Support Vector Machine Optimized by Henry Gas Solubility Optimization Algorithm and Archimedes Optimization Algorithm to Solve Data Classification Problems

Ji-Sheng Yu, Sheng-Kai Zhang *, Jie-Sheng Wang, Song Li, Ji Sun, Rui Wang

Abstract—**Support vector machine (SVM) is the minimization of structural risk to construct a better hyperplane to maximize the distance between the hyperplane and the sample points on both sides of hyperplane. Two improved physics-wise swarm intelligence optimization algorithms (Henry gas solubility optimization algorithm and Archimedes optimization algorithm) were proposed based on Lévy flight operator, Brownian motion operator and Tangent flight motion operator to optimize the penalty factor and kernel function parameters of SVM so as to enhance its global and local search ability. Finally, the Iris datasets, Strip surface defect datasets, Wine datasets and Wisconsin datasets of breast cancer in UCI datasets were selected to carry out the simulation experiment. Simulation results show that optimizing SVM based on improved physical-wise swarm intelligence algorithms can effectively improve the classification accuracy.**

Index Terms—**SVM, data classification, henry gas solubility optimization algorithm, archimedes optimization algorithm**

I. Introduction

OPTIMIZATION is to find the optimal solution in the shortest time and in a specific memory space based on mathematical knowledge [1]. Practical problems in many areas need to be optimized, such as system adaptive control, artificial intelligence robots, pattern recognition and deep learning. Swarm intelligent algorithms are the most popular and the most extensive class of algorithms, most of which are inspired from animals or human behaviors and follow the principle of survival of the fittest, and intelligent searching modes. A kind of swarm intelligence optimization algorithms based on physical phenomena were discussed. The lightning search algorithm was proposed based on lightning phenomena [2], a natural physical phenomenon and the mechanism of step-wise pilot transmission. The gravitational search algorithm was inspired by Newton's law of gravity [3]. The particle is abstracted as feasible solutions, and its mass is regarded as the fitness value. The gravitation between individuals leads to the motion of the feasible solution towards the optimal solution. The wind driving algorithm was proposed according to Newton's second law [4]. It uses the air particle as individuals and it is gradually optimized with its final flow position as a feasible solution. The atom search algorithm was proposed in 2019, and its research background comes from molecular dynamics models. Global optimal atoms and other atoms generate gravitational repulsion to each other to efficiently search for the optimal solution [5]. Its idea is similar to the newly proposed charged particle algorithm [6], where not all particles interact, but some particles are select to interact. This paper focuses on the Henry gas solubility optimization algorithm (HGSO) and the Archimedes optimization algorithm (AOA) .

Henry gas solubility optimization (HGSO) is a new swarm intelligence algorithm based on physical background [7]. HGSO algorithm is based on Henry's law, which shows the process of the solubility of gas in liquid. The solubility is inseparable from the change of temperature. Domestic scholar Xie et al. proposed a hybrid Harris Hawks Henry Gas Solubility Optimization algorithm (HHO-HGSO) based on the basic HGSO algorithm [8]. It takes the Harris hawks predation strategy and switches from global search to local search based on energy judging. The experimental results show that HHO-HGSO algorithm greatly improves the search accuracy than HGSO algorithm. Elaziz et al. proposed an improved HGSO algorithm (HGSWC) based on whale optimization algorithm (WOA) and all-round opposite learning [9]. HGSWC performs better than all other meta-heuristic algorithms on 14 different task scheduling problems. A novel chaos HGSO algorithm (CHGSO) was proposed by Betul [10]. Its main strategy is to use 10 common chaotic maps to change the position update mode of HGSO algorithm. A quantum HGSO algorithm (QHGSO) was proposed by taking the quantum theory instead of the

Manuscript received December 10, 2022; revised March 9, 2023. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293 and LJKZ1199), and the 2021 Enterprise Doctor Dual initiative of YingKou Science and Technology Bureau (Grant No. QB-2021-02 and QB-2021-02).

Ji-Sheng Yu is an associate professor of School of Electrical Engineering, Yingkou Institute of Technology, Yingkou, 115014, P. R. China (e-mail: 121388969@qq.com).

Sheng-Kai Zhang is an associate professor of School of Electrical Engineering, YingKou Institute of Technology, Yingkou, 115014, P. R. China (Corresponding author, phone: 86-0417-3555556; e-mail: 94060765@qq.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: wang_jiesheng@126.com).

Song Li is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 812738843@qq.com).

Ji Sun is a senior engineer of School of Electrical Engineering, Yingkou Institute of Technology, Yingkou, 115014, P. R. China (e-mail: 5122585@qq.com).

Rui-Wang is an engineer of Yingkou Northern Steel Drum Manufacturing Equipment Technology Co., Ltd, Liaoning, Yingkou, 115014, P. R. China (e-mail: 447652090@qq.com)

original search method in HGSO algorithm. On the other hand, the chaos coefficient was introduced and the Monte-Carlo method is used when the gas particles update their positions [11].

The Archimedes optimization algorithm (AOA) was proposed by Fatma A. Hashim [12], which comes directly from Archimedes' laws of physics. It illustrates the relationship between an object immersed in a fluid and the buoyancy it receives. Iranian scholar Hayati proposed an improved Archimedes optimization algorithm (IAOA) based on absolute error [13]. The chaotic mapping was used to change three random numbers in AOA and Lévy flight operator was introduced to change the original position update strategy. Numerous test results show that IAOA can greatly reduce the cost of building fuel cells. Houssein et al. proposed an improved Archimedes optimization algorithm based on local escape operator (LEO) and orthogonal learning (OL) [14]. LEO operator improves solution accuracy through special position update criteria and OL operator is used to expand the range of optimal solutions in the searching space. A determination of the best polymer electrolyte parameters by IAOA shows it caned effectively enhances the local and global search capability.

In this paper, SVM is optimized by the improved physical-wise swarm intelligence algorithms to solve the data classification problem, and the effectiveness of the proposed algorithm is tested through four different scale data sets.

II. Support Vector Machine (SVM)

Support vector machine (SVM) is an algorithm to classify data. Its basic idea is to achieve minimum risk and classification interval as large as possible [15].

A. Linear Support Vector Machine

When using SVM to solve the classification, a reasonable training set $\Omega = \{(x_i,\ y_i), \mathrm{i} = 1, ..., n\}$ ($x_i \epsilon R$ and $y_i \epsilon R$) should be choose. Then, the classification model $f(x)$ is trained through the selected training set. Comparing the results obtained from the testing set with the predicted data y, the more the same number, the higher the classification accuracy. The regression function of SVM in the high-dimensional space is shown in Eq. (1).

$$f(x) = w * x + b \tag{1}$$

According to Eq. (2)-(3) and the basic principles of SVM, it can be transformed into basic optimization problem.

$$\begin{aligned} minJ &= \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ y_i - w * x_1 - b &\le \varepsilon + \xi_i \\ w * x_i + b - y_i &\le \varepsilon + \xi_i^* \end{aligned} \tag{2}$$

$$\xi_i,\ \xi_i^* \ge 0 \tag{3}$$

where, ε is insensitive loss coefficient, ξ_i and ξ_i^* are the relaxation variables, and C is the penalty parameter. Then the dual variables and Lagrangian function are added to obtain:

$$\begin{aligned} w(\alpha,\ \alpha^*) = &-\frac{1}{2}\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(x_i * x_j) \\ &+ \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\, y_i - \sum_{i=1}^{l}(\alpha_i + \alpha_i^*)\,\varepsilon \end{aligned} \tag{4}$$

The linear regression function can be described as:

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)(x_i * x) + b \tag{5}$$

where, α_i^* is the Lagrangian mathematical operator and b is used to ensure that the interval is larger and the error rate is as small as possible.

B. Non-linear Support Vector Machine

In order to solve some special nonlinear classification problems, the idea of nonlinear transformation is usually adopted. The dual variables must also be adopted in such cases, and the inner product among samples is determined by the objective and classification decision function. The kernel function needs to be used as the inner product. $L(x, z)$ is a class of kernel functions and $\varphi(x)$ represents this map. Then a nonlinear SVM is shown in Eq. (6).

$$f(x) = sign\left[\sum_{i=1}^{N}\alpha_i^*\, y_i L(x, x_i) + b^*\right] \tag{6}$$

Then the kernel function $K(x, z)$ and the penalty parameters were selected and the classification decision function is shown in Eq. (7).

$$\begin{aligned} f(x) &= sign(w^{*T}\varphi(x) + b^*) \\ &= sign\left\{\sum_{i=1}^{N} y_i\,\alpha_i^*\varphi^T(x_i)\varphi(x) + b^*\right\} \\ &= sign\left\{\sum_{i=1}^{N} y_i\,\alpha_i^* L(x_i, x_j) + b^*\right\} \end{aligned} \tag{7}$$

C. Kernel Functions

When realizing the classification with SVM, different kernel functions can be adopted. There are 4 different types of kernel functions.

(1) Linear kernel function.

$$K(x_i, x_j) = x_i^T x_j \tag{8}$$

(2) Polynomial kernel function. d is the order of polynomial, and the larger the value, the more complex the operation.

$$K(x_i, x_j) = (x_i^T x_j + 1)^d \tag{9}$$

(3) Sigmoid kernel function.

$$K(x_i, x_j) = tanh(\beta_0 x_i^T x_j + \beta) \tag{10}$$

(4) Gaussian radial basis kernel function.

$$\begin{aligned} K(x_i, x_j) &= exp\left(-\frac{1}{\sigma^2}\|x_i - x_j\|^2\right) \\ &= exp\left(-g\|x_i - x_j\|^2\right) \end{aligned} \tag{11}$$

Reasonable selection of kernel functions will greatly affect the performance of the SVM classification model. The radial

basis kernel function is most widely used, which is characterized by high accuracy and small computation. But it has two important parameters, penalization parameter C and the kernel parameter γ. These two parameters determine the accuracy of the optimal hyperplane classification. The parameter C is essentially the regularization coefficient. It represents the tolerance value of the SVM for the error. The ultimate goal of C is to balance data accuracy with data complexity and balance both empirical and structural risks. To determine these two important parameters, the physical phenomenon-based swarm intelligence optimization algorithms are used to determine these parameters [16].

III. Swarm Intelligence Optimization Algorithm Based on Physical Phenomena

A. Henry Gas Solubility Optimization Algorithm

Henry gas solubility optimization (HGSO) is a physical phenomenon heuristic algorithm according to the dissolution law of gas in liquid [7]. It successfully simulates the process of the solubility of gas in liquid changing with temperature. Henry's law states that at a constant temperature, the amount of a given gas dissolved in a liquid of a given type and volume is proportional to the partial pressure of the gas in equilibrium with the liquid. The relationship between these physical quantities can be represented by:

$$S_g = H \times P_g \tag{12}$$

where, S_g indicates the solubility of the gas and P_g indicates the partial pressure of the gas. H indicates the Henry constant, which is a specific constant for a given gas and solvent combination at a given temperature. The relationship between H and the current temperature T^θ can be expressed as:

$$H(T) = H^\theta \times exp\left(\frac{-\nabla_{sol}E}{R}\left(\frac{1}{T} - 1/T^\theta\right)\right) \tag{13}$$

where, $\nabla_{sol}E$ indicates the Enthalpy of dissolution; R indicates the gas constant; T^θ is the reference temperature, $T^\theta = 289.15K$; T indicates the current temperature; H^θ is the reference Henry constant; $H(T)$ is a Henry constant varying with temperature. When $\nabla_{sol}E$ is a constant, the Van' t Hoff equation is valid, so Eq. (13) can be further written as:

$$H(T) = H^\theta \times \exp\left(-C \times \frac{1}{T} - 1/T^\theta\right) \tag{14}$$

Let the population size be N and the number of clusters be n. All particles are assigned randomly to a certain cluster and the number of particles in each cluster is N/n. The optimal particle X_{best} in the population and the optimal particle X_{best} in each cluster were found based on their fitness values. Similar to the parameters in Eq. (12)-(13), some parameters related to Henry's law are defined in HGSO and their initial values are initialized by Eq. (15).

$$H_j(t) = l_1 \times rand(0,1), P_{i,j=}l_2 \times rand(0,1), C_j = l_3 \times rand(0,1), (i = 1,2, \ldots N)(j = 1,2, \ldots, n) \tag{15}$$

where, l_1, l_2, l_3 are constants. The updated formula for the Henry constant in each cluster can be described as:

$$H_j(t+1) = H_j(t) \times \exp\left(-C_j \times \frac{1}{T_{(t)}} - 1/T^\theta\right) \tag{16}$$

$$T(t) = \exp\left(-t/iter\right)$$

where, $T^\theta = 289.15$, t and $iter$ are the current iteration number and the maximum iteration number respectively. Based on Eq. (15)-(16), the current solubility of each particle X_i can be expressed as:

$$S_i(t) = K \times H_j(t+1) \times P_{i,j}(t) \tag{17}$$

where, K and $P_{i,j}(t)$ are constants. According to the expression of $H_j(t+1)$, $S_i(t)$ continuously decreases with the number of iterations, which is a process in which the solubility of the gas particles changes with time. The position update for the particle X_i is realized by:

$$\begin{aligned} X_i(t+1) = X_i(t) + F \times r \times \gamma \times (X_{i,best}(t) - X_i(t) \\ + F \times r \times \alpha \times \\ (S_i(t) \times (X_{best}(t) - X_{i,j}(t)) \\ \gamma = \beta \times exp\left(-\frac{F_{best}(t) + \varepsilon}{F_{i,j}(t) + \varepsilon}\right), \varepsilon = 0.05, \alpha = 1 \end{aligned} \tag{18}$$

where, r represents a random number between (0,1); F is a symbolic function, which has the numerical value of 1 or -1; $F_{best}(t)$ is the fitness value of the X_{best}; $F_{i,j}(t)$ is the fitness of the $X_i(t)$; β is constant and fixed to 1; γ is the force generated when the gas particle j in cluster i is interacting with the particles in other clusters. With the increase of t, γ and $S_i(t)$ will gradually decrease, the search range of particles will constantly decrease, and HGSO algorithm will gradually change from global search mode to local search mode.

In order to avoid falling into the local optimum during each iteration, after the particles complete the position update by Eq. (18) during each iteration, some particles with poor fitness values in each cluster are reset. The number of selected particles is defined as:

$$N_w = N \times (rand(c_2 - c_1) + c_1), c_1 = 0.1, c_2 = 0.2 \tag{19}$$

where, N is the population number. These selected poor particles are reset by:

$$X_{i,worst} = LB + rand \times (UB - LB) \tag{20}$$

where, LB and UB are the prescribed upper and lower bounds, $X_{i,worst}$ is the selected worse particle.

Lévy-HGSO based on Lévy motion, Brown-HGSO based on Brownian motion and BL-HGSO Lévy motion and Brownian motion are respectively formed based on Lévy flight operator and Brownian motion operator to enhance the global and local search capabilities of HGSO algorithm.

B. Archimedes Optimization Algorithm

Archimedes optimization algorithm (AOA) was proposed by Fatma A. Hashim [12], which comes directly from the Archimedes law of physics. In AOA, objects immersed in water can be seen at the initial population. The important parameters of the algorithm are their own unique density, volume, and the acceleration generated during motion. The idea of AOA is to make the combined external force of the fluid as equal to zero as possible. If the buoyancy F_b is as large as the mass of the object W_o, the object is in equilibrium.

$$\begin{aligned} F_b = W_o \\ p_b v_b a_b = p_o v_o a_o \end{aligned} \tag{21}$$

where, p represents the density, v represents the volume, a represents the density gravity or acceleration. Subscript b and o represent fluid and immersed object. This equation can also be rewritten as:

$$a_o = \frac{p_b v_b a_b}{p_o v_o a_o} \tag{22}$$

If there are other forces acting on the object and there is a collision with other neighbors (r), the equilibrium state is defined in Eq. (23).

$$\begin{aligned} F_b &= W_o \\ W_b - W_r &= W_o \\ p_b v_b a_b - p_r v_r a_r &= p_o v_o a_o \end{aligned} \tag{23}$$

AOA is similar to other swarm intelligence optimization algorithms. The initial population have immersed objects, and AOA also starts the random search process from the initial population with random volume, density, and acceleration (candidate solution). In this phase, the object immersed in the fluid updates the density, volume, and acceleration. The acceleration of an object is updated in terms of its collision with other adjacent objects. Taking the fitness of the initial population as the evaluation index, 1000 iterations are conducted until the termination condition is reached. Firstly initialize the location of all individuals with Eq. (24).

$$O_i = lb_i + rand \times (ub_i - lb_i); i = 1,2\cdots, N \tag{24}$$

where, O_i is the i- th object in the N object population; ub_i and lb_i are the upper and lower limits of the searching space respectively. Initialize the density, volume of the objects based on Eq. (25).

$$\begin{aligned} den_i &= rand \\ vol_i &= rand \end{aligned} \tag{25}$$

where, $rand$ is a D-dimension vector, which randomly generates numbers between [0,1]. Finally initialize the acceleration with Eq. (26). In this process, the population is evaluated to select the target for the best fitness. Notes are given as x_{best}, den_{best}, vol_{best}, acc_{best}.

$$acc_i = lb_i + rand \times (ub_i - lb_i) \tag{26}$$

Individual i is updated with Eq. (27) during the $t+1$ iteration.

$$\begin{aligned} {den_i}^{t+1} &= {den_i}^{t+1} + rand \times \ (den_{best} - {den_i}^{t}) \\ {vol}^{t+1} &= {vol}^{t+1} + rand \times \ (vol_{best} - {vol_i}^{t}) \end{aligned} \tag{27}$$

where, vol_{best} and den_{best} are by far the best volume and density, and $rand$ is a uniformly distributed random number.

In the early stages of motion, objects collide between them, and after a period of time, objects try to reach equilibrium. This function is implemented with transfer operator TF defined in Eq. (28), which can better transform the exploration into exploitation.

$$TF = exp\frac{t - t_{max}}{t_{max}} \tag{28}$$

TF increases with time until it reaches 1. t and t_{max} are the number and maximum number of iterations, respectively. Similarly, density factor d can also help AOA search transition from global to local.

$$d^{t+1} = exp\left(\frac{t - t_{max}}{t_{max}}\right) - \frac{t}{t_{max}} \tag{29}$$

d^{t+1} decreases over time, and it should be noted that the use of d^{t+1} correctly ensures the balance of AOA in exploration and exploitation. If $TF \leq 0.5$, collisions will occur between the objects. A random material (mr) was selected to update the material of the object. The acceleration update strategy was realized by Eq. (30).

$${acc_i}^{t+1} = \frac{den_{mr} + vol_{mr} + acc_{mr}}{{den_i}^{t+1} \times {vol_i}^{t+1}} \tag{30}$$

where, den_i, vol_i and acc_i are density, volume and acceleration of the object i, respectively; den_{mr}, vol_{mr} and acc_{mr} are density, volume, and acceleration of random materials, respectively. If $TF > 0.5$, there will be no collision between the objects, and the acceleration strategy becomes different. The acc_{best} is the acceleration of the best individual.

$${acc_i}^{t+1} = \frac{den_{best} + vol_{best} + acc_{best}}{{den_i}^{t+1} \times {vol_i}^{t+1}} \tag{31}$$

The acceleration was normalized by using Eq. (32).

$${acc_{i-norm}}^{t+1} = u \times \frac{{acc_i}^{t+1} - min\ (acc)}{max\ (acc) - min\ (acc)} + l \tag{32}$$

where, both u and l are the normalized range and are set to 0.9 and 1. If the individual i is far from the global optimum, the acceleration will be larger, which indicates that it is in the exploration stage. Otherwise, it is in exploitation stage. This also explains how the search transforms from global to local search. If $TF \leq 0.5$, AOA belongs to the large-scale exploration stage and Eq. (33) is used to update the target i^{th}.

$${x_i}^{t+1} = {x_i}^{t} + C_1 \times rand \times {acc_{i-norm}}^{t+1} \times d \times (x_{rand} - {x_i}^{t}) \tag{33}$$

where, C_1 is the fixed coefficient equal to 2. On the contrary, if $TF > 0.5$, it belongs to the local exploitation stage and Eq. (34) is used to realize the position update.

$${x_i}^{t+1} = {x_{best}}^{t} + F \times C_2 \times rand \times {acc_{i-norm}}^{t+1} \times d \times (T \times x_{best} - {x_i}^{t}) \tag{34}$$

where, C_2 is a fixed coefficient equal to 6; T is directly proportional to the time and the transfer operator, $T = C_3 \times TF$ and T increases in the range of $[C_3 \times 0.3,\ 1]$; F is a sign of changes the direction of movement by using Eq. (35).

$$F = \begin{cases} +1\ if\ p \leq 0.5 \\ -1\ if\ p > 0.5 \end{cases} \tag{35}$$

where, $P = 2 \times rand - C_4$. The object function was used to evaluate each object in the population, and record the optimal solution found so far. Take notes x_{best}, den_{best}, vol_{best}, acc_{best}.

The buoyancy energy was introduced into the original AOA as a judgment condition for performing the global or local search. Then Lévy flight operator and Tangent flight motion operator are introduced into the position update strategy, respectively. Thus AOA based on buoyant energy (EAOA), AOA based on Lévy motion and buoyancy energies (LevyEAOA), AOA based on Brownian motion and buoyancy energies (TanEAOA) and AOA based on Lévy

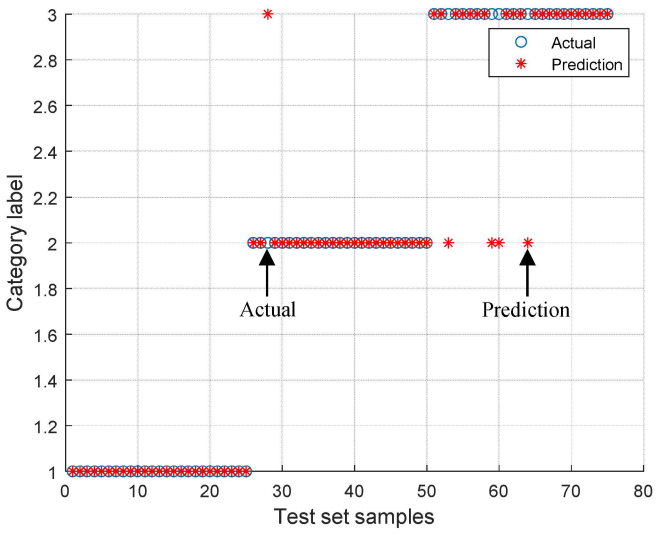
flight and tangent flight (BL-HGSO) were constituted so as to enhance the global search capabilities and local search capabilities of AOA.

IV. Simulation Experiment and Result Analysis

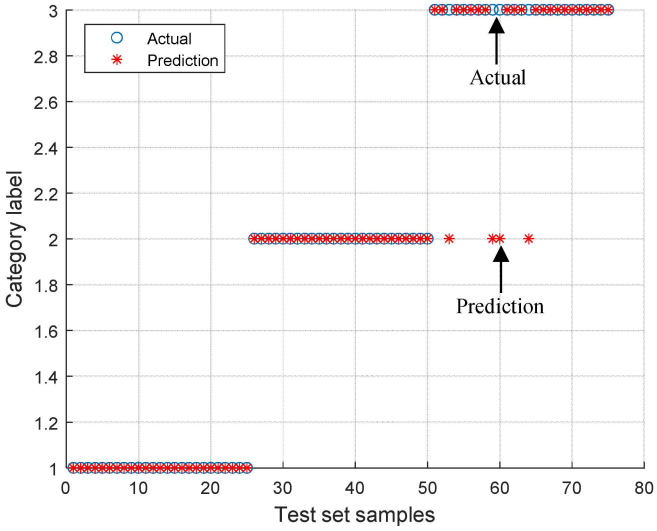
The comprehensive performance of the physical phenomenon-based swarm intelligent optimization algorithm is tested by selecting different datasets from UCI database. To verify the robustness of the methods, ten simulation experiments were repeated, and the mean and standard deviation of the classification accuracy were recorded. The maximum number of experimental iterations was 200. The initialized population is placed in two-dimension space for optimization test. Each individual position represents two important parameters (C,γ). The optimal solution to find in the classification problem is classification accuracy. Various algorithms are used to optimize the parameters of SVM model, and the best combination of parameters is selected for training so that the test set can have less classification errors.

A. Classification of Iris Datasets

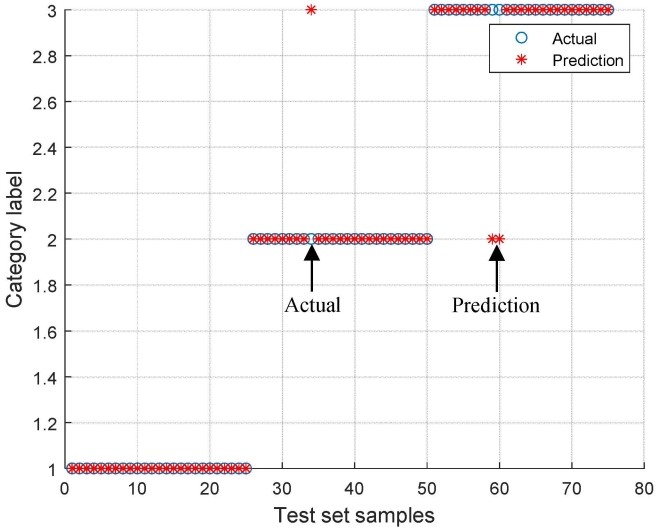
To verify the feasibility of the classification of SVM in practical problems, the Iris data set in the UCI database is firstly selected. The Iris datasets contains three flower types: Iris-versicolor, Iris-virginica, and Iris-setosa. The Iris database has a total of 150 samples, and the above three types of flowers are all 50. There are four characteristic values in the iris database: petal length, petal width, sepal length, and sepal width, which all measure in centimeters. To facilitate the planning of the training and testing models, the datasets is first divided into the training and test sets. To reduce the impact between different data, the data are normalized. All data were normalized to the interval [0,1] by using the function mapminmax. At this time, nine algorithms were used to optimize the penalty factor C and the kernel parameter g . These include HGSO, Lévy-HGSO, Brown-HGSO, BL-HGSO, AOA, EAOA, TanEAOA, LévyEAOA, and TanLévyEAOA. The classification diagrams for each algorithm are shown in Fig. 1, and the specific classification accuracy is listed in Table 1.



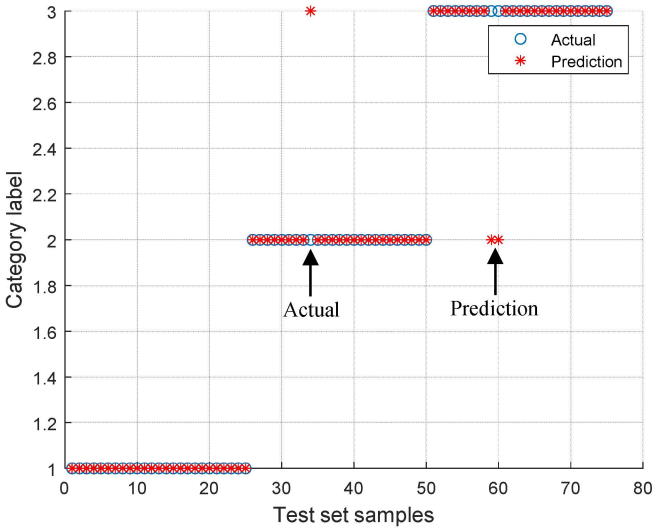
(a) AOA



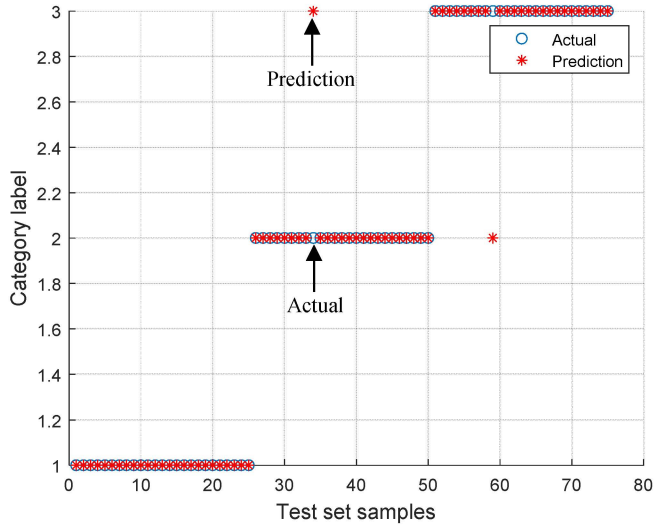
(b) EAOA



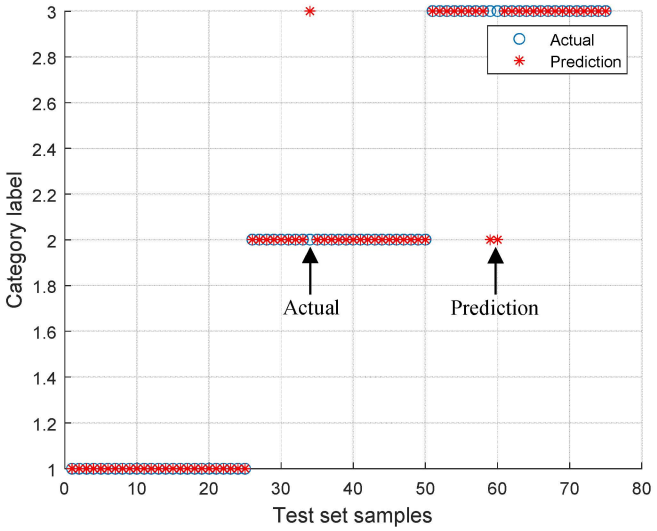
(c) levyEAOA



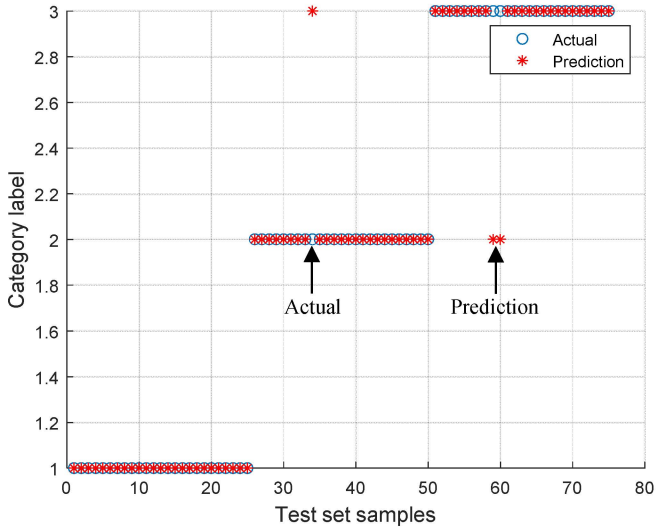
(d) tanEAOA



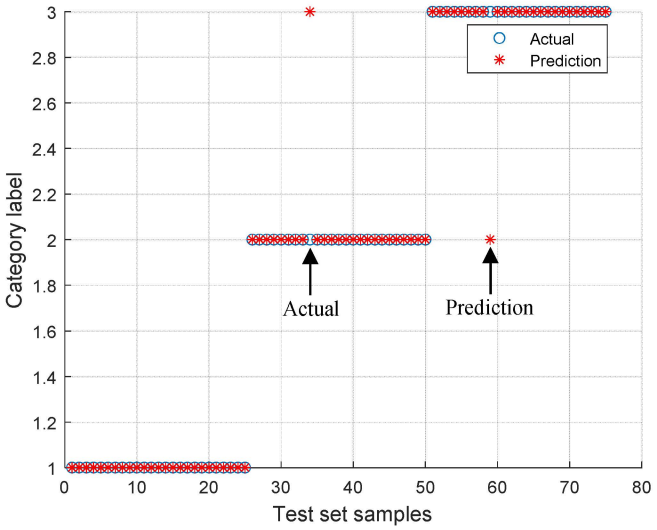
(e) tanlevyEAOA



(h) BrownGSO

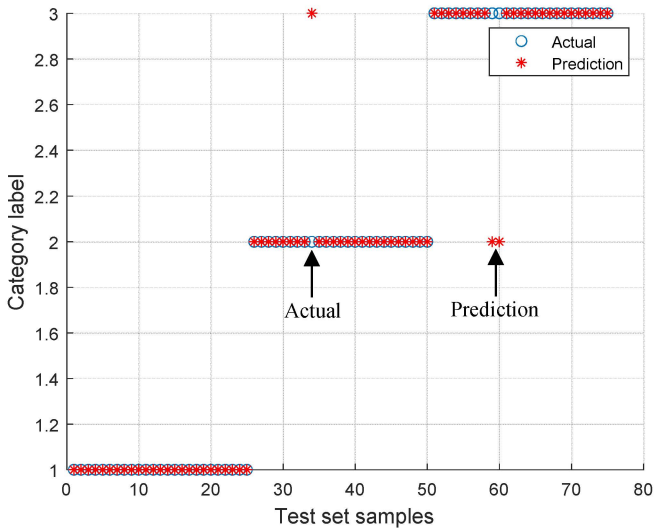


(f) HGSO



(i) BLHGSO

Fig. 1 Classification results on iris datasets.



(g) LevyHGSO

TABLE 1. CLASSIFICATION ACCURACY ON IRIS DATASETS

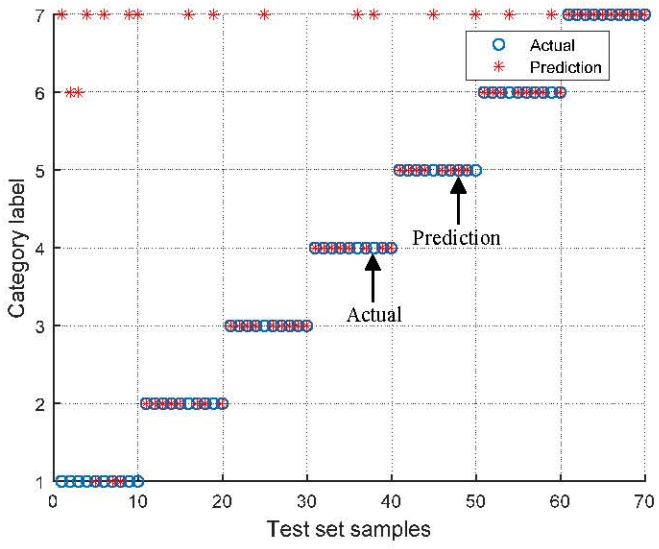
Algorithm	MSE（AVE+STD）	Accuracy rate	Number of correct classifications
AOA	0.1554±0.0932	0.9333	（70/75）
EAOA	0.1511±0.0364	0.9333	（70/75）
LévyEAOA	0.1541±0.1364	0.96	（72/75）
TanEAOA	0.1491±0.0364	0.96	（72/75）
TanLévyEAOA	0.1523±0.1679	0.9733	（73/75）
HGSO	0.1516±0.2441	0.96	（72/75）
LévyHGSO	0.1552±0.1038	0.96	（72/75）
BrownHGSO	0.1543±0.0789	0.96	（72/75）
BLHGSO	0.1531±0.1317	0.9733	（73/75）

It can be seen from Table 1 and Fig. 1 that 9 kinds of algorithms can clearly realize the classification on Iris data set. The algorithms with highest accuracy are the BLHGSO algorithm and TanLévyHGSO algorithm, whose accuracy reached 97.33%. The second most effective algorithms are Brown-HGSO，Lévy-HGSO，LévyEAOA，TanEAOA with the accuracy of 96%. The most common effect is the original algorithm (AOA) with the accuracy of 93%.

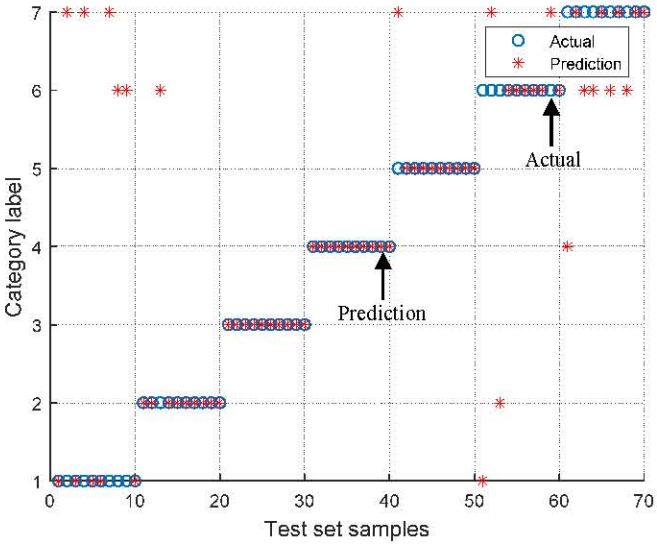
B. Classification of Strip Surface Defect Datasets

In this subsection, it is verified whether the SVM model optimized by multiple algorithms can accurately classify the strip surface defects. Ribbon surface defect type can be roughly summarized as surface stains, damage, etc. Strip surface defect data set in UCI database with 1941 samples is selected to carry out simulation experiments. Each sample has 27 characteristic values, such as Xmax, steel plate thickness, edge index and 300 type steel, and there are 7 different types of defects. If these defects are correctly classified, it will be easily and greatly improve production efficiency. Similarly, this data set must be divided into training set and testing set, and all sample data must be shuffled and randomly divided into 7 groups according to 7 different types of defects. Each group takes 10 testing data, and the rest of the data as training sets. This partition method is more universal and reduces the effect of randomness. The two important parameters of SVM are optimized with various algorithms. Finally, the classification accuracy can be obtained. The experiment will be repeated many times to avoid contingency. The classification curves are shown in Fig. 2 and the specific classification accuracy are listed in Table 2.

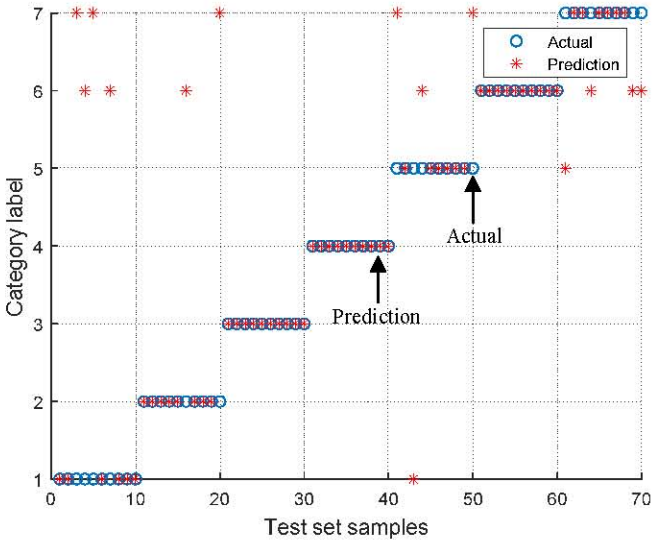
It can be seen from Fig. 2 and Table 2 that the best effect is the BL-HGSO algorithm that combines Lévy motion and Brownian motion. Its classification accuracy reaches 85.71%, which shows that the algorithm can correctly classify the surface defects of strip steel products. However, due to the complexity of the strip surface defects data set, there will be 10 incorrectly classified data. The algorithms with second highest effect are the Brown-HGSO algorithm, TanEAOA and TanLévyEAOA, whose accuracy is about 80%. These three algorithms are also improved on the basis of the original HGSO algorithm and AOA.



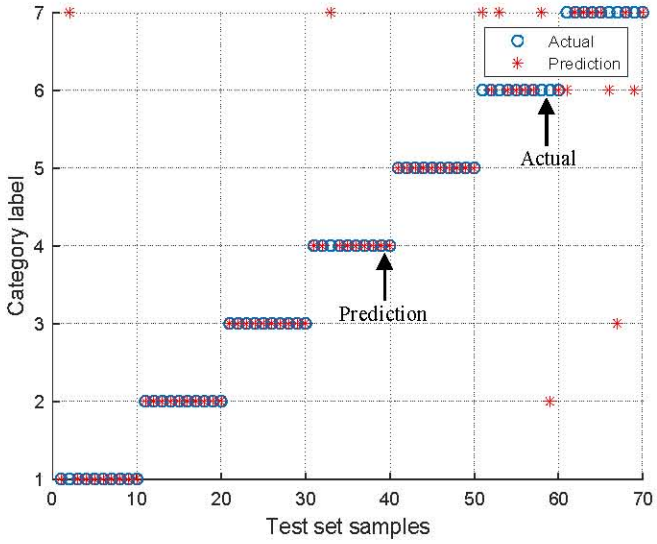
(a) HGSO



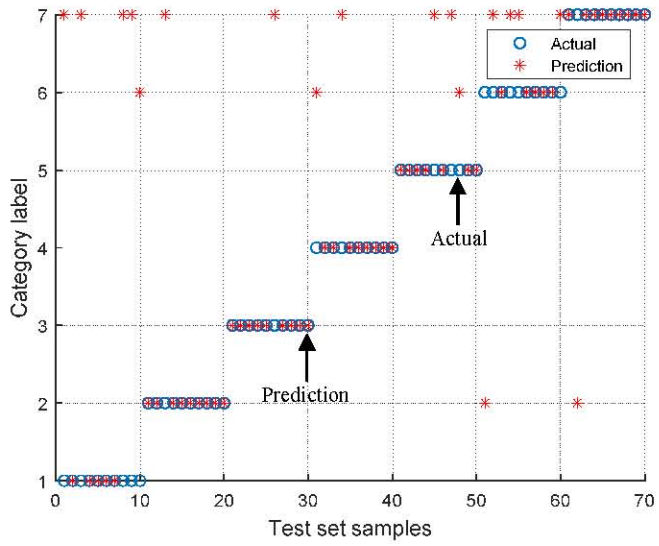
(b) LevyHGSO



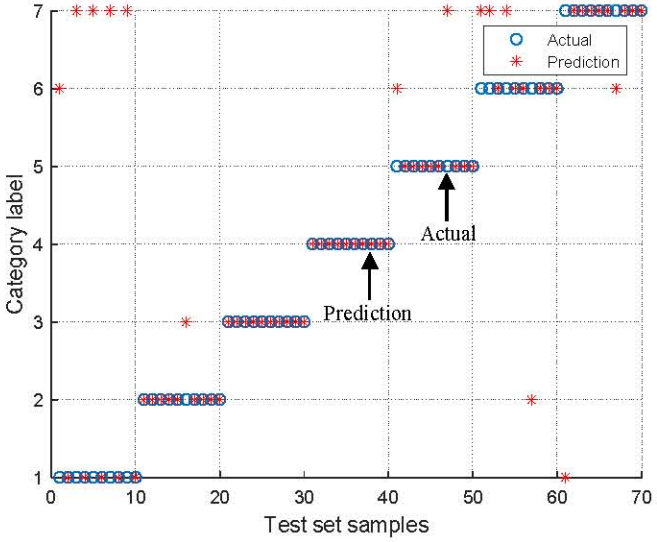
(c) BrownGSO



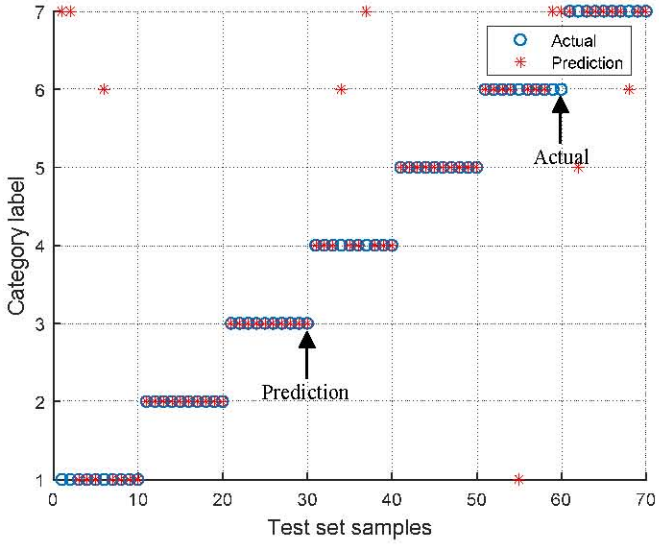
(d) BLHGSO



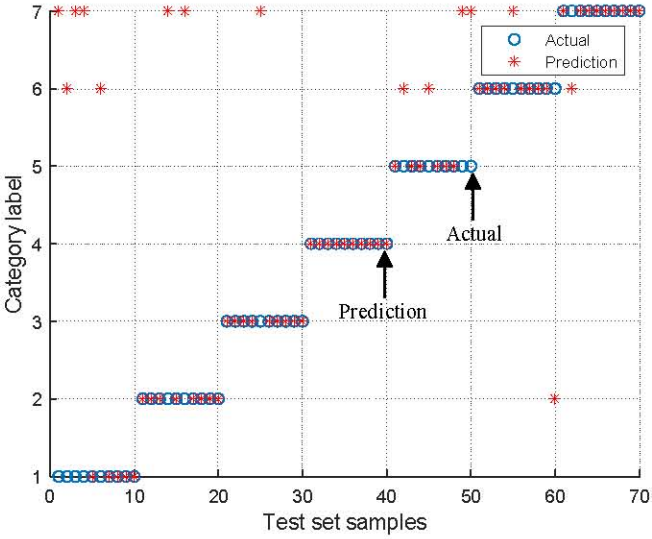
(e) AOA



(h) levyEAOA

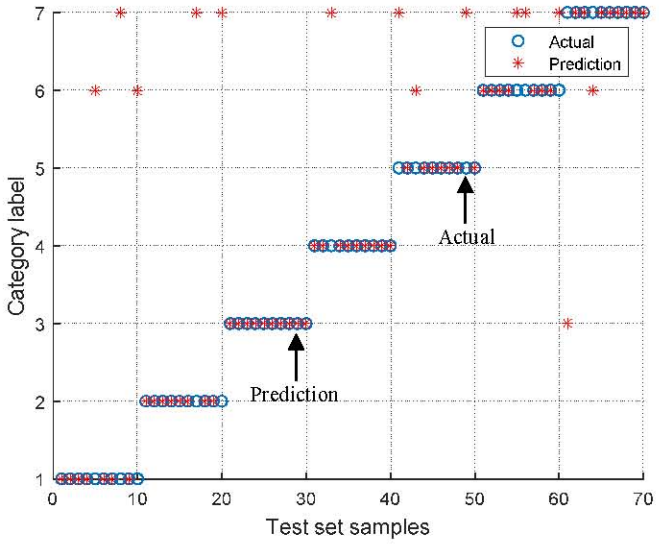


(f) tanlevyEAOA



(i) EAOA

Fig. 2 Classification results on strip surface defect datasets.



(g) tanEAOA

TABLE 2. CLASSIFICATION ACCURACY ON STRIP SURFACE DEFECT DATASETS

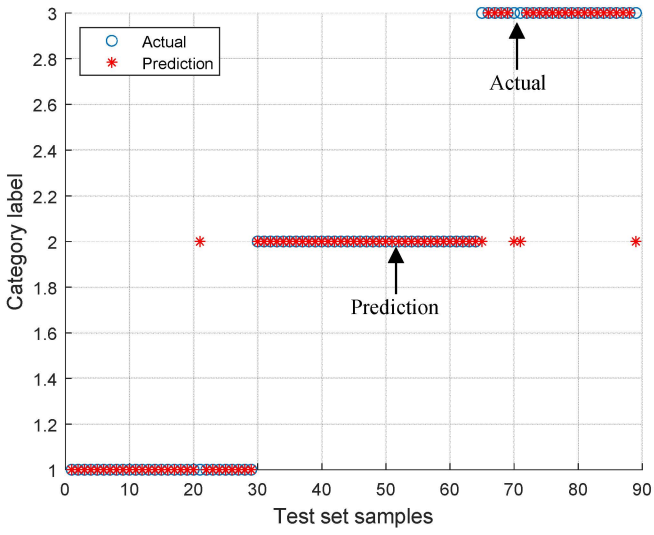
Algorithm	MSE（AVE+STD）	Accuracy rate	Number of correct classifications
AOA	0.2017±0.0742	0.7571	（53/70）
EAOA	0.2595±0.0364	0.7857	（55/70）
LévyEAOA	0.1441±0.1004	0.7714	（54/70）
TanEAOA	0.1814±0.0148	0.80	（56/70）
TanLévyEAOA	0.1217±0.081	0.8143	（57/70）
HGSO	0.1510±0.033	0.7714	（54/70）
LévyHGSO	0.1420±0.1997	0.7857	（55/70）
BrownHGSO	0.1374±0.0897	0.80	（56/70）
BLHGSO	0.1533±0.0471	0.8571	（60/70）

It can also be seen that compared with the original algorithms, the accuracy has also been improved. The algorithms in the third gear are EAOA and Lévy-HGSO algorithm, with an accuracy of about 78%. Compared with the basic algorithms, all kinds of improved algorithms do play an obvious optimization role, and the classification speed is also relatively faster, about 0.38 seconds.

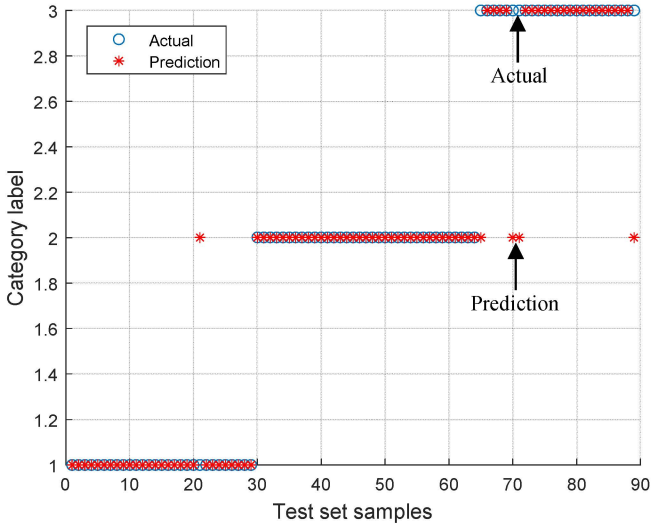
C. *Classification of Wine Datasets*

The composition of a wine is often more complex, so it is very important to correctly distinguish the types of wine according to its composition. In this paper, wine types are separated based on SVM optimized by swarm intelligent algorithms. Similarly, the wine data set in the UCI database was the first imported. According to the adopted wine data set, there are 178 samples and wine are divided into three categories. There are 13 eigenvalues in total. Firstly, the missing data in the data set should be filled up and the outliers of the data are corrected. The process of data cleaning is to judge discrete values and noise values. Since the wine datasets is continuous, there is no need to fill in the vacant values. The labels for the corresponding testing data are also separated from the overall labels. The simulation results on wine classification are shown in Fig. 3, and the classification accuracy of each algorithm is listed in Table 3. It can be clearly seen from the above simulation results that when classifying red wine datasets, TanLévyEAOA and BLHGSO algorithm, which combined the advantages of other algorithms, have the highest accuracy, reaching 98.87%. TanLévyEAOA integrates the advantages of the original AOA and several other operators, while BLHGSO algorithm integrates the advantages of the original HGSO algorithm, LévyHGSO algorithm and BrownHGSO algorithm. These two algorithms show superior performance in SVM model classification.

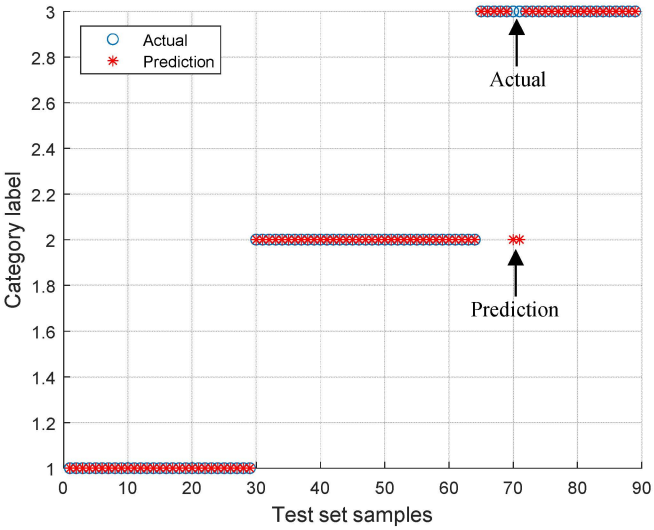
Three algorithms (BrownHGSO, LévyHGSO, TanEAOA) rank in the second echelon of effectiveness. The accuracy of these three algorithms is also good, reaching 97.75%. The third echelon algorithms are EAOA, AOA and HGSO algorithm，whose accuracy is located between 94% and 96%. It can be seen that there is an obvious performance gap with other improved algorithms.



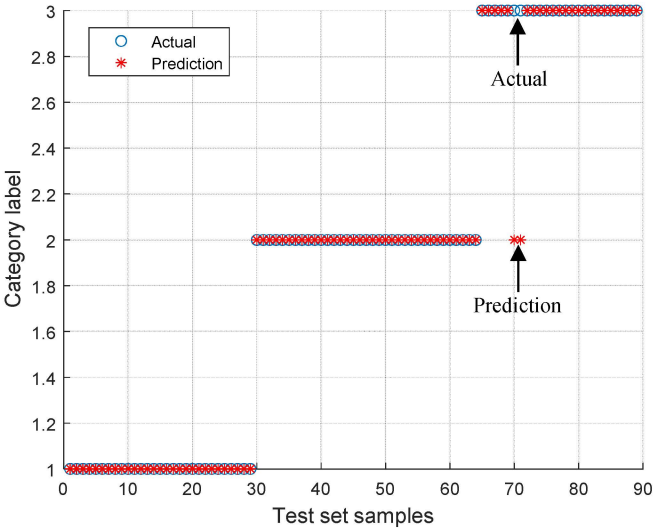
(a) AOA



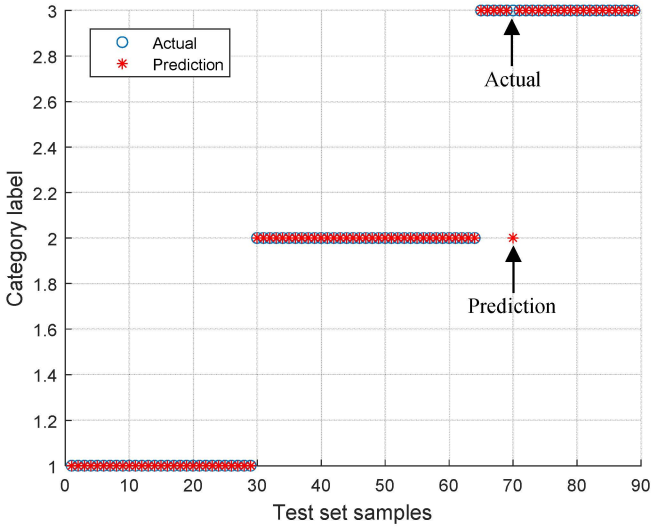
(b) EAOA



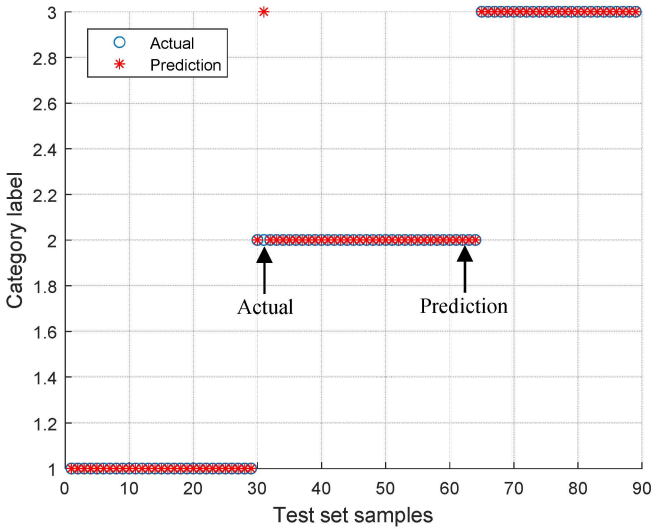
(c) tanEAOA



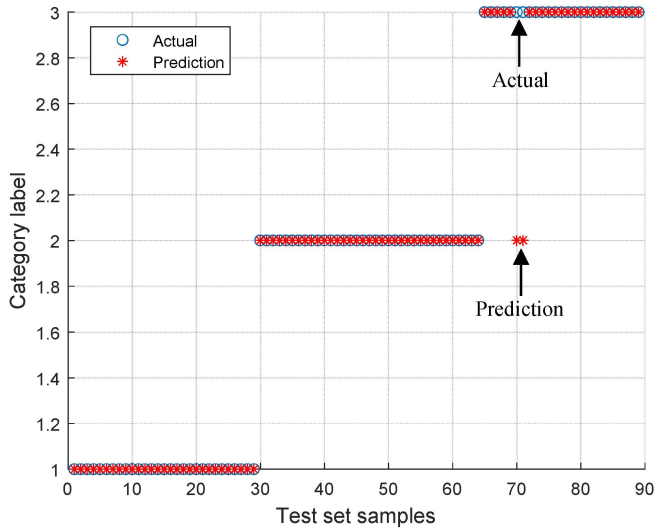
(d) levyEAOA



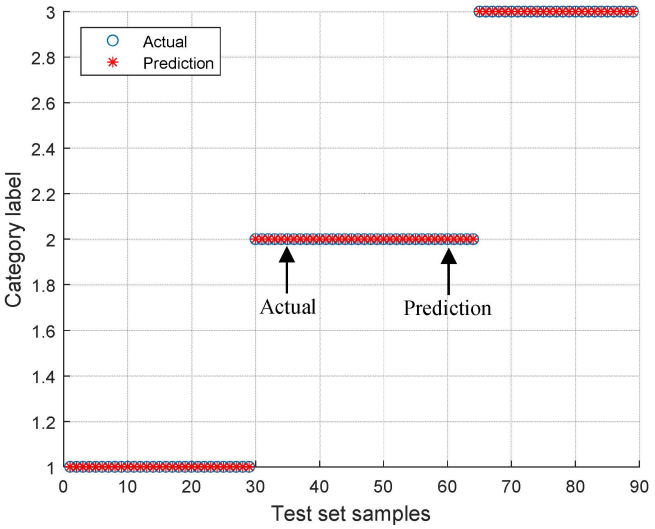
(e) tanlevyEAOA



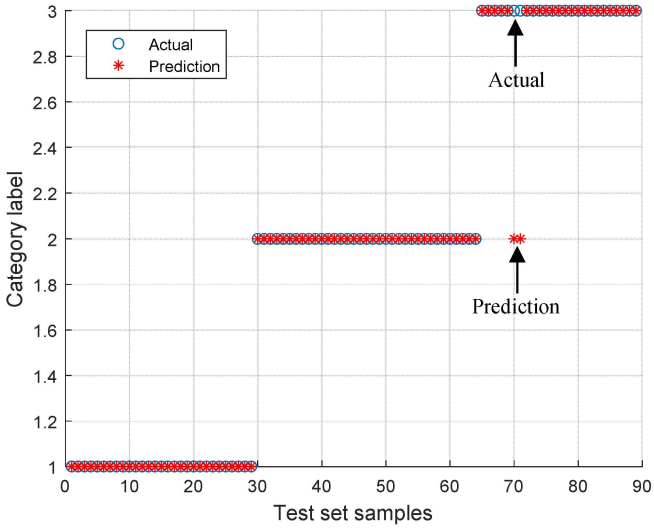
(h) BrownGSO



(f) HGSO



(i) BLHGSO



(g) LevyHGSO

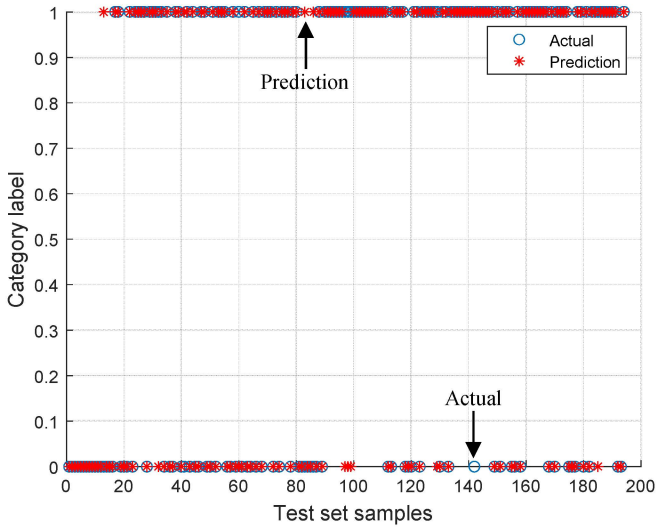
Fig. 3 Classification results on wine datasets.

TABLE 3. CLASSIFICATION ACCURACY ON WINE DATASETS

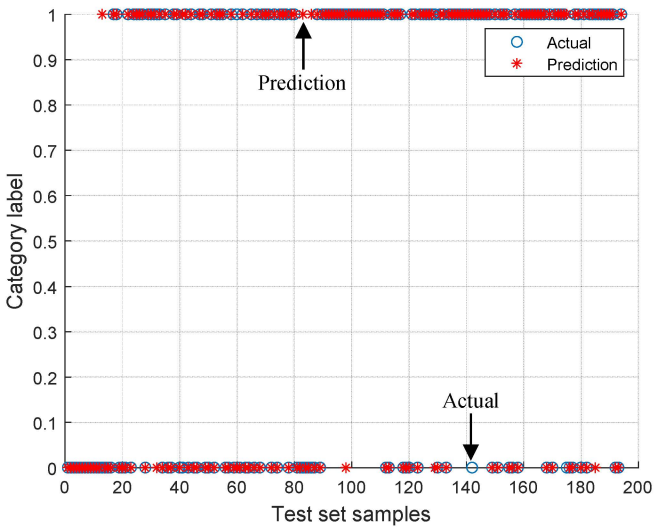
Algorithm	MSE（AVE+STD）	Accuracy rate	Number of correct classifications
AOA	0.1157±0.0620	0.9438	（84/89）
EAOA	0.1563±0.0472	0.9550	（85/89）
LévyEAOA	0.1237±0.1980	0.9550	（85/89）
TanEAOA	0.1820±0.0421	0.9775	（87/89）
TanLévyEAOA	0.1139±0.1451	0.9887	（88/89）
HGSO	0.1881±0.0991	0.9663	（86/89）
LévyHGSO	0.1507±0.0743	0.9775	（87/89）
BrownHGSO	0.1337±0.1035	0.9775	（87/89）
BLHGSO	0.1539±0.1294	0.9887	（88/89）

D. Classification of Breast Cancer Wisconsin Datasets

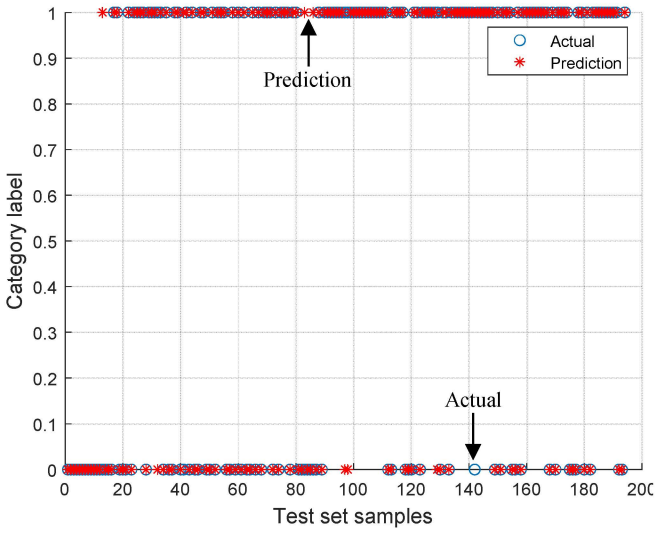
Breast cancer is one of the most common diseases among women worldwide, accounting for 15% of all female cancers. Although there are risks, breast cancer can be controlled through early prevention. Evaluation of data obtained from patients and physicians is the most valuable element in diagnostic pathology. In view of the above requirements, data classification can be used to eliminate the errors that doctors may make in the diagnosis process to a certain extent, and medical statistical data can be analyzed. The Wisconsin Breast Cancer data set from the UCI database is adopted to carry out simulation experiments, which contains 569 different examples and 32 attributes. In the process of data processing, 375 samples were selected as training data and the remaining 194 samples as testing data. Fig. 4 shows the classification diagram and Table 4 lists the specific accuracy rate of each algorithm for Wisconsin datasets of breast cancer data set.



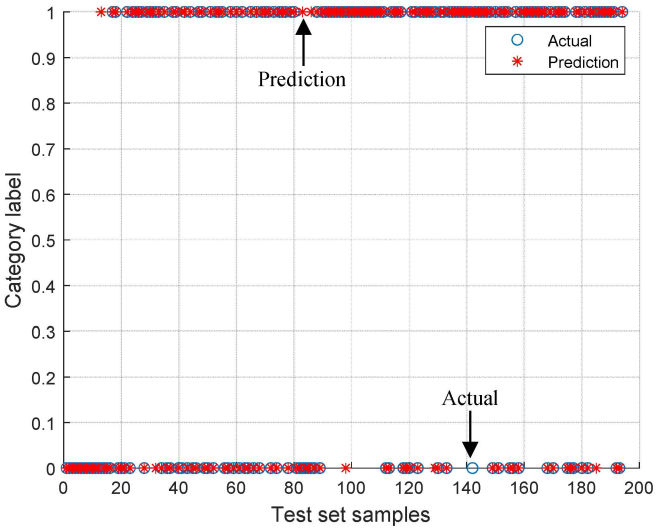
(a) AOA



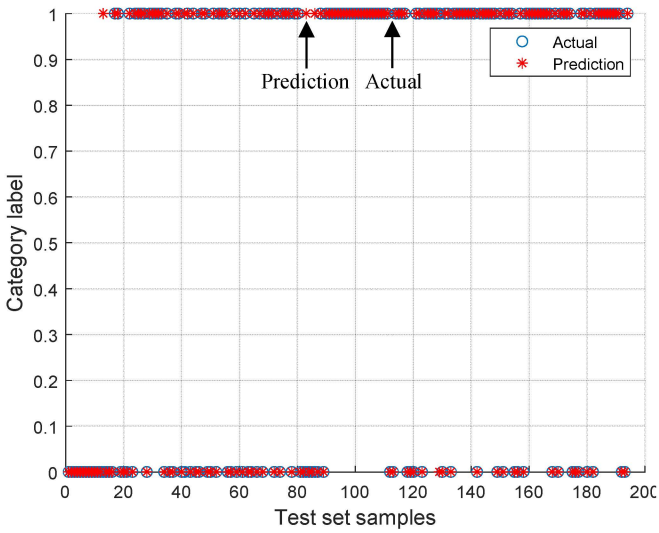
(b) EAOA



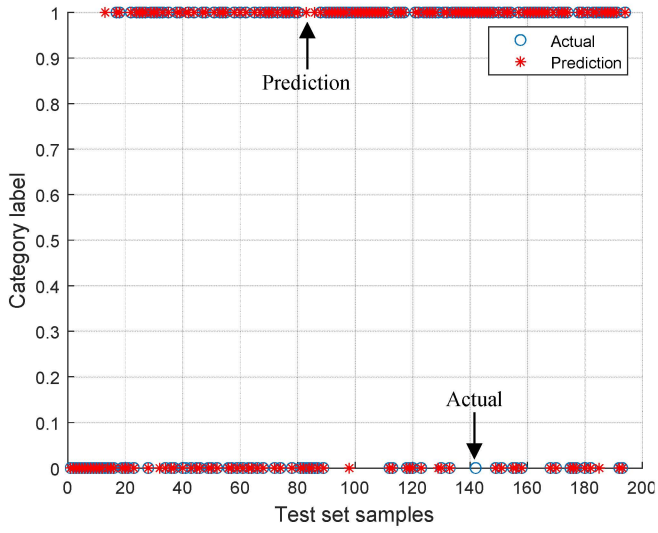
(c) levyEAOA



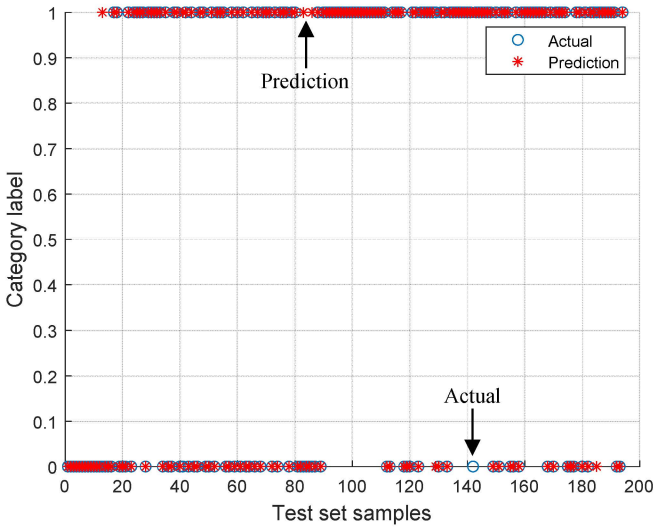
(d) tanEAOA



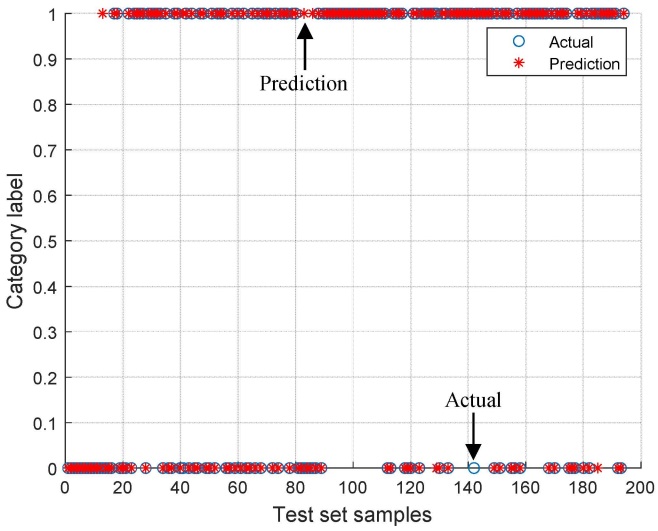
(e) tanlevyEAOA



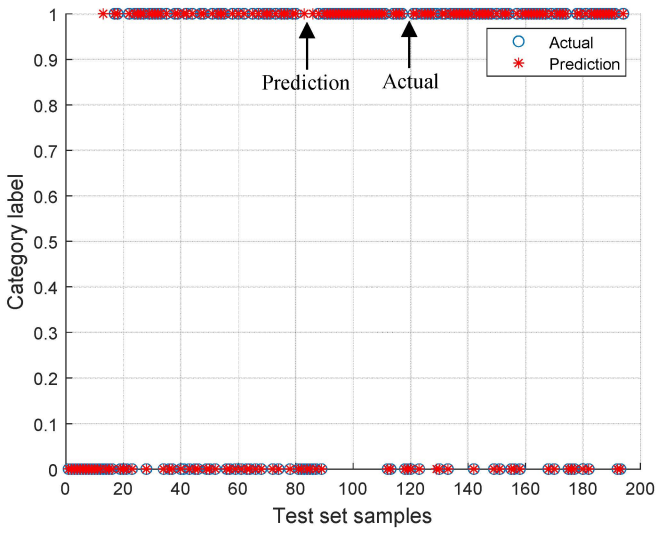
(f) HGSO



(g) LevyHGSO



(h) BrownGSO



(i) BLHGSO

Fig. 4 Classification results on Wisconsin datasets of breast cancer.

TABLE 4. CLASSIFICATION ACCURACY ON WISCONSIN DATASETS OF BREAST CANCER

Algorithm	MSE（AVE+STD）	Accuracy rate	Number of correct classifications
AOA	0.1305±0.0442	0.9329	（181/189）
EAOA	0.1347±0.1530	0.9432	（183/194）
LévyEAOA	0.1341±0.0963	0.9639	（187/194）
TanEAOA	0.1574±0.1882	0.9536	（185/194）
TanLévyEAOA	0.1523±0.1679	0.9793	（190/194）
HGSO	0.1117±0.0708	0.9536	（185/194）
LévyHGSO	0.1646±0.0981	0.9639	（187/194）
BrownHGSO	0.1543±0.0789	0.9639	（187/194）
BLHGSO	0.1538±0.091	0.9742	（189/194）

V. CONCLUSIONS

In this paper, nine algorithms are applied to optimize SVM parameters so as to deal with classification problems. Four representative datasets were selected, including iris in the field of plants, strip steel surface defect in the field of industrial production, red wine in the field of food, and Wisconsin breast cancer in the field of medicine. It can be seen from the simulation experiments that the improved algorithms have significantly effect compared with the original HGSO algorithm and AOA to find more excellent and accurate solutions in classification problems. BLHGSO and TanLevyHGSO are the two algorithms with the best performance. The former integrated the advantages of LevyHGSO and BrownHGSO, while the latter integrated the advantages of EAOA, TanEAOA and LevyEAOA. These algorithms also have certain reference value in practical classification problems.

REFERENCES

[1] L. Abualigah, A. Diabat, S. Mirjalili, M. A. Elaziz, and A. H. Gandomi, "The Arithmetic Optimization Algorithm," *Computer Methods in*

Applied Mechanics and Engineering, vol. 376, pp. 113609.1-113609.38, 2021.

[2] H. Shareef, A. A. Ibrahim, and A. H, Mutlag, "Lightning Search Algorithm," *Applied Soft Computing*, vol. 26, pp. 315-333, 2015.

[3] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A Gravitational Search Algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232-2248, 2009.

[4] I. A. Ibrahim, M. J. Hossain, B. C. Duck, and M. Nadarajah, "An Improved Wind Driven Optimization Algorithm for Parameters Identification of a Triple-dide Photovolatic Cell Model," *Energy Conversion and Management*, vol. 213, pp. 112872, 2020

[5] W. Zhao, L. Wang, and Z. Zhang, "A Novel Atom Search Optimization for Dispersion Coefficient Estimation in Groundwater," Future Generation Computer Systems, vol. 91, pp. 601-610, 2019.

[6] H. R. E. H. Bouchekara, "Electric Charged Particles Optimization and Its Application to the Optimal Design of a Circular Antenna Array," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1767-1802, 2021.

[7] F. A. Hashim, E. H. Houssein, S. M. Mai, W. Al-Atabany, and S. Mirjalilim, "Henry Gas Solubility Optimization: a Novel Physics-based Algorithm," Future Generation Computer System, vol. 101, pp. 646-667, 2019.

[8] W. Xie, C. Xing, J. S. Wang, S. S. Guo, M. W. Guo, and L. F. Zhu, "Hybrid Henry Gas Solubility Optimization Algorithm Based on the Harris Hawk Optimization," *IEEE Access*, vol. 8, pp. 144665-144692, 2020.

[9] M. A. Elaziz, and I. Attiya, "An Improved Henry Gas Solubility Optimization for Task Scheduling in Cloud Computing," *Artificial Intelligence Review*, vol. 54, pp. 3599-3637, 2021.

[10] Betül Sultan Yldz, N. Pholdee, N. Panagant, S. Bureerat, and S. M. Sait, "A Novel Chaotic Henry Gas Solubility Optimization Algorithm for Solving Real-word Engineering Problems," *Engineering with Computers*, vol. 38, pp. 871-883, 2021.

[11] D. Mohammadi, M. Abd Elaziz, R. Moghdani, E. Demir, and S. Mirjalili, "Quantum Henry Gas Optimization Algorithm for Global Optimization," *Engineering with Computers*, vol. 38, pp. 2329-2348, 2022

[12] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes Optimization Algorithm: a New Metaheuristic Algorithm for Solving Optimization Problems," *Applied Intelligence*, vol. 51, no. 3, pp. 1531-1551, 2021.

[13] B. Yao, and H. Hosein, "Model Parameters Estimation of a Proton Exchange Membrane Fuel Cell Using Improved Version of Archimedes Optimization Algorithm," *Energy Reports,* vol. 7, pp. 5700-5709, 2021.

[14] E. H. Houssein, E. D. Helmy, H. Rezk, and A. M. Nassef, "An Enhanced Archimeds Optimization Algorithm Based on Local Escaping Operator and Orthogonal Learning for PEM Fuel Cell Parameter Identification," *Engineering Applications of Artificial Intelligence*, vol. 103, pp. 104309, 2021.

[15] C. W. Hsu, and C. J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.

[16] W. Z. Sun, Y. Ma, Z. Y. Yin，J. S. Wang，A. Gu，and F. J. Guo, "WOA-MLSVMs Dirty Degree Identification Method Based on Texture Features of Paper Currency Images," *IAENG International Journal of Computer Science*, vol. 48, no. 4, pp. 952-964, 2021.