = 900$
F10	Shifted Schwefel's function	[-100, 100]	$f_{10}\{X_{\min}\} = 1000$
F11	Shifted and rotated Schwefel's function	[-100, 100]	$f_{11}\{X_{\min}\} = 1100$
F12	Shifted and rotated Katsuura function	[-100, 100]	$f_{12}\{X_{\min}\} = 1200$
F13	Shifted and rotated HappyCat function	[-100, 100]	$f_{13}\{X_{\min}\} = 1300$
F14	Shifted and rotated HGBat function	[-100, 100]	$f_{14}\{X_{\min}\} = 1400$
F15	Shifted and rotated expanded Griewank's plus Rosenbrock's function	[-100, 100]	$f_{15}\{X_{\min}\} = 1500$
F16	Shifted and rotated expanded Scaffer's F6 function	[-100, 100]	$f_{16}\{X_{\min}\} = 1600$
Hybrid functions			
F17	Hybrid function 1 ($N = 3$)	[-100, 100]	$f_{15}\{X_{\min}\} = 1700$
F18	Hybrid function 2 ($N = 3$)	[-100, 100]	$f_{16}\{X_{\min}\} = 1800$
F19	Hybrid function 3 ($N = 4$)	[-100, 100]	$f_{17}\{X_{\min}\} = 1900$
F20	Hybrid function 4 ($N = 4$)	[-100, 100]	$f_{18}\{X_{\min}\} = 2000$
F21	Hybrid function 5 ($N = 5$)	[-100, 100]	$f_{19}\{X_{\min}\} = 2100$
F22	Hybrid function 6 ($N = 5$)	[-100, 100]	$f_{22}\{X_{\min}\} = 2200$
Composition functions			
F23	Composition function 1 ($N = 5$)	[-100, 100]	$f_{23}\{X_{\min}\} = 2300$
F24	Composition function 2 ($N = 3$)	[-100, 100]	$f_{24}\{X_{\min}\} = 2400$
F25	Composition function 3 ($N = 3$)	[-100, 100]	$f_{25}\{X_{\min}\} = 2500$
F26	Composition function 4 ($N = 5$)	[-100, 100]	$f_{26}\{X_{\min}\} = 2600$
F27	Composition function 5 ($N = 5$)	[-100, 100]	$f_{27}\{X_{\min}\} = 2700$
F28	Composition function 6 ($N = 5$)	[-100, 100]	$f_{28}\{X_{\min}\} = 2800$
F29	Composition function 7 ($N = 3$)	[-100, 100]	$f_{29}\{X_{\min}\} = 2900$
F30	Composition function 8 ($N = 3$)	[-100, 100]	$f_{30}\{X_{\min}\} = 3000$

solution in a random searching space. SCA obtains the optimal solution through triangle sine cosine functions [16].

The following location updating formulas are proposed for two phases:

$$X_i^{t+1} = X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t|, \qquad (1)$$

$$X_i^{t+1} = X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t|, \qquad (2)$$

where X_i^t is the position of the current solution in i-th

dimension at t-th iteration, r_1, r_2, r_3 are random numbers, P_i is the position of the target point in i-th dimension, and $||$ indicates absolute value.

Combining the above, the following equation can be obtained:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 < 0.5, \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - X_i^t|, & r_4 \geq 0.5, \end{cases} \qquad (3)$$

where r_4 is a random number in the range [0, 1].

TABLE 2: Parameters setting for involved algorithms.

Method	Parameter
ESCA_PSO	$a = 2$; $c_1 = 2$; $c_2 = 2$; $v_{max} = 6$
SCADE	cmin = 0.2; cmax = 0.8; pCR = 0.8
SCA	$a = 2$
SCA-PSO	$M = 4$; $N = 9$; vmax = 6; wmax = 0.9; wmin = 0.2
CGSCA	Delta = 0.1
GWO	$a = [0, 2]$
MFO	$b = 1$; $t = [-1, 1]$; $a = [-1, -2]$
BA	$a = 0.5$; $r = 0.5$
PSO	$c_1 = 2$; $c_2 = 2$; $v_{max} = 6$
LSHADE	Pb = 0.1; Ar = 1.4

As the above formulae reveal, r_1, r_2, r_3 and r_4 are four important parameters with different meanings. The r_1 mainly decides whether the position of the next move is within the boundary range of solution and destination or outside the range. r_2 represents the distance required for the movement to reach the destination. The coefficient r_3 carries a random load for the destination to stochastically emphasize ($r_3 > 1$) or deemphasize ($r_3 < 1$) the effect of destination in describing the distance. r_4 switches from sine function to cosine function or vice versa in Eq. (3). The values of r_2 were set in $[0, 2\pi]$ in this study.

To achieve a steady global and local search, the range of sine and cosine functions in Eqs. (1)–(3) are altered adaptively according to the following formula:

$$r_1 = a - t\frac{a}{T}, \tag{4}$$

where t means the current iteration, T means the maximum number of iterations, and a is a constant with the value of 2.

2.2. SCA_PSO Algorithm. To eliminate the shortcoming of SCA, Nenavath et al. [17] came up with an improved hybrid SCA_PSO for dealing with optimization tasks. The SCA_PSO integrates PSO's strengths in exploitation and the SCA's strengths in exploration to approach the global optimal solution.

By adding internal storage to SCA, each individual is permitted to follow the coordinates associated with the adaptive values in the search space. And the personal historical best solution of each search agent in the present population is stored in the form of a matrix, SCA-P_{best}, which is the same as the concept of P_{best} in PSO each iteration. In addition, the solution also keeps track of the optimal value achieved so far by any nearby solution. As a new concept in SCA, P_{best} and G_{best} enhance the exploitation ability of SCA. The pseudocode of SCA_PSO is shown in Pseudocode 1.

3. Proposed ESCA_PSO

The improved ESCA_PSO is combined with two efficient strategies. Firstly, a DE with the "random variation" mode is successfully combined with the SCA_PSO to strengthen the global search capability of the SCA_PSO. Then, a combined mutation with the mixed distributions of Gaussian, Cauchy, and Lévy was added to the combination mutation strategy, which can further improve the accuracy of the solution.

3.1. Combined Mutation of Gaussian, Cauchy, and Lévy. Gaussian distribution (GD) [48] is a significant probability distribution in many subjects such as engineering and mathematics. GD has many excellent features. Plenty of random variables and objects in nature can be presumably expressed as GD, and many probability distributions can be approximated or exported by this distribution. The probability density function of the GD can be expressed according to Eq. (5):

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{5}$$

where μ and σ represent the mean and standard deviation, respectively.

Cauchy distribution [49] is also called Cauchy-Lorentz distribution. It is a continuous probability distribution named after Augustine-Louis-Cauchy and Hendrick-Lorentz. It is similar to normal distribution. Cauchy distribution is also widely used in statistics. It has the characteristics of the nonexistence of expectation and variance and additivity. The probability density function of the Cauchy distribution can be expressed according to Eq. (6):

$$f(x; x_0, y) = \frac{1}{\pi}\left(\frac{\gamma}{(x-x_0)^2 + \gamma^2}\right), \tag{6}$$

where x_0 is the position parameter defining the location of the distribution peak; γ is the scale parameter defining half the width of the maximum half.

Lévy flights [50] based on Lévy distribution are consistent with the search behavior of many organisms in the nature and are widely used in optimization algorithms and optimal search processes. Moreover, the stochastic process can maximize the search efficiency of resources under uncertain conditions. In the search process, Lévy flights can make the whole search process more effective and stable, balancing the proportion of local search and global search. Due to the existence of the random process of the Lévy flight, the short-range exploratory local search and the occasional long-distance walk are in phase. Thus, the algorithm's local searching speed is faster, and the solutions are more easily searched near the current optimum. The decomposition can search far away from the current optimal value, thus ensuring that the algorithm does not fall into the local optimum (LO). The Lévy formula used is as follows:

$$\sigma = \left(\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[1+\beta/2]\beta 2^{\beta-1/2}}\right)^{(1/\beta)}, \tag{7}$$

where $\Gamma(x)$ is a continuation function of the factorial, that is, when the x is a positive integer $\Gamma(x+1) = x!$ The value

TABLE 3: Results of ESCA_PSO versus other algorithms on 30 benchmark functions.

| | F1 | | F2 | | F3 | |
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	8.77E+06	3.47E+06	7.49E+07	7.38E+06	9.44E+03	2.27E+03
SCADE	4.23E +08	9.19E+07	2.97E+10	4.56E+09	5.46E+04	6.21E+03
SCA	2.52E+08	7.02E+07	1.76E+10	2.68E+09	3.72E+04	5.71E+03
SCA_PSO	8.01E+06	2.83E+06	3.83E+07	8.40E+06	2.99E+03	1.04E+03
CGSCA	2.85E+08	8.98E+07	1.82E+10	4.06E+09	4.15E+04	5.63E+03
GWO	5.76E+07	4.69E+07	2.18E+09	2.05E+09	3.24E+04	8.29E+03
MFO	8.50E+07	1.06E+08	1.22E+10	7.01E+09	1.10E+05	6.34E+04
BA	7.40E+05	2.88E+05	5.24E+05	2.80E+05	4.11E+02	1.40E+02
PSO	9.30E+06	2.45E+06	1.46E+08	1.63E+07	9.60E+02	9.65E+01
LSHADE	4.41E+03	7.34E+03	2.00E+02	3.77E-14	3.00E+02	1.74E-11

| | F4 | | F5 | | F6 | |
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	4.73E+02	3.72E+01	5.21E+02	4.33E-02	6.29E+02	3.45E+00
SCADE	2.40E+03	5.90E+02	5.21E+02	3.58E-02	6.35E+02	1.86E+00
SCA	1.43E+03	2.20E+02	5.21E+02	4.98E-02	6.34E+02	2.32E+00
SCA_PSO	4.77E+02	4.19E+01	5.21E+02	6.37E-02	6.29E+02	3.03E+00
CGSCA	1.63E+03	2.92E+02	5.21E+02	5.14E-02	6.34E+02	2.33E+00
GWO	7.02E+02	2.37E+02	5.21E+02	5.08E-02	6.13E+02	2.06E+00
MFO	1.34E+03	8.51E+02	5.20E+02	1.59E-01	6.24E+02	3.73E+00
BA	4.40E+02	3.98E+01	5.21E+02	5.19E-02	6.34E+02	3.36E+00
PSO	4.68E+02	3.39E+01	5.21E+02	4.82E-02	6.23E+02	2.99E+00
LSHADE	4.04E+02	1.61E+01	5.20E+02	1.00E-03	6.11E+02	2.23E+00

| | F7 | | F8 | | F9 | |
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	7.02E+02	5.88E-02	9.83E+02	2.14E+01	1.13E+03	2.09E+01
SCADE	9.10E+02	3.53E+01	1.07E+03	1.63E+01	1.21E+03	1.63E+01
SCA	8.43E+02	2.36E+01	1.04E+03	2.05E+01	1.18E+03	1.64E+01
SCA_PSO	7.01E+02	7.70E-02	9.97E+02	3.65E+01	1.13E+03	3.53E+01
CGSCA	8.67E+02	2.74E+01	1.06E+03	1.61E+01	1.18E+03	1.59E+01
GWO	7.23E+02	1.75E+01	8.81E+02	1.77E+01	9.99E+02	2.43E+01
MFO	8.15E+02	7.39E+01	9.36E+02	2.91E+01	1.12E+03	5.57E+01
BA	7.01E+02	1.55E-01	1.03E+03	4.75E+01	1.20E+03	6.92E+01
PSO	7.02E+02	1.62E-01	9.71E+02	2.31E+01	1.12E+03	3.57E+01
LSHADE	7.00E+02	1.17E-02	8.00E+02	1.82E-01	9.36E+02	1.36E+01

| | F10 | | F11 | | F12 | |
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	4.71E+03	6.56E+02	5.69E+03	6.28E+02	1.20E+03	3.92E-01
SCADE	7.41E+03	3.71E+02	8.10E+03	3.58E+02	1.20E+03	2.15E-01
SCA	6.93E+03	4.80E+02	8.10E+03	3.38E+02	1.20E+03	2.81E-01
SCA_PSO	5.26E+03	7.19E+02	5.62E+03	7.81E+02	1.20E+03	3.01E-01
CGSCA	7.19E+03	3.88E+02	8.05E+03	3.20E+02	1.20E+03	2.86E-01
GWO	3.34E+03	4.08E+02	3.93E+03	4.80E+02	1.20E+03	8.77E-01
MFO	4.56E+03	9.40E+02	5.40E+03	8.30E+02	1.20E+03	2.55E-01
BA	5.16E+03	6.27E+02	5.43E+03	6.13E+02	1.20E+03	3.10E-01
PSO	4.97E+03	5.63E+02	5.70E+03	5.81E+02	1.20E+03	2.82E-01
LSHADE	1.00E+03	7.66E-01	2.91E+03	4.12E+02	1.20E +03	3.80E-02

TABLE 3: Continued.

	F13		F14		F15	
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	1.30E+03	9.71E-02	1.40E+03	3.81E-02	1.52E+03	1.41E+00
SCADE	1.30E+03	3.33E-01	1.49E+03	1.18E+01	2.17E+04	7.85E+03
SCA	1.30E+03	3.27E-01	1.45E+03	8.31E+00	5.16E+03	3.72E+03
SCA_PSO	1.30E+03	6.25E-02	1.40E+03	4.86E-02	1.52E+03	1.49E+00
CGSCA	1.30E+03	3.16E-01	1.45E+03	1.06E+01	8.10E+03	5.63E+03
GWO	1.30E+03	3.97E-01	1.41E+03	7.06E+00	1.82E+03	8.98E+02
MFO	1.30E+03	1.45E+00	1.43E+03	2.16E+01	1.28E+05	3.08E+05
BA	1.30E+03	1.38E-01	1.40E+03	4.97E-02	1.53E+03	6.26E+00
PSO	1.30E+03	7.09E-02	1.40E+03	1.19E-01	1.52E+03	1.11E+00
LSHADE	1.30E+03	7.43E-02	1.40E+03	9.17E-02	1.51E+03	1.76E+00
	F16		F17		F18	
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	1.61E+03	3.67E-01	4.35E+05	2.41E+05	1.93E+06	6.44E+05
SCADE	1.61E+03	2.26E-01	1.58E+07	5.49E+06	1.54E+08	9.16E+07
SCA	1.61E+03	2.37E-01	5.52E+06	2.43E+06	1.74E+08	7.49E+07
SCA_PSO	1.61E+03	4.47E-01	1.79E+05	1.09E+05	7.62E+05	4.43E+05
CGSCA	1.61E+03	2.29E-01	6.77E +06	3.75E+06	1.43E+08	7.47E+07
GWO	1.61E+03	7.29E-01	1.49E+06	1.54E+06	2.88E+06	1.02E+07
MFO	1.61E+03	4.74E-01	2.40E+06	3.10E+06	2.64E+07	9.98E+07
BA	1.61E+03	2.59E-01	1.09E+05	8.01E+04	9.42E+04	4.63E+04
PSO	1.61E+03	5.31E-01	2.76E+05	1.21E+05	2.19E+06	7.43E+05
LSHADE	1.61E+03	3.85E-01	1.49E+04	6.33E+04	1.95E+03	5.46E+01
	F19		F20		F21	
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	1.92E+03	2.29E+00	3.81E+03	1.57E+03	1.29E+05	1.10E+05
SCADE	2.02E+03	1.98E+01	2.52E+04	1.06E+04	2.18E+06	1.12E+06
SCA	1.99E+03	2.80E+01	1.56E+04	3.39E+03	1.23E+06	6.00E+05
SCA_PSO	1.92E+03	2.72E+00	2.47E+03	2.22E+02	9.35E+04	4.63E+04
CGSCA	1.99E+03	1.65E+01	1.88E+04	5.54E+03	1.67E+06	7.22E+05
GWO	1.94E+03	2.47E+01	1.97E+04	1.35E+04	6.56E+05	1.16E+06
MFO	1.96E+03	4.95E+01	7.19E+04	8.18E+04	8.20E+05	1.10E+06
BA	1.92E+03	1.83E+01	2.36E+03	1.13E+02	5.26E +04	3.10E+04
PSO	1.92E+03	2.43E+00	2.32E+03	7.07E+01	1.17E+05	6.92E+04
LSHADE	1.91E+03	1.84E+00	3.11E+03	3.51E+03	2.78E+03	2.77E+02
	F22		F23		F24	
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	2.97E+03	2.43E+02	2.50E+03	0.00E+00	2.60E+03	0.00E+00
SCADE	3.11E+03	1.85E+02	2.50E+03	0.00E+00	2.60E+03	5.15E-06
SCA	2.99E+03	1.42E+02	2.67E+03	1.07E+01	2.60E+03	1.00E-01
SCA_PSO	3.17E+03	2.89E+02	2.50E+03	0.00E+00	2.60E+03	0.00E+00
CGSCA	3.07E+03	1.26E+02	2.50E+03	0.00E+00	2.60E+03	3.55E-06
GWO	2.59E+03	1.83E+02	2.63E+03	9.77E+00	2.60E+03	9.56E-04
MFO	3.12E+03	2.90E+02	2.66E+03	2.48E+01	2.68E+03	2.64E+01
BA	3.33E+03	3.17E+02	2.62E+03	3.12E-03	2.67E+03	3.46E+01
PSO	2.90E+03	2.61E+02	2.62E+03	7.09E-01	2.63E+03	5.18E+00
LSHADE	2.43E+03	9.48E+01	2.62E+03	1.88E-12	2.64E+03	6.32E+00

TABLE 3: Continued.

| | F25 | | F26 | | F27 | |
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	2.70E+03	0.00E+00	2.70E+03	1.27E-01	2.90E+03	0.00E+00
SCADE	2.70E+03	0.00E+00	2.70E+03	4.66E-01	3.20E+03	2.10E+02
SCA	2.73E+03	8.20E+00	2.70E+03	5.56E-01	3.57E+03	3.51E+02
SCA_PSO	2.70E+03	0.00E+00	2.77E+03	4.48E+01	2.90E+03	0.00E+00
CGSCA	2.70E+03	0.00E+00	2.70E+03	3.72E-01	2.90E+03	0.00E+00
GWO	2.71E+03	3.94E+00	2.74E+03	6.01E+01	3.32E+03	1.24E+02
MFO	2.72E+03	8.77E+00	2.70E+03	1.42E+00	3.66E+03	1.76E+02
BA	2.73E+03	1.44E+01	2.70E+03	1.35E-01	3.91E+03	4.20E+02
PSO	2.71E+03	5.98E+00	2.78E+03	4.07E+01	3.39E+03	2.97E+02
LSHADE	2.71E+03	3.23E+00	2.71E+03	3.04E+01	3.25E+03	9.56E+01
	F28		F29		F30	
	Avg	Std	Avg	Std	Avg	Std
ESCA_PSO	3.00E+03	0.00E+00	3.19E+03	8.63E+01	3.88E+03	6.83E+02
SCADE	4.96E+03	9.21E+02	1.91E+07	1.01E+07	4.90E+05	1.70E+05
SCA	4.84E+03	2.97E+02	1.37E+07	7.32E+06	2.39E+05	7.38E+04
SCA_PSO	3.00E+03	0.00E+00	1.26E+04	3.99E+04	1.53E+04	1.08E+04
CGSCA	3.00E+03	0.00E+00	3.10E+03	0.00E+00	3.20E+03	0.00E+00
GWO	3.84E+03	2.03E+02	6.26E+05	2.36E+06	4.88E+04	4.04E+04
MFO	3.98E+03	2.29E+02	3.06E+06	3.58E+06	5.65E+04	4.66E+04
BA	5.29E+03	6.59E+02	3.13E+07	3.38E+07	1.30E+04	1.23E+04
PSO	7.32E+03	8.69E+02	9.61E+04	1.83E+05	1.44E+04	7.29E+03
LSHADE	3.73E+03	7.27E+01	3.68E+03	4.12E+01	5.47E+03	1.26E+03

of the variable β determines the shape of the Lévy probability density function, especially in the tail area.

$$\gamma = \frac{\mu}{|\nu|^{1/\beta}}, \tag{8}$$

where γ is the flight step, ν, is the standard normal distribution, and μ is a normal distribution with the mean of 0 and the variance of σ^2.

3.2. Combined Mutation Strategy. As mentioned in the foregoing, the DE algorithm can enhance the global search capability of SCA_PSO, while mutation of Gaussian, Cauchy, and Lévy is algorithms that can make the algorithm perform better in local search. A combined mutation strategy was proposed in this study, which combined the different characteristics of the three mutation strategies, making the SCA_PSO find a more balanced manner in performing explorative search and exploitative search. The whole algorithm steps are as shown below:

Step 1. Search using the mentioned SCA.

Step 2. Mutate the SCA_PSO by the DE algorithm and the new individual will be retained if its fitness value is better than the original one.

Step 3. Update using the formula of SCA_PSO. The update formula used is as follows:

$$v(k) = w \bullet v(k) + c1 \bullet r2 \bullet \left(p\text{best} - x_i^k \right) + c2 \bullet r3 \bullet \left(SCA_{G_{\text{best}_{x_i^k}}} \right), \tag{9}$$

$$x_i^k = x_i^k + v(k), \tag{10}$$

where $c1$, $c2 = 2$. $r2$, and $r3$ are in the range of [0, 1].

Step 4. Use combined mutation of Gaussian, Cauchy, and Lévy to mutate the current optimal individuals, find the individuals with the smallest of the three results, and update the fitness values and corresponding individuals.

$$X = \min \{X\_m\_\text{Levy}, X\_m\_\text{gaus}, X\_m\_\text{cauchy}\}, \tag{11}$$

$$X_{m_{\text{gaus}}} = X_i^t \bullet (1 + k \bullet \text{randn}), \tag{12}$$

$$X_{m_{\text{cauchy}}} = X_i^t \bullet (1 + k \bullet \text{cauchy}), \tag{13}$$

$$X_{m_{\text{Levy}}} = X_i^t \bullet (1 + k \bullet \text{Levy}(1)), \tag{14}$$

where $X\_m\_$Lévy, $X\_m\_$gaus, and $X\_m\_$cauchy are the values obtained by the Lévy, Gaussian, and Cauchy strategies, respectively.

TABLE 4: The p value of ESCA_PSO compared with other algorithms.

(a)

F	ESCA_PSO	SCADE		SCA		SCA_PSO		CGSCA		GWO	
F1	N/A	1.73E-06	+	1.73E-06	+	2.80E-01		1.73E-06	+	2.88E-06	+
F2	N/A	1.73E-06	+	1.73E-06	+	1.73E-06	—	1.73E-06	+	1.73E-06	+
F3	N/A	1.73E-06	+	1.73E-06	+	1.92E-06	—	1.73E-06	+	1.73E-06	+
F4	N/A	1.73E-06	+	1.73E-06	+	8.77E-01		1.73E-06	+	1.92E-06	+
F5	N/A	1.78E-01		1.25E-02	—	6.27E-02		1.48E-02	—	1.59E-01	
F6	N/A	3.52E-06	+	1.02E-05	+	4.53E-01		1.24E-05	+	1.73E-06	—
F7	N/A	1.73E-06	+	1.73E-06	+	1.73E-06	—	1.73E-06	+	1.73E-06	+
F8	N/A	1.73E-06	+	1.92E-06	+	1.53E-01		1.73E-06	+	1.73E-06	—
F9	N/A	1.73E-06	+	2.35E-06	+	5.30E-01		1.73E-06	+	1.73E-06	—
F10	N/A	1.73E-06	+	1.73E-06	+	9.84E-03	+	1.73E-06	+	1.73E-06	—
F11	N/A	1.73E-06	+	1.73E-06	+	6.88E-01		1.73E-06	+	1.73E-06	—
F12	N/A	2.37E-05	+	2.58E-03	+	8.94E-01		4.53E-04	+	3.18E-01	
F13	N/A	1.73E-06	+	1.73E-06	+	4.99E-03	—	1.73E-06	+	2.43E-02	—
F14	N/A	1.73E-06	+	1.73E-06	+	2.70E-02	—	1.73E-06	+	5.22E-06	+
F15	N/A	1.73E-06	+	1.73E-06	+	5.44E-01		1.73E-06	+	4.86E-05	+
F16	N/A	3.88E-06	+	1.73E-06	+	1.96E-03	+	1.92E-06	+	2.88E-06	—
F17	N/A	1.73E-06	+	1.73E-06	+	1.48E-04	—	1.73E-06	+	7.71E-04	+
F18	N/A	1.73E-06	+	1.73E-06	+	2.35E-06	—	1.73E-06	+	2.07E-02	+
F19	N/A	1.73E-06	+	1.73E-06	+	7.97E-01		1.73E-06	+	1.74E-04	+
F20	N/A	1.73E-06	+	1.73E-06	+	1.92E-06	—	1.73E-06	+	2.88E-06	+
F21	N/A	1.73E-06	+	1.73E-06	+	7.19E-02		1.73E-06	+	1.15E-04	+
F22	N/A	6.42E-03	+	3.18E-01		4.11E-03	+	1.36E-01		7.69E-06	—
F23	N/A	1.00E+00		1.73E-06	+	1.00E+00		1.00E+00		1.73E-06	+
F24	N/A	1.56E-02	+	1.73E-06	+	1.00E+00		2.44E-04	+	1.73E-06	+
F25	N/A	1.00E+00		1.73E-06	+	1.00E+00		1.00E+00		3.79E-06	+
F26	N/A	1.73E-06	+	1.73E-06	+	1.64E-05	+	1.73E-06	+	4.68E-03	+
F27	N/A	5.96E-05	+	1.73E-06	+	1.00E+00		1.00E+00		1.73E-06	+
F28	N/A	1.23E-05	+	1.73E-06	+	1.00E+00		1.00E+00		1.73E-06	+
F29	N/A	1.73E-06	+	1.73E-06	+	5.11E-03	+	5.61E-06	—	1.73E-06	+
F30	N/A	1.73E-06	+	1.73E-06	+	6.75E-05	+	1.32E-04	—	1.73E-06	+
+/-/=	/	27/0/3		28/1/1		6/8/10		22/3/5		20/8/2	
Avg	2	10		9		3		7		5	

(b)

F	MFO		BA		PSO		LSHADE	
F1	3.18E-06	+	1.73E-06	—	2.99E-01		1.73E-06	—
F2	1.73E-06	+	1.73E-06	—	1.73E-06	+	1.73E-06	—
F3	1.73E-06	+	1.73E-06	—	1.73E-06	—	1.73E-06	—
F4	1.73E-06	+	1.57E-02	—	9.75E-01		1.73E-06	—
F5	1.73E-06	—	8.45E-01		1.53E-01		1.73E-06	—
F6	1.15E-04	—	3.06E-04	+	7.69E-06	—	1.73E-06	—
F7	1.73E-06	+	1.73E-06	—	1.73E-06	+	1.73E-06	—
F8	1.64E-05	—	2.83E-04	+	5.45E-02		1.73E-06	—
F9	2.21E-01		7.51E-05	+	2.80E-01		1.73E-06	—
F10	4.53E-01		4.68E-03	+	1.85E-01		1.73E-06	—
F11	1.92E-01		1.11E-01		9.26E-01		1.73E-06	—

Table 4: Continued.

F	MFO		BA		PSO		LSHADE	
F12	1.73E-06	—	1.73E-06	—	4.20E-04	+	1.73E-06	—
F13	1.73E-06	+	6.56E-02		7.16E-04	—	1.80E-05	—
F14	1.73E-06	+	1.64E-05	+	1.48E-02	+	8.19E-05	+
F15	1.73E-06	+	1.92E-06	+	5.32E-03	—	1.73E-06	—
F16	4.29E-06	+	1.73E-06	+	5.30E-01		1.73E-06	—
F17	2.41E-03	+	2.88E-06	—	2.41E-03	—	2.13E-06	—
F18	3.59E-04	+	1.73E-06	—	1.85E-01		1.73E-06	—
F19	2.05E-04	+	6.87E-02		5.45E-02		1.73E-06	—
F20	1.73E-06	+	1.92E-06	—	1.92E-06	—	3.59E-04	—
F21	1.36E-04	+	2.84E-05	—	7.19E-01		1.73E-06	—
F22	4.72E-02	+	1.15E-04	+	2.21E-01		1.73E-06	—
F23	1.73E-06	+	1.73E-06	+	1.73E-06	+	4.32E-08	+
F24	1.73E-06	+	1.73E-06	+	1.73E-06	+	1.73E-06	+
F25	1.73E-06	+	1.73E-06	+	1.73E-06	+	1.73E-06	+
F26	1.73E-06	+	7.52E-02		1.24E-05	+	1.02E-01	
F27	1.73E-06	+	1.73E-06	+	1.73E-06	+	1.73E-06	+
F28	1.73E-06	+	1.73E-06	+	1.73E-06	+	1.73E-06	+
F29	1.73E-06	+	1.73E-06	+	8.73E-03	+	1.73E-06	+
F30	1.73E-06	+	1.73E-06	+	1.73E-06	+	4.29E-06	+
+/-/=	23/4/3		15/10/5		12/6/12		8/21/1	
Avg	8		6		4		1	

Pseudocode 2 and Figure 1 display the detailed steps and flowchart of ESCA_PSO, respectively.

First of all, the current optimal individual SCA_G_{best} is obtained by SCA. Then, the particle swarm population is initialized with the help of SCA_G_{best} and mutated with the help of DE strategy. Next, the population is updated using SCA_PSO. Finally, it is updated by Gaussian, Cauchy, and Lévy flight strategies.

3.3. Complexity Analysis. The time complexity of the ESCA_PSO is mainly related to the number of four factors, which are algorithm iterations (T), PSO's iterations (P), population (N), and dimensions (D). And the whole time complexity is analyzed as follows: $O(\text{ESCA\_PSO}) = O(\text{initialize}) + T \times (O(\text{calculate the fitness of population}) + O(\text{update location with SCA}) + P \times (O(\text{calculate the fitness of population}) + O(\text{update location with PSO})) + O(\text{perform combined mutation strategy}))$. The time complexity of initialization is $O(N \times D)$. Since there are N individuals, the fitness of the initial populations is $O(N)$. Updating location with SCA is $O(N)$. Updating location with PSO is $O(N \times D)$. Performing the combined mutation strategy is $O(N)$. All in all, it is not difficult to conclude that the total time complexity of ESCA_PSO is $O(\text{ESCA\_PSO}) = O(N \times D) + T \times (O(N) + O(N) + P \times (O(N) + O(N \times D)) + O(N)) = O(N \times D) + T \times (2O(N) + P \times (O(N) + O(N \times D)) + O(N))$.

4. Experimental Results

To confirm the effectiveness of ESCA_PSO, the proposed ESCA_PSO is compared with other competitive metaheuristic algorithms on 30 functions of CEC2014 in this part. And then ESCA_PSO carries on the variation mechanism contrast experiment. Finally, ESCA_PSO is used for tuning SVM's parameters for medical diagnosis purposes.

4.1. Benchmark Functions. This experiment used 30 classical functions to substantiate the proposed method and other competitors. These functions include unimodal, multimodal, composition, and hybrid functions. F1-F3 are unimodal functions, F4-F16 are multimodal functions, and F17-F22 are hybrid functions. F23-F30 are composition functions, which are selected from CEC2014. These 30 different benchmark functions can comprehensively estimate the performance of the ESCA_PSO. The related descriptions are demonstrated in Table 1, where range means the boundary of the search space for the relevant functions. As we all know, a unimodal function corresponds to a globally optimal solution; so, it can be employed to benchmark development capability. Conversely, the multimodal function possesses a lot of LO solutions, which leads to the algorithm falling into LO. Such functions can test the capability of the method to refrain from stagnation and exploration ability. Moreover, both the hybrid function and multimodal function only have one global optimum but multiple LO solutions. The structures of composition functions are more complex.

All the algorithms in the following experiments are coded on MATLAB 2014b. And to be fair, the experimental verification is carried out under the unified condition, i.e., the population size is set to 30, the maximum evaluation
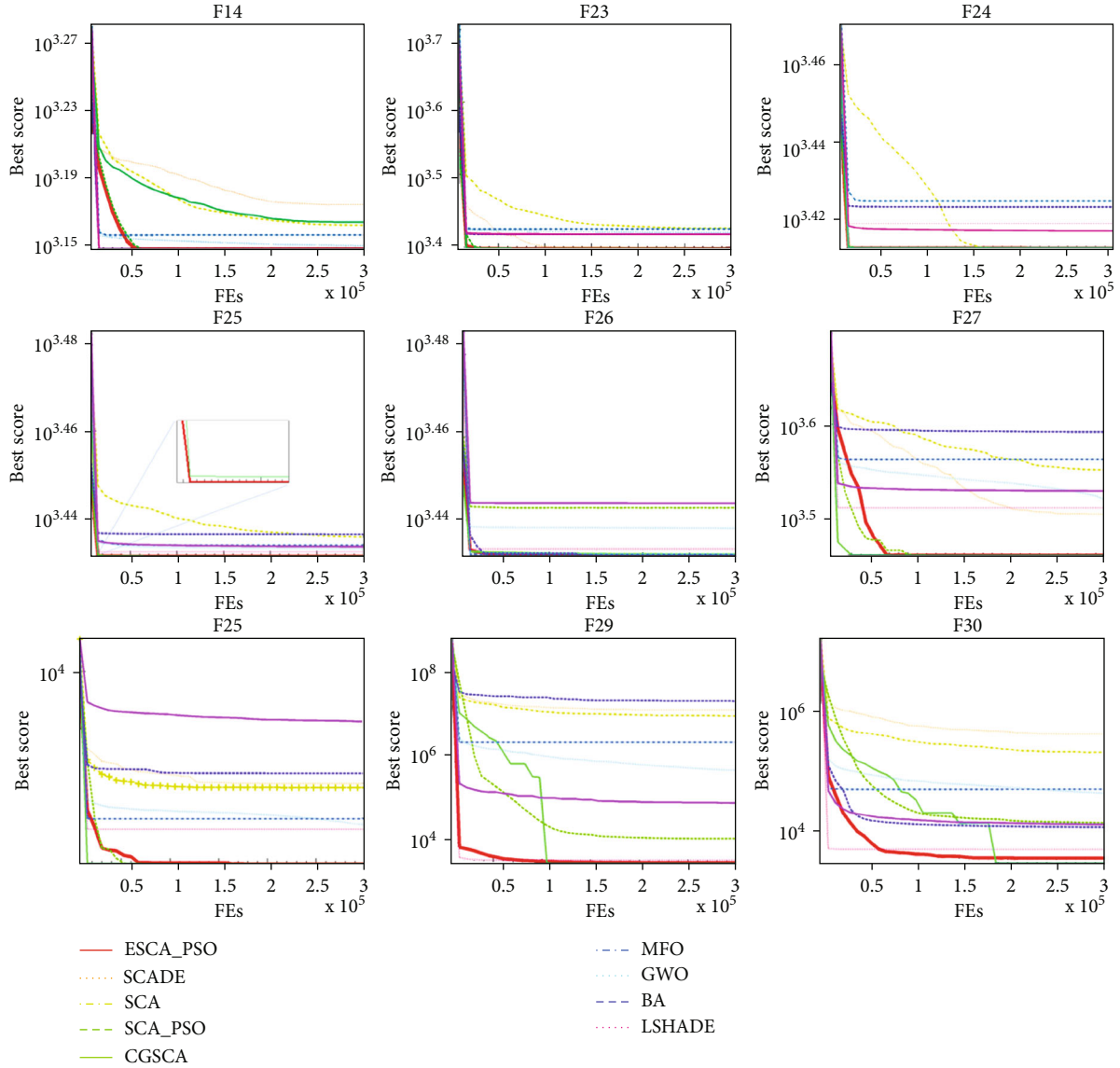
FIGURE 2: Convergence curves of ESCA_PSO and other algorithms 9 selected benchmark functions.

time is set to 300000, the dimension is set to 30, and the number of runs is set to 30.

4.2. Comparison with Other Algorithms. In this experiment, the ESCA_PSO was contrasted to SCA, GWO [51], MFO [52], BA [53], and PSO [54] on the functions presented in Table 1. To further validate the effect of the proposed ESCA_PSO, two improved SCA variants including SCA_PSO and SCADE [26] were involved for comparison and compare with LSHADE [55] which is the champion algorithm of CEC2014. The parameter configuration of algorithms is shown in Table 2. The detailed comparison results including the average value (Avg) of the best solution and standard deviation (Std) of every approach in 30 independent runs are displayed in Table 3.

We can see that the advantages of ESCA_PSO are not very obvious in the unimodal functions and multimodal functions. In these functions, ESCA_PSO is slightly better than the original algorithm SCA and its variants CGSCA and SCADE. But compared with high-quality algorithms such as LSHADE, there is still a certain gap. However, ESCA_PSO has a very good performance in the complex structure of the composition functions. Compared with other algorithms on F23-F30, it ranks first or second.

By the Friedman test, we can get the average ranking of test algorithms, which is usually used to get the difference between many test results. At the same time, to further analyze the experimental structure, Wilcoxon signed-rank test was adopted for statistical work.

In Table 4, all experimental results were taken from those two tests mentioned above. AVG in the table represents the average ranking of algorithms obtained by the Friedman test, and "+/-/=" represents the performance of the function compared with ESCA_PSO. Specifically, "+"

TABLE 5: Results of mutation mechanism comparison experiment.

	F1		F2		F3	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	1.0121E+07	4.1281E+06	7.2870E+07	6.7257E+06	9.0781E+03	1.6748E+03
ESCA_PSO1	1.0956E+07	6.3712E+06	7.5147E+07	7.5579E+06	9.4351E+03	2.5065E+03
ESCA_PSO2	8.8248E+06	4.3297E+06	7.5579E+07	6.4189E+06	8.9495E+03	2.0387E+03
ESCA_PSO3	8.9362E+06	3.2913E+06	7.6554E+07	5.5330E+06	8.2034E+03	2.8244E+03

	F4		F5		F6	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	4.7228E+02	4.7567E+01	5.2095E+02	7.3218E-02	6.2832E+02	3.6455E+00
ESCA_PSO1	4.8246E+02	4.3094E+01	5.2093E+02	5.9358E-02	6.2994E+02	3.6753E+00
ESCA_PSO2	4.8304E+02	3.1844E+01	5.2095E+02	4.0250E-02	6.2907E+02	2.8954E+00
ESCA_PSO3	4.6323E+02	3.5952E+01	5.2094E+02	6.0599E-02	6.2602E+02	3.6698E+00

	F7		F8		F9	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	7.0165E+02	6.6719E-02	9.7250E+02	1.7061E+01	1.1273E+03	2.1663E+01
ESCA_PSO1	7.0165E+02	5.8391E-02	9.7622E+02	2.0619E+01	1.1213E+03	2.3921E+01
ESCA_PSO2	7.0164E+02	4.8112E-02	9.7878E+02	1.5477E+01	1.1222E+03	2.4011E+01
ESCA_PSO3	7.0163E+02	6.2317E-02	9.5729E+02	1.6848E+01	1.1358E+03	3.5018E+01

	F10		F11		F12	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	4.9137E+03	7.2648E+02	5.6985E+03	5.0594E+02	1.2023E+03	3.2946E-01
ESCA_PSO1	4.9133E+03	6.5258E+02	5.6409E+03	7.1573E+02	1.2023E+03	2.7883E-01
ESCA_PSO2	4.9278E+03	6.4952E+02	5.6196E+03	5.2887E+02	1.2023E+03	2.8632E-01
ESCA_PSO3	3.9106E+03	7.0806E+02	5.7900E+03	6.4145E+02	1.2025E+03	1.9371E-01

	F13		F14		F15	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	1.3005E+03	8.0129E-02	1.4003E+03	5.5878E-02	1.5178E+03	1.8461E+00
ESCA_PSO1	1.3004E+03	6.5982E-02	1.4002E+03	3.9799E-02	1.5176E+03	1.4145E+00
ESCA_PSO2	1.3005E+03	1.3432E-01	1.4003E+03	5.0665E-02	1.5173E+03	1.5482E+00
ESCA_PSO3	1.3004E+03	9.2885E-02	1.4003E+03	1.0046E-01	1.5168E+03	1.2616E+00

	F16		F17		F18	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	1.6119E+03	3.7281E-01	3.5341E+05	1.5577E+05	1.9265E+06	5.4288E+05
ESCA_PSO1	1.6121E+03	2.7974E-01	4.7094E+05	5.3309E+05	1.9803E+06	5.1404E+05
ESCA_PSO2	1.6120E+03	3.9442E-01	4.2443E+05	3.0521E+05	2.1497E+06	4.0650E+05
ESCA_PSO3	1.6117E+03	3.5803E-01	4.4610E+05	3.0212E+05	1.7955E+06	4.8630E+05

	F19		F20		F21	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	1.9176E+03	2.6584E+00	4.6963E+03	2.1120E+03	1.5933E+05	1.2738E+05
ESCA_PSO1	1.9195E+03	1.1064E+01	4.8166E+03	2.4521E+03	1.6422E+05	1.1183E+05
ESCA_PSO2	1.9182E+03	2.6893E+00	4.2130E+03	1.8120E+03	1.7741E+05	1.5670E+05
ESCA_PSO3	1.9210E+03	1.6909E+01	2.5728E+03	3.9978E+02	1.2953E+05	9.6487E+04

TABLE 5: Continued.

	F22		F23		F24	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	3.0075E+03	2.5470E+02	2.5000E+03	0.0000E+00	2.6000E+03	0.0000E+00
ESCA_PSO1	3.0277E+03	1.8557E+02	2.5000E+03	0.0000E+00	2.6000E+03	0.0000E+00
ESCA_PSO2	2.9865E+03	2.5463E+02	2.5000E+03	0.0000E+00	2.6000E+03	0.0000E+00
ESCA_PSO3	2.8237E+03	2.0355E+02	2.5942E+03	4.7914E+01	2.6000E+03	3.1173E-04
	F25		F26		F27	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	2.7000E+03	0.0000E+00	2.7004E+03	7.7168E-02	2.9000E+03	0.0000E+00
ESCA_PSO1	2.7000E+03	0.0000E+00	2.7004E+03	9.2230E-02	2.9000E+03	0.0000E+00
ESCA_PSO2	2.7000E+03	0.0000E+00	2.7137E+03	3.4412E+01	2.9000E+03	0.0000E+00
ESCA_PSO3	2.7000E+03	0.0000E+00	2.7004E+03	7.1707E-02	3.4906E+03	3.4856E+02
	F28		F29		F30	
	Avg	Stdv	Avg	Stdv	Avg	Stdv
ESCA_PSO	3.0000E+03	0.0000E+00	3.1619E+03	5.6888E+01	3.7103E+03	6.0049E+02
ESCA_PSO1	3.0000E+03	0.0000E+00	3.4574E+03	4.0471E+02	5.0126E+03	1.8702E+03
ESCA_PSO2	3.0000E+03	0.0000E+00	7.4310E+04	2.6321E+05	1.1615E+04	8.8958E+03
ESCA_PSO3	3.9181E+03	8.8744E+02	3.1047E+03	4.4337E+00	3.2472E+03	4.9782E+01

means ESCA_PSO is better than this algorithm, "-" indicates that ESCA_PSO is inferior to this algorithm, and "=" means that the performance is similar to ESCA_PSO. In Wilcoxon signed-rank test, when the p value is less than 0.05, the performance between the two algorithms is significant. It was also used to evaluate the significance of ESCA_PSO versus other approaches. It can see from the table that ESCA ranks second on average, which is better than other algorithms overall. Compared with SCA, SCADE, CGSCA, GWO, and MFO, it is significantly better than 20 functions. However, it is indeed weaker than the champion algorithm LSHADE on multimodal functions and unimodal functions.

Figure 2 shows nine graphs of convergence we selected. As shown in Figure 2, it can be seen that ESCA_PSO does have a good convergence rate on these functions. It quickly converges to a lower point. And ESCA_PSO has a significant improvement over than original SCA. Of course, it is undeniable that some algorithms converge faster than ESCA_PSO, but ESCA_PSO has higher quality solutions.

Despite the great potential of the proposed ESCA_PSO, the approach of sacrificing a certain time complexity in exchange for an increase in terms of accuracy is insufficient side. Nevertheless, the algorithm is still competitive with LSHADE in unimodal and multimodal functions.

4.3. Comparison of Mutation Mechanism. As mentioned earlier, three mutation mechanisms were added to ESCA_PSO. To further analyze ESCA_PSO, we conducted comparative experiments on the mutation mechanism of ESCA in this section.

To compare the mutation mechanism, we construct three algorithms, namely, ESCA_PSO1, ESCA_PSO2, and ESCA_PSO3. Compared with ESCA_PSO, ESCA_PSO1 only uses Gaussian mutation while others remain unchanged. By analogy, ESCA_PSO2 only uses the Cauchy mutation while ESCA_PSO3 only uses the Lévy mutation. The population dimension and the total number of iterations of this experiment are the same as those in the previous experiment settings.

The results obtained from the experiments are shown in Table 5, and there is not much difference between these four algorithms, which can be concluded by comparing the whole data. This is because most of the four algorithms are the same, and only the mutation mechanism has changed. From the numerical value obtained from the experiment, ESCA_PSO has not achieved the best results in functions many times. But relatively, ESCA_PSO is rarely ranked last. This is also because ESCA_PSO integrates three mutation mechanisms, which makes it applicable to more functions.

Figure 3 shows several convergence graphs in this experiment. From the figure, we can see that in F2, F17, and F19, the convergence curves of the four algorithms are relatively similar, and there is no big difference in general. In F27 and F28, the performance of ESCA_PSO3 is not as good as the other three algorithms. In F29 and F30, ESCA_PSO2 is quite different from the other three algorithms. However, ESCA_PSO can keep a good level in these functions. This shows that the combination of three different mutation mechanisms can help the algorithm adapt to more functions.

However, ESCA_PSO that we proposed is not perfect, and there are certain limitations. In the benchmark functions experiment, it can be seen that there is still a gap between the performance of this algorithm and champion algorithms in unimodal functions and multimodal functions.
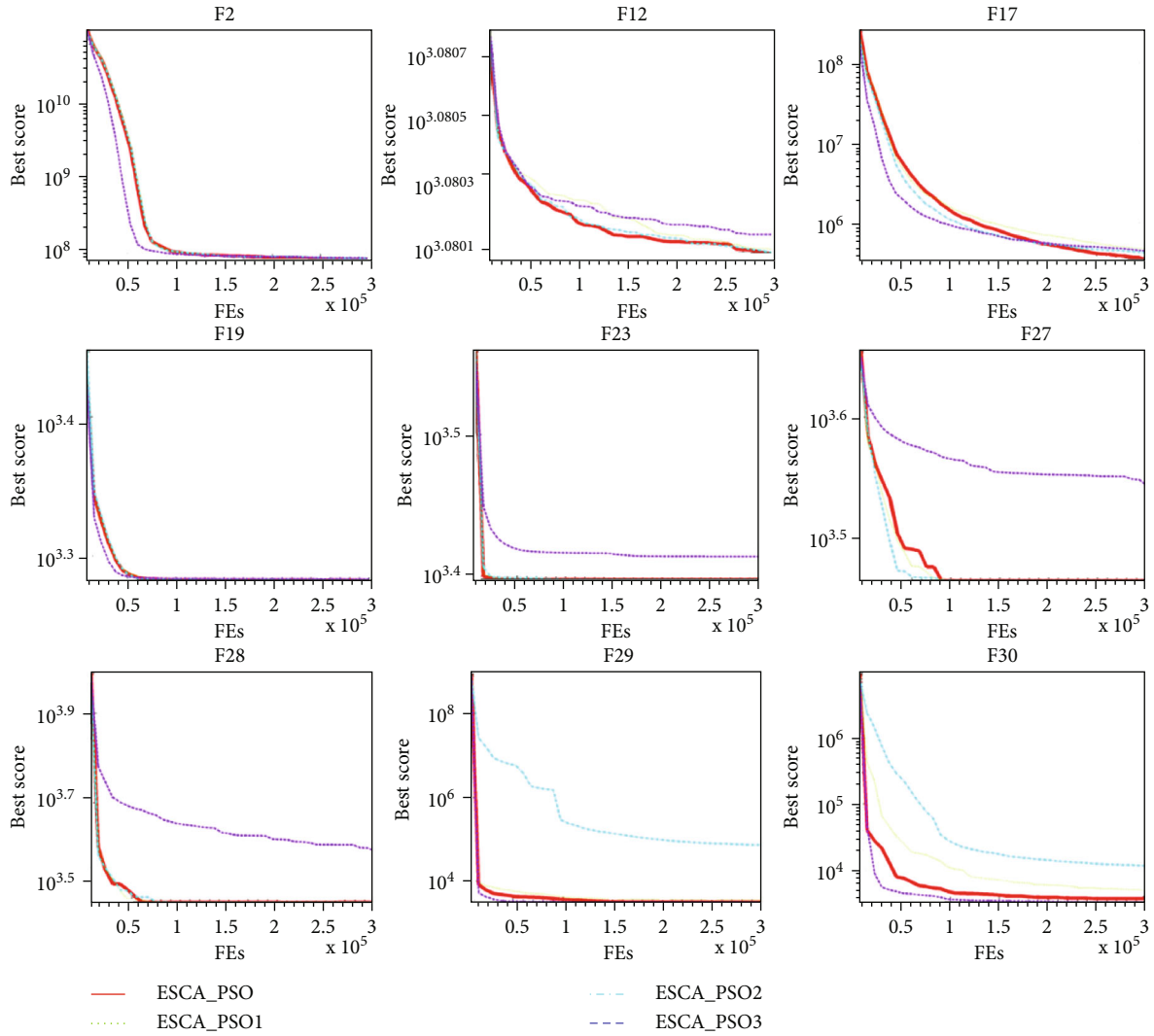
FIGURE 3: Convergence curves of ESCA_PSO and other algorithms 9 selected benchmark functions.

4.4. ESCA_PSO For Optimization of SVM.

Like many other machine learning methods [56], SVM has many advantages such as "simple structure," "overcoming dimension disaster," and "small sample," which can overcome the weaknesses of conventional neural networks such as poor learning and generalization ability [57]. Since its introduction, SVM has found its application in many practical problems. Practice shows that the penalty factor C and kernel function variable g have the key influence on the recognition accuracy of the SVM model when solving the recognition problem based on the radial basis kernel function. When the penalty factor C is small, the recognition rate of training and test samples is low, and the SVM is under learning. When C is too large, the accuracy of the training sample is higher, the test sample recognition rate is lower, and the SVM is overlearning. The smaller the kernel function parameter g is, the higher the training sample recognition rate is and the lower the accuracy of the test sample is. When g is larger, the accuracy of training and test samples becomes lower, and SVM is under learning. Traditional methods such

as trial and error method and network search method cannot meet the requirements of accuracy in practical application. Currently, with the development and maturity of MAs, good results have been achieved in improving the performance of the SVM model. For example, Li et al. [58] proposed moth-flame optimization (MFO) to tune the best parameters of SVM and applied it to the diagnosis of tuberculous pleural effusion. Li et al. [59] proposed a chaotic enhanced gravitational search algorithm (GSA) for optimizing the parameters of SVM. Das et al. [60] proposed to use the teaching-learning-based optimization (TLBO) for parameter optimization of SVM, and the good performance was validated by a financial case. Tang et al. [61] proposed a Lévy flight-based shuffled frog-leaping algorithm for determining the best parameters of SVM. Ahmadi et al. [62] developed the imperialist competition algorithm (ICA) to determine the best parameters of SVM for stock market timing. Li et al. [22] proposed SCA to tune the best parameters of SVM, and the good results were verified on several benchmark datasets. Rojas-Dominguez et al. [63] proposed to use
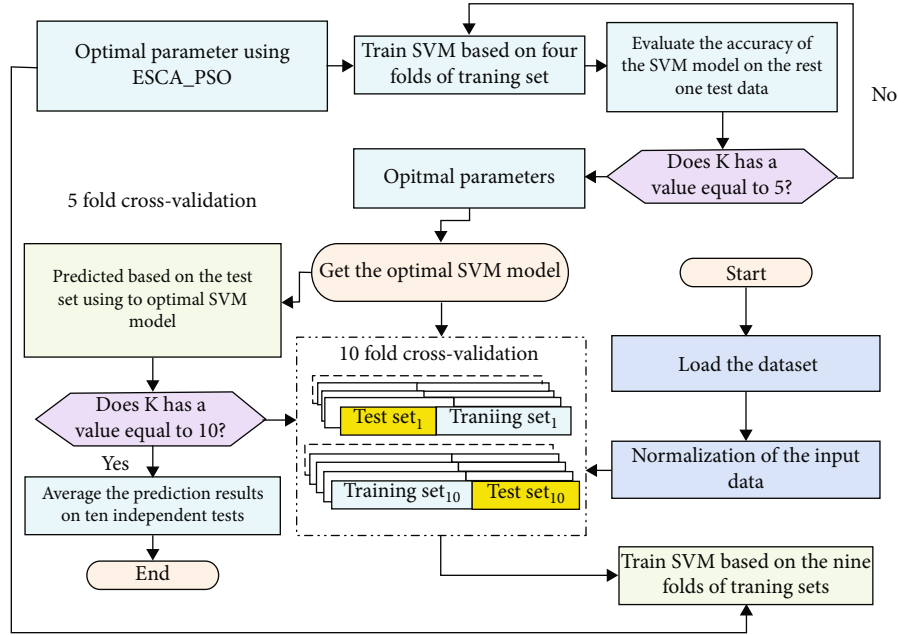
FIGURE 4: Flowchart of ESCA_PSO-SVM.

TABLE 6: Results obtained by ESCA_PSO-SVM on the Bupa liver problem.

Fold	ACC (%)	Sensitivity (%)	Specificity (%)	MCC
No. 1	74.29	61.54	81.82	0.4414
No. 2	67.65	36.36	82.61	0.2092
No. 3	80.00	71.43	85.71	0.5794
No. 4	82.35	90.91	78.26	0.6517
No. 5	82.35	64.29	95.00	0.6404
No. 6	71.43	54.55	79.17	0.3371
No. 7	64.71	55.56	75.00	0.3099
No. 8	62.86	38.89	88.24	0.3102
No. 9	77.14	65.00	93.33	0.5893
No. 10	67.65	53.33	78.95	0.3354
Avg.	73.04	59.18	83.81	0.4404
Max	82.35	90.91	95.00	0.6517
Min	62.86	36.36	75.00	0.2092

several metaheuristics to search for the best parameters of SVM, and the results showed that the estimation of distribution algorithms can achieve the best results. Tharwat et al. [64] proposed a chaotic antlion optimizer for tuning the best parameters, and the effectiveness was validated on an array of well-known datasets. Bablani et al. [65] proposed to use the bat algorithm (BA) to simultaneously determine the optimal parameters of SVM and the best subset of features and applied the model for dealing with the electroencephalography (EEG) data.

In this study, we applied ESCA_PSO to search for the best parameters of SVM, and the resultant model was called ESCA_PSO-SVM as shown in Figure 4. ESCA_PSO-SVM

was applied to predict two different medical problems including the Bupa liver and the Cleveland heart.

The Bupa liver diabetes dataset has a total of 345 samples and 7 features. Table 6 demonstrates the detailed results got by ESCA_PSO-SVM via 10-fold crossvalidation. As shown in Table 6, ESCA_PSO-SVM has got an average accuracy (ACC) of 73.04%, an average sensitivity of 59.18%, an average specificity of 83.81%, and an average Mathews correlation coefficient (MCC) of 0.4404.

From Figure 5, it is clear that ESCA_PSO-SVM has more excellent performance than SCA-SVM in such four indexes. Moreover, compared with the prediction accuracy, the ESCA_PSO-SVM has the best precision, while the KNN
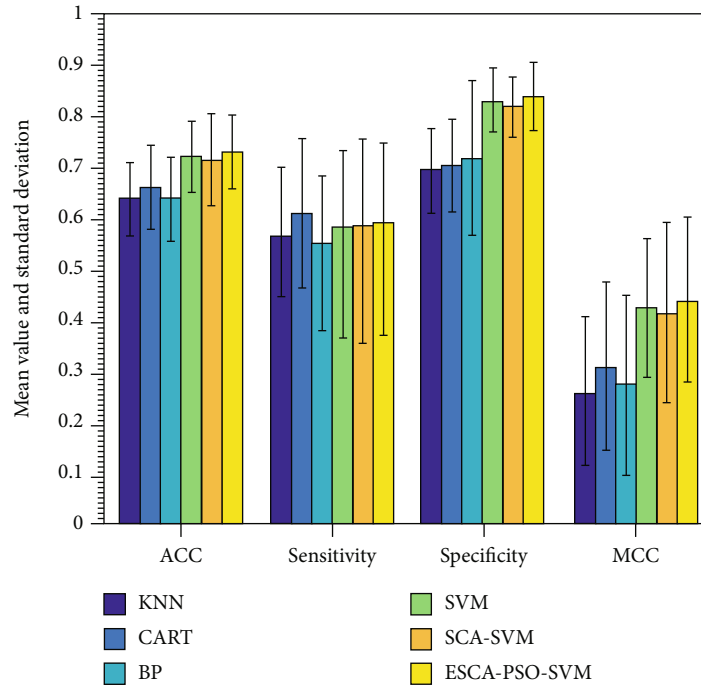
FIGURE 5: Comparison between the ESCA_PSO-SVM and other methods on the Bupa liver problem.

TABLE 7: Results obtained by ESCA_PSO-SVM on the Cleveland heart problem.

Fold	ACC (%)	Sensitivity (%)	Specificity (%)	MCC
No. 1	76.67	72.22	83.33	0.5443
No. 2	87.10	78.57	94.12	0.7427
No. 3	83.33	76.92	88.24	0.6591
No. 4	77.42	76.92	77.78	0.5424
No. 5	80.00	69.23	88.24	0.5909
No. 6	76.67	78.57	75.00	0.5345
No. 7	83.33	87.50	78.57	0.6652
No. 8	83.33	50.00	100.00	0.6325
No. 9	93.55	90.00	95.24	0.8524
No. 10	86.67	88.89	83.33	0.7222
Avg.	82.81	76.88	86.38	0.6486
Max	93.55	90.00	100.00	0.8524
Min	76.67	50.00	75.00	0.5345

model has the lowest precision. Based on the sensitivity metric, ESCA_PSO-SVM ranked second, while BP obtained the worst values. In terms of the obtained specificity, ESCA_PSO-SVM was the best, followed by SVM, SCA-SVM, BP, KNN, and CART. In terms of the MCC, ESCA_PSO-SVM provided the best value, followed by SVM, SCA-SVM, CART, BP, and KNN. It suggests that ESCA_PSO-SVM is more advantageous and stable in solving the Bupa liver problem.

The Cleveland heart data was got from the UCI repository, and it includes 303 samples and 76 features. Table 7 shows the detailed results of ESCA_PSO-SVM through 10-fold crossvalidation on this dataset. From Table 7, ESCA_PSO-SVM has got an average ACC of 82.81%, a sensitivity of 76.88%, a specificity of 86.38%, and an MCC of 0.6486.

In Figure 6, ESCA_PSO-SVM is superior to SCA-SVM in terms of four evaluation indexes. Concerning the classification accuracy, it can be seen clearly that the ESCA_PSO-SVM has got the best ACC, whereas BP has the lowest precision. In terms of the sensitivity metric, the value of SVM is the same as that of ESCA_PSO-SVM which takes the first place. As for the specificity metric, although SCA-SVM ranked first place, it was only slightly superior to ESCA_
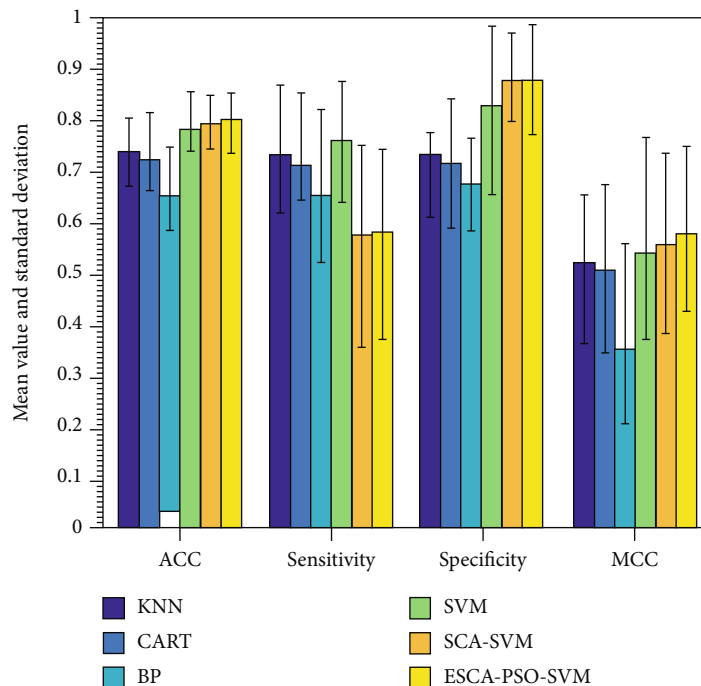
FIGURE 6: Comparison between the ESCA_PSO-SVM and other methods on the Cleveland heart problem.

PSO-SVM. According to the MCC metric, ESCA_PSO provided the best value, followed successively by SCA-SVM, SVM, KNN, CART, and BP. These all proved the robustness and stableness of the ESCA_PSO-SVM on the Cleveland heart problem. Shortly, many problems are waiting to be optimized for which ESCA_PSO can be applied, such as disease module identification [66], molecular signatures identification for cancer diagnosis [67], drug-disease associations prediction [68], drug discovery [69], and pharmacoinformatic data mining [70].

5. Conclusions and Future Directions

To make up for the deficiency of SCA_PSO, this paper proposed ESCA_PSO, an enhanced version of SCA_PSO. The chance of prematurely falling into convergence was effectively reduced by introducing DE and joint mutation mechanisms. To verify its performance, it was compared with seven advanced algorithms on 30 benchmark function sets. The experimental results showed that the performance of the proposed algorithm was better than that of the traditional optimization algorithms and had certain competitiveness with LSHADE. Inspired by the "No Free Lunch" theory, this paper further explored the application of ESCA_PSO in medical diagnosis and successfully applied it to hyperparameter optimization of support vector machine.

The results showed that the support vector machine model combined with the proposed algorithm outperformed the other five existing models and achieved an average accuracy of 82.81%. In conclusion, the proposed algorithm can be regarded as a reliable technique for solving practical problems.

For future work, there are still many problems worthy of study. First of all, we will continue to improve the algorithm, by means such as trying to introduce other metaheuristic algorithms or optimizing the time complexity while ensuring the effect. In addition, we will try to apply ESCA_PSO to other fields such as image segmentation, clustering optimization, and discrete optimization.

Data Availability

The data involved in this study are all public data, which can be downloaded through public channels.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] C. A. Floudas and C. E. Gounaris, "A review of recent advances in global optimization," *Journal of Global Optimization*, vol. 45, no. 1, pp. 3–38, 2009.

[2] M. Wang, H. Chen, B. Yang et al., "Toward an optimal kernel extreme learning machine using a chaotic moth-flame

optimization strategy with applications in medical diagnoses," *Neurocomputing*, vol. 267, pp. 69–84, 2017.

[3] M. Chen, G. Zeng, K. Lu, and J. Weng, "A two-layer nonlinear combination method for short-term wind speed prediction based on ELM, ENN, and LSTM," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6997–7010, 2019.

[4] H. Zhou, J. Pang, P.-K. Chen, and F.-D. Chou, "A modified particle swarm optimization algorithm for a batch-processing machine scheduling problem with arbitrary release times and non-identical job sizes," *Computers & Industrial Engineering*, vol. 123, pp. 67–81, 2018.

[5] L. Shen, H. Chen, Z. Yu et al., "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[6] G.-Q. Zeng, X.-Q. Xie, M.-R. Chen, and J. Weng, "Adaptive population extremal optimization-based PID neural network for multivariable nonlinear control systems," *Swarm and Evolutionary Computation*, vol. 44, pp. 320–334, 2019.

[7] W. Deng, J. Xu, Y. Song, and H. Zhao, "An effective improved co-evolution ant colony optimisation algorithm with multi-strategies and its application," *International Journal of Bio-Inspired Computation*, vol. 16, no. 3, pp. 158–170, 2020.

[8] H. Zhao, H. Liu, J. Xu, and W. Deng, "Performance prediction using high-order differential mathematical morphology gradient spectrum entropy and extreme learning machine," *IEEE transactions on instrumentation and measurement*, vol. 69, no. 7, pp. 4165–4172, 2020.

[9] Z. J. Wang, Z. H. Zhan, W. J. Yu et al., "Dynamic group learning distributed particle swarm optimization for large-scale optimization and its application in cloud workflow scheduling," *IEEE transactions on cybernetics*, vol. 50, pp. 2715–2729, 2019.

[10] D. Liang, Z.-H. Zhan, Y. Zhang, and J. Zhang, "An efficient ant colony system approach for new energy vehicle dispatch problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, pp. 4784–4797, 2019.

[11] F. Zhao, S. Di, J. Cao, and J. Tang, "A novel cooperative multi-stage hyper-heuristic for combination optimization problems," *Complex System Modeling and Simulation*, vol. 1, no. 2, pp. 91–108, 2021.

[12] X. Lai and Y. Zhou, "Analysis of multiobjective evolutionary algorithms on the biobjective traveling salesman problem (1, 2)," *Multimedia Tools and Applications*, vol. 79, no. 41-42, pp. 30839–30860, 2020.

[13] Z. Liang, J. Zhang, L. Feng, and Z. Zhu, "A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking," *Expert Systems with Applications*, vol. 138, p. 112798, 2019.

[14] Z. Yang, K. Li, Y. Guo, H. Ma, and M. Zheng, "Compact real-valued teaching-learning based optimization with the applications to neural network training," *Knowledge-Based Systems*, vol. 159, pp. 51–62, 2018.

[15] J. Ye and W. Cui, "Modeling and stability analysis methods of neutrosophic transfer functions," *Soft Computing*, vol. 24, no. 12, pp. 9039–9048, 2020.

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

[17] H. Nenavath, R. K. Jatoth, and S. Das, "A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking," *Swarm and Evolutionary Computation*, vol. 43, pp. 1–30, 2018.

[18] 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.

[19] S. Das, A. Bhattacharya, and A. K. Chakraborty, "Solution of short-term hydrothermal scheduling using sine cosine algorithm," *Soft Computing*, vol. 22, no. 19, pp. 6409–6427, 2018.

[20] M. A. Tawhid and V. Savsani, "Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems," *Neural Computing and Applications*, vol. 31, no. S2, pp. 915–929, 2019.

[21] J. Wang, W. Yang, P. Du, and T. Niu, "A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm," *Energy Conversion and Management*, vol. 163, pp. 134–150, 2018.

[22] S. Li, H. Fang, and X. Liu, "Parameter optimization of support vector regression based on sine cosine algorithm," *Expert Systems with Applications*, vol. 91, pp. 63–77, 2018.

[23] A. Lin, Q. Wu, A. A. Heidari et al., "Predicting intentions of students for master programs using a chaos-induced sine cosine-based fuzzy K-nearest neighbor classifier," *IEEE Access*, vol. 7, pp. 67235–67248, 2019.

[24] J. Tu, A. Lin, H. Chen, Y. Li, and C. Li, "Predict the entrepreneurial intention of fresh graduate students based on an adaptive support vector machine framework," *Mathematical Problems in Engineering*, vol. 2019, 16 pages, 2019.

[25] M. Issa, A. E. Hassanien, D. Oliva, A. Helmi, I. Ziedan, and A. Alzohairy, "ASCA-PSO: adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment," *Expert Systems with Applications*, vol. 99, pp. 56–70, 2018.

[26] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Applied Soft Computing Journal*, vol. 62, pp. 1019–1043, 2018.

[27] M. Abd Elaziz, D. Oliva, and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Systems with Applications*, vol. 90, pp. 484–500, 2017.

[28] C. Qu, Z. Zeng, J. Dai, Z. Yi, and W. He, "A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 4231647, 2018.

[29] N. Kumar, I. Hussain, B. Singh, and B. K. Panigrahi, "Single sensor-based MPPT of partially shaded PV system for battery charging by using cauchy and gaussian sine cosine optimization," *IEEE Transactions on Energy Conversion*, vol. 32, no. 3, pp. 983–992, 2017.

[30] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert Systems with Applications*, vol. 123, pp. 108–126, 2019.

[31] O. E. Turgut, "Thermal and economical optimization of a shell and tube evaporator using hybrid backtracking search-sine-cosine algorithm," *Arabian Journal for Science and Engineering*, vol. 42, no. 5, pp. 2105–2123, 2017.

[32] W. Guo, Y. Wang, F. Zhao, and F. Dai, "Riesz fractional derivative Elite-guided sine cosine algorithm," *Applied Soft Computing*, vol. 81, p. 105481, 2019.

[33] S. Gupta and K. Deep, "A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons," *Applied Intelligence*, vol. 50, no. 4, pp. 993–1026, 2020.

[34] S. Gupta and K. Deep, "A hybrid self-adaptive sine cosine algorithm with opposition based learning," *Expert Systems with Applications*, vol. 119, pp. 210–230, 2019.

[35] R. Wang, Z. Wu, J. Lou, and Y. Jiang, "Attention-based dynamic user modeling and deep collaborative filtering recommendation," *Expert Systems with Applications*, vol. 188, article 116036, 2022.

[36] B. Bai, G. Li, S. Wang, Z. Wu, and W. Yan, "Time series classification based on multi-feature dictionary representation and ensemble learning," *Expert Systems with Applications*, vol. 169, article 114162, 2021.

[37] W. Yan, G. Li, Z. Wu, S. Wang, and P. S. Yu, "Extracting diverse-shapelets for early classification on time series," *World Wide Web*, vol. 23, no. 6, pp. 3055–3081, 2020.

[38] Z. Wu, R. Wang, Q. Li et al., "A location privacy-preserving system based on query range cover-up or location-based services," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5244–5254, 2020.

[39] Z. Wu, S. Shen, X. Lian, X. Su, and E. Chen, "A dummy-based user privacy protection approach for text information retrieval," *Knowledge-Based Systems*, vol. 195, p. 105679, 2020.

[40] Z. Mei, H. Zhu, Z. Cui et al., "Executing multi-dimensional range query efficiently and flexibly over outsourced ciphertexts in the cloud," *Information Sciences*, vol. 432, pp. 79–96, 2018.

[41] Z. Wu, H. Zhu, G. Li et al., "An efficient Wikipedia semantic matching approach to text document classification," *Information Sciences*, vol. 393, pp. 15–28, 2017.

[42] Z. Wu, L. Lei, G. Li et al., "A topic modeling based approach to novel document automatic summarization," *Expert Systems with Applications*, vol. 84, pp. 12–23, 2017.

[43] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, article 114864, 2021.

[44] I. Ahmadianfar, A. Asghar Heidari, S. Noshadian, H. Chen, and A. H. Gandomi, "INFO: an efficient optimization algorithm based on weighted mean of vectors," *Expert Systems with Applications*, vol. 195, article 116516, 2022.

[45] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems-the International Journal of Escience*, vol. 97, pp. 849–872, 2019.

[46] 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.

[47] I. Ahmadianfar, A. Asghar Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN Beyond the Metaphor: An Efficient Optimization Algorithm Based on Runge Kutta Method," *Expert Systems with Applications*, vol. 181, article 115079, 2021.

[48] K. Daoden and T. Thaiupthump, "A modified shuffled frog leaping algorithm using truncated Gaussian distribution in frog's Position updating process," in *Information Science and Applications (ICISA) 2016*, pp. 965–974, Singapore, 2016.

[49] W. Hui, L. Hui, L. Yong, L. Changhe, and Z. Sanyou, "Opposition-based particle swarm algorithm with cauchy mutation," in *2007 IEEE Congress on Evolutionary Computation*, pp. 4750–4756, Singapore, 2007.

[50] M. Jamil and H.-J. Zepernick, "3 - Lévy flights and global optimization," in *Swarm Intelligence and Bio-Inspired Computation*, X.-S. Yang, Z. Cui, R. Xiao, A. H. Gandomi, and M. Karamanoglu, Eds., pp. 49–72, Elsevier, Oxford, 2013.

[51] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[52] S. Mirjalili, "Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm," *Knowledge-Based Systems*, vol. 89, pp. 228–249, 2015.

[53] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, Berlin Heidelberg, 2010.

[54] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds., pp. 760–766, Springer US, Boston, MA, 2010.

[55] R. Tanabe and A. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE congress on evolutionary computation (CEC)*, Beijing, China, 2014.

[56] Y. Li, W.-G. Cui, H. Huang, Y.-Z. Guo, K. Li, and T. Tan, "Epileptic seizure detection in EEG signals using sparse multiscale radial basis function networks and the fisher vector approach," *Knowledge-Based Systems*, vol. 164, pp. 96–106, 2019.

[57] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[58] C. Li, L. Hou, B. Sharma et al., "Developing a new intelligent system for the diagnosis of tuberculous pleural effusion," *Computer Methods & Programs in Biomedicine*, vol. 153, pp. 211–225, 2018.

[59] C. Li, X. An, and R. Li, "A chaos embedded GSA-SVM hybrid system for classification," *Neural Computing & Applications*, vol. 26, no. 3, pp. 713–721, 2015.

[60] S. P. Das, N. S. Achary, and S. Padhy, "Novel hybrid SVM-TLBO forecasting model incorporating dimensionality reduction techniques," *Applied Intelligence*, vol. 45, no. 4, pp. 1148–1165, 2016.

[61] D. Tang, J. Yang, S. Dong, and Z. Liu, "A lévy flight-based shuffled frog-leaping algorithm and its applications for continuous optimization problems," *Applied Soft Computing*, vol. 49, pp. 641–662, 2016.

[62] E. Ahmadi, M. Jasemi, L. Monplaisir, M. A. Nabavi, A. Mahmoodi, and P. A. Jam, "New efficient hybrid candlestick technical analysis model for stock market timing on the basis of the support vector machine and heuristic algorithms of imperialist competition and genetic," *Expert Systems with Applications*, vol. 94, pp. 21–31, 2018.

[63] A. Rojas-Dominguez, L. Carlos Padierna, J. M. Carpio Valadez, H. J. Puga-Soberanes, and H. J. Fraire, "Optimal hyperparameter tuning of SVM classifiers with application to medical diagnosis," *IEEE Access*, vol. 6, pp. 7164–7176, 2018.

[64] A. Tharwat and A. E. Hassanien, "Chaotic antlion algorithm for parameter optimization of support vector machine," *Applied Intelligence*, vol. 48, no. 3, pp. 670–686, 2018.

[65] A. Bablani, D. R. Edla, D. Tripathi, S. Dodia, and S. Chintala, "A synergistic concealed information test with novel approach for EEG channel selection and SVM parameter optimization," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 11, pp. 3057–3068, 2019.

[66] Y. Su, C. Liu, Y. Niu, F. Cheng, and X. Zhang, "A community structure enhancement-based community detection algorithm for complex networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, pp. 2833–2846, 2019.

[67] Y. Su, S. Li, C. Zheng, and X. Zhang, "A heuristic algorithm for identifying molecular signatures in cancer," *IEEE Transactions on Nanobioscience*, vol. 19, no. 1, pp. 132–141, 2020.

[68] L. Cai, C. Lu, J. Xu et al., "Drug repositioning based on the heterogeneous information fusion graph convolutional network," *Briefings in Bioinformatics*, vol. 22, no. 6, article bbab319, 2021.

[69] Y. Li, X. Li, J. Hong et al., "Clinical trials, progression-speed differentiating features and swiftness rule of the innovative targets of first-in-class drugs," *Briefings in Bioinformatics*, vol. 21, no. 2, pp. 649–662, 2020.

[70] J. Yin, W. Sun, F. Li et al., "VARIDT 1.0: variability of drug transporter database," *Nucleic Acids Research*, vol. 48, no. D1, pp. D1042–D1050, 2020.