IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 March 2023, accepted 13 March 2023, date of publication 20 March 2023, date of current version 3 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3259548

RESEARCH ARTICLE

Hybrid Strategy Improved Sparrow Search Algorithm in the Field of Intrusion Detection

LIU TAO[1,2] AND MENG XUEQIANG[1]
[1]College of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an, Shaanxi 710600, China
[2]Xi'an Key Laboratory of Heterogeneous Network Convergence Communication, Xi'an, Shaanxi 710000, China

Corresponding author: Meng Xueqiang (mxueqiang1996@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0808, and in part by the Key Research and Development Projects in Shaanxi Province under Grant 2019ZDLSF07-06.

ABSTRACT Aiming at the problem that Sparrow Search Algorithm(SSA) may fall into local optima and have slow convergence speed, a hybrid strategy improved sparrow search algorithm(HSISSA) is proposed in this paper, and it is applied to feature selection and model optimization of intrusion detection. First, a hybrid circle-piecewise map is proposed to initialize the population and improve the uniformity of the initial population distribution; second, merging the spiral search method in the vulture search algorithm and Levy's flight formula to update the positions of the discoverer and scouter, respectively, to expand the population search range and enhance the search capability; and finally, the simplex method and pinhole imaging method are used to optimize the position of sparrows with poor fitness and optimal fitness, to avoid stagnation in the population search and fall into local optima. The performance of the algorithm was optimized using the aforementioned methods. The algorithm was tested on 10 classical benchmark functions and combined with Wilcoxon rank-sum test analysis to verify its effectiveness, which showed improvements in convergence speed and accuracy. Finally, it was applied to the feature selection and model optimization of intrusion detection. On average, 7.6 features and 10.1 features were retained on the CIC-IDS2017 and UNSW-NB15 datasets, respectively, and 99.5% and 96.01% accuracies were achieved. The number and accuracy of the optimized features were better than those of the original algorithm. For the DenseNet and random forest models, HSISSA achieved 99.34% and 97.22% accuracy after optimization, respectively, which improved the performance of the models. Thus, the algorithm showed a better performance than the other algorithms.

INDEX TERMS Sparrow search algorithm, spiral search, simplex method, pinhole imaging, intrusion detection, feature selection, DenseNet, random forest.

I. INTRODUCTION

In the current era of data and information explosion, the dimensionality of information on the internet is growing exponentially. In the fields of machine learning and data analysis, dealing with high-dimensional data is becoming increasingly difficult. A large number of irrelevant and redundant data features leads to inefficient and inaccurate model classification. Intrusion detection systems rely on network data to monitor and respond to malicious threats in real time. Therefore it is especially important to reduce redundant features and improve model accuracy. With the advantages of fast convergence rate, flexibility and robustness, swarm intelligence algorithms have attracted much attention. They can effectively solve various optimization and data mining problems, and in an intrusion detection system, the essence of feature selection and model parameter optimization is to solve combinatorial optimization problems. Therefore, swarm intelligence algorithms are widely used in the field of intrusion detection [1].

Over the past two decades,inspired by natural phenomena, many scholars have proposed a series of swarm intelligence algorithms. For example, the ant colony algorithm (ACO) [2], fruit fly optimization algorithm (FOA) [3], harris

The associate editor coordinating the review of this manuscript and approving it for publication was Kuo-Ching Ying.

hawks optimization (HHO) [4], whale optimization algorithm (WOA) [5], chimp optimization algorithm (ChOA) [6], colony predation algorithm (CPA) [7], rat swarm optimizer (RSO) [8], golden jackal optimization (GJO) [9] and sparrow search algorithm (SSA) [10]. SSA has the advantages of high precision and strong stability. However, the algorithm has the defects of large randomness and the possibility of falling into local optima.

In view of the shortcomings of SSA, scholars proposed a series of improvement measures. Tang et al. [11] used cat mapping to initialize the population and enhance the overall exploration ability, performed cauchy mutation and Tent chaotic disturbance for all locally optimal solutions that arise, finally, an adaptive adjustment strategy of discoverer-follower is proposed to balance the optimization ability of the early and late iterations of the algorithm and improve the convergence accuracy of the algorithm. Yuan et al. [12] used the center of gravity reverse learning to initialize the population, increased the distribution range of the initial solution, and added variable learning coefficients and mutation factors to improve the update formulas of discoverers and followers, which enhanced the global search capability. Ren et al. [13] proposed a sine cosine and firefly perturbed sparrow search algorithm (SFSSA). First, the tent map is used to initialize the population, and then the sine cosine algorithm with random inertia weights and the firefly perturbation algorithm are introduced to improve the ability to jump out of the local optimum. Ma et al. [14] propose an enhanced multi-strategies sparrow search algorithm (EMSSA). Proposing an uniform diversification orientation strategy to initialize the population, using hazard shift strategy to update the discoverer position to increase the global exploration ability and convergence accuracy, and making use of dynamic evolution strategy to perturb the optimal individual after the current iteration to avoid falling into a local optimum.

For the application of swarm intelligence algorithm in the field of intrusion detection, scholars have provided many methods. Zhang et al. [15] proposed an intrusion detection technology based on improved genetic algorithm and DBN, and adaptively generated the optimal number of hidden layers and neurons through genetic algorithm to adapt to different attack types. Dash [16] borrowed the idea of particle swarm algorithm, endowed the particles in gravitational search algorithm with memorability and information sharing, and optimized the BP network weight thresholds using improved gravitational search algorithm to ensure a high detection rate of attacks while having good stability. Li et al. [17] proposed an improved krill swarm algorithm (LNNLS-KH) based on the linear nearest neighbor lasso step for network intrusion detection feature selection.The algorithm can not only jump out of the local optimal solution quickly, but also show good performance in the number of feature selections, false alarm detection rate and other aspects.Du et al. [18] proposed an industrial control intrusion detection model based on optimized kernel limit learning machine, and jointly optimized the regularization coefficient C and kernel parameter g of KELM by improving the sparrow search algorithm. Experiments show that the algorithm has the advantages of high detection rate and low false positive rate.

Inspired by the above literature, this paper proposes a hybrid strategy improved sparrow search algorithm (HSISSA), the major contributions are as follows:

i. We proposed HSISSA improved the SSA from multiple perspectives:

- Initializing the population with the hybrid circle-piecewise map (CPM), this is used to enhance the uniformity of the initial population distribution.
- Adding the spiral search method to update the location of the discoverer and improve the global search performance.
- Using levi's flight disturbance mechanism to update the position of the scouter, reducing the impact that may fall into the local optimum.
- Combining the simplex method and the pinhole imaging reverse learning mechanism to update the position of poor and optimal sparrows respectively, so as to avoid the effects of stagnation and the possibility of multiple local optima, ultimately improving the optimisation capability of the algorithm.

ii. In the experiments, ten test functions and five other popular algorithms (gray wolf optimization algorithm (GWO), whale optimization algorithm (WOA), sine cosine optimization algorithm (SCA), particle swarm optimization (PSO) and sparrow search algorithm (SSA))were selected for comparison. All of the above algorithms work well for solving optimization problems, and the better performance of HSISSA is verified by comparing the results of function tests.

iii. We propose a feature selection method based on HSISSA and a model parameter optimization method based on DenseNet and Random Forest, apply them to intrusion detection, and conduct experiments on the CIC-IDS2017 dataset and UNSW-NB15 dataset to achieve the goal of feature dimension reduction and accuracy improvement.

The remainder of this paper is organized as follows. Section II introduces the relevant contents of the sparrow search algorithm. Section III introduces relevant content for intrusion detection. Section IV describes the proposed HSISSA algorithm in detail. Section V conducts performance tests of HSISSA, including comparison experiments with other algorithms. Section VI describes the application of HSISSA in intrusion detection, first, preprocessing the CIC-IDS2017 dataset and UNSW-NB15 dataset; second, setting the fitness function and using HSISSA for feature selection, and finally, optimizing a set of DenseNet network parameters and Random Forest model parameters to demonstrate the superior performance of HSISSA and analyze the experimental results. Section VII summarizes the study and discusses future directions for improvement.

II. SPARROW SEARCH ALGORITHM

The sparrow search algorithm builds a model by simulating the feeding behavior and anti-predation behavior of sparrows. According to the different division of labor, the sparrows in the population are divided into three roles: discoverer, follower and scouter. The discoverer has a wider search scope, can find food first and provide foraging directions for other sparrows; followers closely follow the discoverer to find food, and can seize the discoverer's food source, at this time, the two identities are exchanged; the scouters are responsible for monitoring the surrounding area and dealing with the coming danger.

Suppose the space consisting of N sparrows is $X = [x(1,i), x(2,i), \ldots x(N,i)]^T$, where $i = 1, 2, \ldots, d$ and d is the dimensionality. The discoverer's location update is described as shown in (1):

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t} \cdot exp(-\dfrac{i}{\alpha \cdot iter_{max}}) & if \;\; R_2 < ST \\ x_{i,j}^{t} + Q \cdot L & if \;\; R_2 \geq ST. \end{cases} \tag{1}$$

where, t represents the current number of iterations, $iter_{max}$ represents the maximum number of iterations, $x_{i,j}^{t}$ represents the information of the ith sparrow in the j dimension, α is a random number between(0, 1], Q is a random number subject to normal distribution, and L is a matrix of size and elements of 1, R_2 is a random number between [0, 1], and $ST \in [0.5, 1]$ represents the alert value. $R_2 < ST$ means that the foraging environment is safe, and the discoverer can search at will, $R_2 \geq ST$ means that the sparrow population is transferred to the safe direction after sensing the danger.

The description of follower's position update is shown in (2):

$$x_{i,j}^{t+1} = \begin{cases} Q \cdot exp(\dfrac{x_{worst}^{t} - x_{i,j}^{t}}{i^2}) & if \;\; i > \dfrac{N}{2} \\ x_{p}^{t+1} + |x_{i,j}^{t} - x_{p}^{t+1}|(A^T(AA^T)^{-1})L & if \;\; i \leq \dfrac{N}{2}. \end{cases} \tag{2}$$

where, x_{worst}^{t} represents the worst position of the current sparrow population, x_{p}^{t+1} represents the optimal position of the current sparrow population, A is a matrix of size, the element is randomly assigned to 1 or -1. $i > N/2$ means that the ith follower has not found food source and needs to fly to other areas for food, otherwise it is at the optimal position x_{p}^{t+1} for food.

The updated description of the scouter's position is shown in (3):

$$x_{i,j}^{t+1} = \begin{cases} x_{best}^{t} + \beta \cdot |x_{i,j}^{t} - x_{best}^{t}| & if \;\; f_i \neq f_g \\ x_{i,j}^{t} + K \cdot |\dfrac{x_{i,j}^{t} - x_{worst}^{t}}{f_i - f_\omega + \varepsilon}| & if \;\; f_i = f_g. \end{cases} \tag{3}$$

where, x_{best}^{t} represents the global optimal position, β is the normal distribution random number with mean value of 0 and variance of 1, it represents the step size control factor, K is the random number between $[-1, 1]$, f_i represents the current sparrow fitness, f_g represents the global optimal fitness, f_ω represents the global worst fitness, ε is a minimum constant, preventing the denominator from being 0. $f_i \neq f_g$ indicates that the sparrows at the edge of the population are in danger and begin to move towards the center, $f_i = f_g$ means that the sparrows in the center of the population are in danger and begin to move closer to the surrounding sparrows.

III. INTRUSION DETECTION

Intrusion detection technology can be divided into host-based intrusion detection technology (HIDS) and network-based intrusion detection technology (NIDS) according to information sources; According to the detection methods, it is mainly divided into anomaly detection and misuse detection. Traditional misuse detection relies on the pattern library. If the attack mode matches the pattern library in the system, it is deemed that an intrusion has occurred. The misuse detection efficiency is high and the false positive rate is low, but the false negative rate is high, and it is difficult to detect unknown attacks. On the contrary, anomaly detection has low false negative rate and high false positive rate. However, in the actual network, the threat of missing reports of intrusion is often large, such as zero day threats can not be identified by misuse detection technology.

Due to the large number of features in the intrusion detection dataset and the existence of redundant features, it not only reduces the accuracy of classification but also increases the calculation time [19]. Feature selection can select the features that are meaningful to the classifier from the high-dimensional features of the original dataset and eliminate those features that are meaningless. The dimension of the original dataset can be reduced through feature selection, so that the effect of the model can be improved, and the speed of model training and prediction can be improved.The ultimate goal is to use as few features as possible to achieve the best performance.

With the continuous development of deep learning technology, it is also widely used in intrusion detection [20]. Different from traditional machine learning algorithms, deep learning technology can automatically learn data features without human intervention, which brings a new idea for processing multi feature intrusion data [21]. By simulating the thinking mode of human brain, neural network has strong self-learning ability, fault tolerance ability and strong nonlinear mapping ability in the learning process. With the continuous optimization and proposal of Convolutional Neural Network(CNN), many scholars have applied it to the field of intrusion detection [22], [23], [24], [25]. In standard convolutional networks, the final output will only use the highest level features extracted. In densely connected convolutional networks(DenseNet), it uses different levels of features and tends to give a smoother decision boundary. The gradient disappearance is reduced, the feature transmission is strengthened, the feature is used more effectively, the number of parameters is reduced to a certain extent, and the over fitting phenomenon is reduced. Applying DenseNet to intrusion

detection can effectively use the feature information between each data and effectively solve the problem of rapid gradient diffusion and weak generalization ability.

IV. THE HYBRID STRATEGY IMPROVED SPARROW SEARCH ALGORITHM

A. THE HYBRID CIRCLE-PIECEWISE MAP

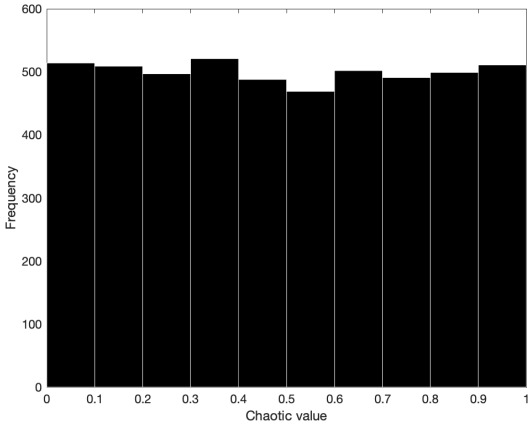
In the initialization stage, sparrows are randomly generated, the population distribution is uneven, and it is easy to gather in a certain area, resulting in small differences between each other's sparrows, which cannot completely cover the entire search area. Adding chaos map initialization can improve the problem. Common chaotic maps include logistic map, tent map, sine map, circle map, etc [26], [27], [28]. Since the logistic map is distributed as chebyshev distribution in the (0, 1) interval, it is densely distributed in the (0.9, 1) interval and unevenly distributed in the entire space. Although the tent map is evenly distributed in the (0, 1) interval, there are small periods and unstable periodic points. Sine map is concentrated in (0, 0.1) and (0.9, 1), and its parameter space in chaotic state is narrow. Circle map is relatively more stable, but mainly distributed in (0.2, 0.5). The paper uses the better ergodicity of piecewise map to combine it with circle map. The formula of CPM is shown in (4), at the bottom of the page, where, β is a control parameter within the range (0, 1), 0.4 is the best choice for this experiment. *rand* is a random number in the (0,1) interval. This experiment sets the dimension of $n = 5000$. The distribution histogram and scatter plot are shown in Figure 1 (a) and Figure 1 (b) respectively.It can be clearly seen from the Figure1 that CPM is evenly distributed.

Introducing CPM into sparrow algorithm can make the population distribution more uniform and increase the diversity of the population. The population initialization method is shown in (5):

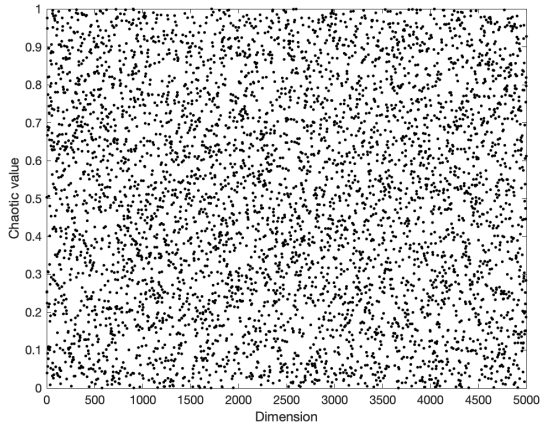
$$x_{i,j} = lb + (ub - lb) \times x(n + 1). \tag{5}$$

Where, *lb* represents the lower bound of population search, and *ub* represents the upper bound of population search. $x_{i,j}$ represents the mapped sparrow. Assuming that the number of sparrow population is 300 and the dimension is 2, the population distribution map and iterative mapping map generated by its random initialization are shown in Figure 2 and Figure 3:

It can be seen that the initial distribution after mapping has a high degree of dispersion of the population, and the number



(a) Distribution histogram



(b) Scatter plot

FIGURE 1. The distribution histogram and scatter plot.

of individuals on the boundary and overlapping individuals is less. It ensures population diversity and reduces the possibility of falling into local optimum.

B. IMPROVEMENTS IN DISCOVERER LOCATION UPDATES

In the SSA, with the increase of the number of iterations, the search scope is decreasing, and the discoverer is getting smaller in each dimension, so it is easy to fall into the local optimum at the initial stage of the search. This paper combines the eagle's moving mode in bald eagle search optimisation algorithm [29], and updates the finder's position

$$x(n+1) = \begin{cases} mod(\frac{x(n)}{\beta} + rand - (\frac{0.25}{\pi})\sin(2\pi x(n)), 1) & if \;\; 0 \le x(n) < \beta \\ mod(\frac{x(n)/\beta}{0.5-\beta} + rand - (\frac{0.25}{\pi})\sin(2\pi x(n)), 1) & if \;\; \beta \le x(n) < 0.5 \\ mod(\frac{(1-x(n))/\beta}{0.5-\beta} + rand - (\frac{0.25}{\pi})\sin(2\pi(1-x(n))), 1) & if \;\; 0.5 \le x(n) < 1-\beta \\ mod(\frac{(1-x(n))}{\beta} + rand - (\frac{0.25}{\pi})\sin(2\pi(1-x(n))), 1) & if \;\; 1-\beta \le x(n) < 1. \end{cases} \tag{4}$$

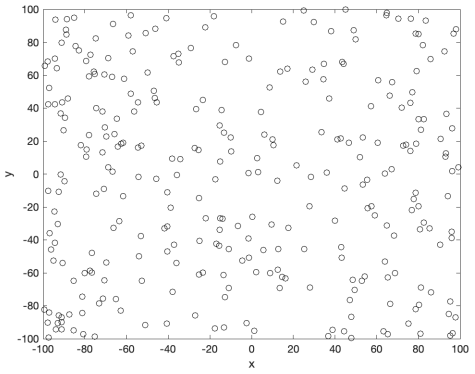


FIGURE 2. Population distribution.

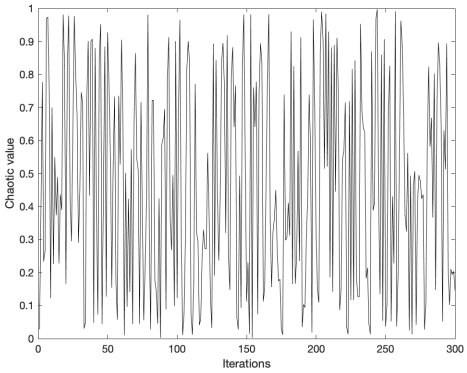


FIGURE 3. Iterative mapping.

by means of spiral flight.As shown in (7):

$$\begin{cases} \theta(i) = x_{best}^{t} + a \cdot \pi \cdot rand \\ r(i) = \theta(i) + R \cdot rand \\ x_r(i) = r(i) \cdot \sin(\theta(i)), y_r(i) = r(i) \cdot \cos(\theta(i)) \\ x(i) = x_r(i)/max(|x_r(i)|), y(i) = y_r(i)/max(|y_r(i)|) \\ \gamma = \coth(2 \cdot (1 - \dfrac{t}{iter_{max}})). \end{cases} \tag{6}$$

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t} \cdot (exp(-\dfrac{i}{\alpha \cdot iter_{max}}))^{\gamma} \cdot x(i) & R_2 < ST \\ x_{i,j}^{t} + L \cdot y(i) & R_2 \geq ST. \end{cases} \tag{7}$$

where, x_{best}^{t} represents the current optimal position, $a$and$R$ are the control parameters of the spiral flight path, the value ranges are (0, 5) and (0.5, 2), *rand* is a random number within (0, 1), $\theta(i)$ and $r(i)$ is the polar angle and polar diameter of the spiral equation, $x(i)$ and $y(i)$ is the position of the sparrow in polar coordinates, both values are$(-1, 1)$. γ is a dynamic adaptive weight, which increases with the number of iterations.

The position update operator of the discoverer is adjusted through continuous iteration. The value is small in the early stage of the search, and the position update is slow, which is convenient for global search. In the later stage, the value becomes larger, the decreasing rate of the operator becomes faster, and the search accuracy is improved. In addition, the spiral movement mode is added to increase the search path of the discoverer at each location update, and improve the global optimization ability of the algorithm.

C. IMPROVEMENTS IN SCOUTER LOCATION UPDATES

In the original SSA, when the scouter senses the danger, the sparrow at the edge of the population starts to move towards the population center, while the sparrow at the population center will move closer to the surrounding sparrows, but this may lead to local optimization. Levy flight is a random walk strategy [30], which generates random step size, can expand the exploration area, improve the diversity of population exploration, and reduce the probability of falling into local optimum.

Therefore, the location update mechanism of the scouter should be improved. The original step size control factor β is a random number with normal distribution, which is unstable and will lead to slow convergence. The use of Levy step size has better ergodicity. If the step size is small, the local optimization ability of the algorithm becomes stronger; if the step size is large, the global search ability of the algorithm becomes stronger. Through the Levy flight mechanism, the change of the position of the sparrow at the edge of the population is more flexible, the movement range of the sparrow at the center of the population is increased, and the probability of falling into the local optimum is further reduced. The improved formula is shown in (8):

$$x_{i,j}^{t+1} = \begin{cases} x_{best}^{t} + lv \cdot |x_{i,j}^{t} - x_{best}^{t}| & if \ f_i \neq f_g \\ x_{i,j}^{t} + lv \cdot |\dfrac{x_{best}^{t} - x_{worst}^{t}}{f_g - f_\omega + \varepsilon}| & if \ f_i = f_g. \end{cases} \tag{8}$$

where, $lv = \frac{c_1}{|c_2|}^{-\zeta}$, c_1 and c_2 are random numbers with normal distribution. $c_1 = N(0, {\sigma_u}^2)$, $c_2 = N(0, 1)$, $\xi = 1.5$, σ_u is shown in (9):

$$\sigma_u = (\frac{\Gamma(1+\xi) \cdot \sin \pi\xi/2}{\Gamma((1+\xi)/2) \cdot \xi \cdot 2^{((\xi-1)/2)}})^{1/\xi} \tag{9}$$

D. OPTIMIZATION OF SIMPLEX METHOD

Simplex method is a direct and fast method to solve the optimal value, which has the advantages of fast convergence and wide application range. The basic idea is to find a point in a search domain and judge whether it is the optimal solution. If not, generate a new solution from the current solution, and then judge. Continue to iterate until the optimal value is found.

Using simplex method to optimize sparrow position can further improve the search ability of sparrow algorithm [31]. First, the sparrow population is sorted according to the fitness value, the optimal position x_{best} and the suboptimal position x_{next} are selected, and the midpoint position x_{medium} is calculated. At the same time, record the fitness value f_{best} and f_{next} of the corresponding position, $x_{medium} = (x_{best} + x_{next})/2$.

Next, record the position x_{worst} of the sparrow with the n worst fitness, and perform reflection operation to make it

move in the opposite direction to increase the search range. The position of the reflection point is x_{reflex}, and the fitness is f_{reflex}, α is the reflection coefficient, which is set as 1.

$$x_{reflex} = x_{medium} + \alpha \cdot (x_{best} - x_{worst}) \tag{10}$$

The fitness value of the reflection point is judged in the following three cases:

1). If $f_{reflex} < f_{best}$, it indicates that the direction is correct, expand the sparrow at this position to search in the opposite direction farther away from the worst position to prevent it from falling into local optimum. Note that the position of sparrow after expansion is x_{expand}, and the fitness is f_{expand}, γ is the expansion coefficient, which is set as 2.

$$x_{expand} = x_{medium} + \gamma \cdot (x_{reflex} - x_{medium}) \tag{11}$$

If $f_{expand} < f_{best}$, it indicates that the expansion is effective, then the expansion point is used to replace the worst point to form a new simplex $x_{worst} = x_{expand}$, otherwise, it means that the expansion is invalid and abandoned x_{expand}, and the reflection point is still used to replace the worst point to form a new simplex $x_{worst} = x_{reflex}$.

2). If $f_{reflex} > f_{worst}$, it shows that the direction of refliection is incorrect, and the sparrow at this position should be contracted to make the sparrow at the worse position closer to the optimal position, so as to enhance the local exploration ability. The position of sparrow after external contraction is x_{ec}, and the fitness is f_{ec}, β is the expansion coefficient, which is set as 0.5.

$$x_{ec} = x_{medium} + \beta \cdot (x_{worst} - x_{medium}) \tag{12}$$

If $f_{ec} < f_{worst}$, at this time, the outer contraction point is used to replace the worst point, forming a new simplex $x_{worst} = x_{ec}$, otherwise, it indicates that the external contraction is invalid, and the reflection point is used to replace the worst point to form a new simplex $x_{worst} = x_{reflex}$.

3). If $f_{best} < f_{reflex} < f_{worst}$, it indicates that the reflection point moves too far at this time, and internal contraction is required. The position of sparrow after internal contraction is x_{ic}, and the fitness is f_{ic}. β is the shrinkage coefficient, which is set as 0.5.

$$x_{ic} = x_{medium} + \beta \cdot (x_{reflex} - x_{medium}) \tag{13}$$

If $f_{ic} < f_{worst}$, at this time, the inner contraction point is used to replace the worst point to form a new simplex $x_{worst} = x_{ic}$, otherwise, it indicates that the internal contraction is invalid, and the reflection point is used to replace the worst point to form a new simplex $x_{worst} = x_{reflex}$.

In this paper, the simplex method is used to improve the individual with poor fitness after each iteration to enhance the ability of the algorithm to jump out of the local optimum.

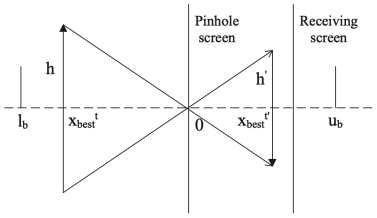


FIGURE 4. Pinhole imaging.

E. OPTIMIZATION OF REVERSE LEARNING STRATEGY FOR PINHOLE IMAGING

In order to solve the problem that most swarm intelligence optimization algorithms are easy to fall into local optimum, some scholars have proposed a reverse learning method. By constructing its inverse solution based on the feasible solution of the current problem, the search scope is expanded to find a better solution. This paper selects a reverse learning strategy for pinhole imaging to obtain the reverse solution of the new current solution [32], increase the range of solution space, and enhance the ability of the algorithm to escape from the local optimal.

The principle of pinhole imaging is shown in the Figure 4:

Suppose that there is a flame with height h in the sparrow population search domain, and its projection x_{best}^t on the x axis is the position of the current optimal solution. The flame passes through the pinhole screen and produces a reflection with height h' on the receiving screen. The projection of this position on the x axis is $x_{best}^{t'}$, and the upper and lower bounds on the x axis correspond to the upper bound u_b and lower bound l_b of the sparrow population search.

According to the principle of keyhole imaging shown in (14), set $\frac{h}{h'} = n$. The position of the sparrow's optimal solution can be obtained through transformation, its expression is shown in (15):

$$\frac{(u_b + l_b)/2 - x_{best}^t}{x_{best}^{t'} - (u_b + l_b)/2} = \frac{h}{h'} \tag{14}$$

$$x_{best}^{t'} = \frac{u_b + l_b}{2} + \frac{u_b + l_b}{2n} - \frac{x_{best}^t}{n} \tag{15}$$

By changing the position of the receiving screen, that is, adjusting the distance between the pinhole screen and the receiving screen, and changing the size of n, a better position solution can be geted. Through the pinhole imaging reverse learning strategy, the sparrow position of the best individual generated after each iteration is updated, which not only expands the search area of the best individual, but also reduces the problem of falling into local optimum.

F. ALGORITHM FLOW AND PSEUDO CODE

The implementation steps and pseudo code of HSISSA are as follows:

Step 1. The CPM is used to initialize the population, set the population number N, the discoverer ratio PD, the scouter ratio SD, the maximum number of iterations $iter_{max}$, the alert

value ST, the search upper bound u_b, the search lower bound l_b, and the dimension dim.

Step 2. Calculating the individual fitness of sparrows and rank them, and recording the sparrow scores of the best position x_{best}, the second best position x_{next} and the worst positionx_{worst}. Their fitness are $f_{best}, f_{next}, f_{worst}$.

Step 3. Select sparrows with population number $N \cdot PD$ as the discoverer, and update them according to (7).

Step 4. Select other sparrows as followers and update according to (2).

Step 5. Select sparrows with population number $N \cdot SD$ as the scouter, and update them according to (8).

Step 6. The simplex method is used for optimization. First, record the midpoint position x_{medium} and fitness value of the best sparrow and the second best sparrow, select n sparrows with poor fitness, perform reflex operation according to (10), and record the fitness f_{reflex}.

Step 7. Compare the size of f_{best}, f_{next}, and f_{worst}, if $f_{reflex} < f_{best}$, expand according to (11), if $f_{reflex} > f_{worst}$, perform external contraction according to (12), if $f_{best} < f_{reflex} < f_{worst}$, perform external contraction according to (13).

Step 8. Record the current optimal individual position, and carry out reverse learning of pinhole imaging according to (15).

Step 9. Calculate the updated fitness value and replace it if it is better than the current position.

Step 10. Judge whether the maximum number of iteration cycles is reached. If yes, go to the next step, output the optimal result, and the algorithm ends, otherwise jump to Step 2.

Algorithm 1 HSISSA

Input: Set the population number N, the discoverer ratio PD, the scouter ratio SD, the maximum number of iterations $iter_{max}$, the alert value ST, the search upper bound u_b, the search lower bound l_b, and the dimension dim

Output: Optimal sparrow position X_{best}. Optimal fitness value f_{best}

for i=1:N **do**
  Initialize the population with CPM
end for
t=1;
while t<$iter_{max}$ **do**
  Calculate the fitness value $f_{best}, f_{next}, f_{worst}$ and its corresponding sparrow position $x_{best}, x_{nest}, x_{worst}$
  for i=1:PD **do**
    Update the position of discoverer according to (7)
  end for
  for i=(PD+1):N **do**
    Update the position of follower according to (2)
  end for
  for l=1:SD **do**
    Update the position of scouter according to (8)
  end for
  for l=1:N **do**
    Perform reflection operation according to (10), record the fitness f_{reflex}
    if f_{reflex}<f_{best} **then**
      Expansion according to (11)
    end if
    if f_{reflex}>f_{worst} **then**
      External contraction according to (12)
    end if
    if f_{best}<f_{reflex}<f_{worst} **then**
      External contraction according to (13)
    end if
  end for
  for l=1:N **do**
    Carry out reverse learning of pinhole imaging according to (15)
  end for
  for l=1:N **do**
    Calculate the new fitness value and judge whether to replace it
  end for
  t=t+1
end while
return x_{best}, f_{best}

G. COMPLEXITY ANALYSIS

Time complexity is one of the important indicators for judging algorithm performance and computing cost. The time complexity of SSA is $O(f(d) \times d)$, where, $f(d)$, d represents the time required to find the fitness function and the data dimension. The time complexity of HSISSA is analyzed as follows:

(1) Suppose the population number of sparrow is N, the time of population initialization parameter is t_1, the time for generating chaotic values of CPM in each dimension is t_2, and the time complexity is $O_1 = O(t_1 + N \times (f(d) + t_2 \times d))$.

(2) The proportion of discoverer is PD, the time required to update the position of each dimension is t_3, the time complexity of the discoverer updating method with the spiral factor is $O_2 = O(PD \times N \times t_3 \times d)$.

(3) The position update formula of the improved algorithm follower has not changed, so the time complexity is O_3, same as the original algorithm.

(4) The proportion of scouter is SD, the time required to update the position of each dimension is t_4, the time complexity of the scouter updating method with Levy operator is $O_4 = O(SD \times N \times t_4 \times d)$.

(5) Suppose that the time required for reflection, expansion, external contraction and internal contraction of each dimension is t_5, t_6, t_7, t_8, the time complexity after simplex optimization is $O_5 = O(N \times (t_5 + t_6 + t_7 + t_8) \times d)$.

(6) Suppose that the time required to generate the reverse solution for each dimension is t_9, the time complexity optimized by the pinhole imaging mechanism is $O_6 = O(N \times t_9 \times d)$.

Therefore, the total time complexity of HSISSA is $O' = O_1 + iter_max \times (O_2 + O_3 + O_4 + O_5 + O_6) =$

TABLE 1. Algorithm parameter setting.

Algorithms	Parameters	Values
GWO	Convergence factor a	[0,2]
WOA	Spiral shape parameter b	1
	Convergence factor a	[0,2]
SCA	Convergence factor a	2
PSO	Inertia weight w	0.9
	Individual learning factor c_1	1.5
	Group learning factor c_2	2
	Speed range	[2,2]
SSA	The alert value ST	0.8
	The discoverer ratio PD	0.7
	The alerter ratio SD	0.2
HSISSA	The alert value ST	0.8
	The discoverer ratio PD	0.7
	The alerter ratio SD	0.2
	Number of poor individuals n	10

$O(f(d) \times d)$. HSISSA and SSA have the same time complexity, the performance improvement did not sacrifice time.

V. ALGORITHM PERFORMANCE TESTING

A. EXPERIMENTAL ENVIRONMENT

In order to ensure the fairness and preciseness of the experimental results, all tests in this paper are conducted in the same environment. The processor is 2.0GHz quad-core 10th-generation Intel Core i5, configured with 16GB of 3733MHz LPDDR4X onboard memory and the operating system is Mac OS 12.0.1. The algorithm verification experiment is carried out in Matlab2020a environment. The subsequent intrusion detection experiments were run in the python 3.9+tensorflow 2.6 environment.

B. PARAMETER SETTING

In order to verify the performance of HSISSA, this paper selects the gray wolf optimization algorithm (GWO), whale optimization algorithm (WOA), sine cosine optimization algorithm (SCA), particle swarm optimization (PSO) and sparrow search algorithm (SSA) for comparative experiments. The common parameters of each algorithm are population size N and maximum number of iterations $iter_{max}$. Set $N = 30$, $iter_{max} = 500$, and each algorithm runs 30 times respectively. The parameters of each algorithm are shown in Table 1:

C. INTRODUCTION TO BENCHMARKING FUNCTIONS

In order to verify the performance of the improved algorithm, this paper selects 10 benchmark test functions for experiments, including 6 single peak test functions and 4 multi peak test functions. As shown in Table 2, where $f_1 \sim f_6$ is unimodal function and $f_7 \sim f_{10}$ is multimodal function.

It can be seen from Table 3 and Table 5, for unimodal functions, f_1, f_2, f_3, f_4, and multimodal functions f_7, f_8, f_9, HSISSA can reach the theoretical optimal solution 0 in both 30 and 200 dimensions. For f_1, f_2, f_3, f_4, and f_8, although SSA can reach the theoretical optimal solution, it is not stable. HSISSA greatly improves the stability while maintaining the performance of the original algorithm. For the solution of f_5, SSA has a higher precision than other algorithms, and the result of HSISSA has been improved by one order of magnitude. For the solution of f_6, the precision of the HSISSA and SSA algorithms is approximately the same. For f_7 and f_9, SSA achieves the theoretical optimal solution, while HSISSA only improves the convergence speed. For the solution of f_{10}, WOA, SSA and HSISSA can reach the theoretical optimal solution, but WOA is less stable than the latter two.

It can be seen from the analysis that the solution accuracy of low dimensional functions of each algorithm is higher than that of high-dimensional functions. HSISSA is superior to other algorithms in terms of convergence speed, convergence accuracy and stability, and can still converge to the optimal solution quickly after the dimension becomes higher. Therefore, the HSISSA proposed in this paper has very obvious advantages.

D. CONVERGENCE CURVE ANALYSIS

For the 30 dimensional benchmark function, the average convergence curve of each algorithm running independently for 30 times is shown in Figure 5. For most functions, HSISSA is superior to other algorithms. It only has no significant improvement compared with SSA when solving function f_6. HSISSA has faster convergence speed and is easier to jump out of the local optimal range.

E. WILCOXON RANK SUM TEST

According to literature [33], the comparison and analysis of the algorithm only based on the average value and standard deviation can not fully explain the performance of the improved algorithm, and it needs to be verified according to the statistical test. In this paper, Wilcoxon rank sum test is used to independently analyze each calculation result, and compare the difference with other algorithms at $P = 5\%$ significance level. When $P < 5\%$, the hypothesis is rejected, which indicates that there is obvious difference between the two algorithms. When $P > 5\%$, the hypothesis is reveived, which indicates that the two algorithms have similar optimization capabilities.

Table 6 shows the P-value of rank sum test between HSISSA and other algorithms under 10 benchmark test functions. If the two algorithms get the optimal value at the same time and cannot be compared, NaN is used to indicate that it is not applicable, C represents the comparison result, "+" represents that HSISSA is superior to other algorithms, "=" represents that HSISSA is equivalent to other algorithms, and "−" represents that HSISSA is inferior to other algorithms. The results in Table 6 show that for the optimization results of most test functions, HSISSA is superior to other algorithms, only slightly worse than WOA and SSA algorithms when solving f_9 and f_6, and the convergence accuracy of HSISSA is better.

VI. APPLICATION OF INTRUSION DETECTION

A. DATA PREPROCESSING

CIC-IDS2017 dataset is a network intrusion detection dataset designed, collected and processed by Sharafaldin et al. [34].

TABLE 2. Benchmark test functions.

Function Name	Formula	Dimension	Range	Optimal value
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30/200	[-100,100]	0
Schwefel 2.22	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30/200	[-10,10]	0
Schwefel 1.2	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j^2)^2$	30/200	[-100,100]	0
Schwefel 2.21	$f_4(x) = max\{\lvert x_i \rvert, 1 \le i \le n\}$	30/200	[-100,100]	0
Quartic	$f_5(x) = \sum_{i=1}^{n} ix_i^4 + random[0,1)$	30/200	[-1.28,1.28]	0
Rosenbrock	$f_6(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30/200	[-30,30]	0
Rastrigin	$f_7(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30/200	[-5.12,5.12]	0
Alpine	$f_8(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	30/200	[-10,10]	0
Griewank	$f_9(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30/200	[-600,600]	0
Ackley	$f_{10}(x) = -20exp(-0.2\sqrt{\sum_{i=1}^{n} \frac{x_i^2}{n}}) - exp(\sqrt{\sum_{i=1}^{n} \frac{\cos(2\pi x_i)}{n}}) + 20 + e$	30/200	[-32,32]	0

TABLE 3. Comparison of benchmark function optimization results/30 dimensions.

Function	Running results	SCA	GWO	WOA	PSO	SSA	HSISSA
f_1	Optimal value	1.97E-03	2.82E-32	1.92E-85	5.37E+00	0.00E+00	0.00E+00
	Worst value	5.10E+02	9.98E-30	8.01E-72	2.12E+01	7.28E-69	0.00E+00
	average value	2.39E+01	1.35E-30	4.36E-73	1.16E+01	2.45E-70	0.00E+00
	standard deviation	9.24E+01	2.12E-30	1.56E-72	3.86E+00	1.39E-69	0.00E+00
f_2	Optimal value	1.68E-05	2.69E-19	4.23E-58	5.68E+00	0.00E+00	0.00E+00
	Worst value	2.45E-01	5.08E-18	2.28E-49	1.79E+01	1.52E-30	0.00E+00
	average value	2.74E-02	1.79E-18	1.01E-50	1.13E+01	5.07E-32	0.00E+00
	standard deviation	6.02E-02	1.20E-18	4.16E-50	3.37E+00	2.78E-31	0.00E+00
f_3	Optimal value	1.40E+03	4.41E-05	2.03E+04	2.77E+02	0.00E+00	0.00E+00
	Worst value	2.60E+04	1.19E-01	8.36E+04	1.63E+03	5.71E-48	0.00E+00
	average value	9.88E+03	8.11E-03	4.49E+04	6.46E+02	1.90E-50	0.00E+00
	standard deviation	6.39E+03	2.20E-02	1.40E+04	3.29E+02	1.04E-49	0.00E+00
f_4	Optimal value	8.01E+00	4.71E-06	1.00E-01	3.48E+00	0.00E+00	0.00E+00
	Worst value	6.48E+01	2.95E-04	8.98E+01	1.13E+01	1.49E-30	0.00E+00
	average value	3.75E+01	4.13E-05	5.70E+01	6.86E+00	4.97E-32	0.00E+00
	standard deviation	1.34E+01	5.49E-05	3.00E+01	1.82E+00	2.73E-31	0.00E+00
f_5	Optimal value	9.29E-03	8.37E-04	1.73E-05	5.50E-02	1.31E-05	1.27E-06
	Worst value	5.18E-01	6.31E-03	2.82E-02	1.63E+01	3.08E-03	2.42E-04
	average value	1.22E-01	3.55E-03	5.16E-03	1.22E+00	6.39E-04	6.85E-05
	standard deviation	1.28E-01	1.45E-03	7.63E-03	3.02E+00	6.71E-04	5.47E-05
f_6	Optimal value	3.75E+01	2.58E+01	2.71E+01	3.80E+02	2.40E-07	3.75E-07
	Worst value	1.94E+05	2.72E+01	2.88E+01	6.60E+03	8.30E-04	6.54E-03
	average value	2.43E+04	2.66E+01	2.80E+01	2.07E+03	1.45E-04	7.32E-04
	standard deviation	4.62E+04	4.53E-01	4.53E-01	1.61E+03	4.86E-04	1.42E-03
f_7	Optimal value	1.02E-02	0.00E+00	0.00E+00	6.72E+01	0.00E+00	0.00E+00
	Worst value	1.02E+02	3.76E+01	5.68E-14	1.58E+02	0.00E+00	0.00E+00
	average value	2.94E+01	1.91E+01	1.89E-15	1.05E+02	0.00E+00	0.00E+00
	standard deviation	2.82E+01	8.16E+00	1.04E-14	2.26E+01	0.00E+00	0.00E+00
f_8	Optimal value	1.30E-04	8.02E-05	9.13E-59	3.08E+00	0.00E+00	0.00E+00
	Worst value	1.01E-02	4.50E-03	5.45E-30	1.03E+01	3.14E-33	0.00E+00
	average value	2.74E-03	1.50E-03	1.82E-29	7.26E+00	1.17E-34	0.00E+00
	standard deviation	2.45E-03	1.03E-03	9.96E-31	1.99E+00	5.74E-34	0.00E+00
f_9	Optimal value	1.06E-03	0.00E+00	0.00E+00	5.27E+00	0.00E+00	0.00E+00
	Worst value	1.34E+01	1.91E-03	1.45E-01	2.23E+01	0.00E+00	0.00E+00
	average value	7.29E-01	4.34E-03	4.82E-03	1.27E+01	0.00E+00	0.00E+00
	standard deviation	3.61E-01	9.60E-02	2.64E-02	4.52E+00	0.00E+00	0.00E+00
f_{10}	Optimal value	3.21E-01	1.51E-14	8.88E-16	3.92E+00	8.88E-16	8.88E-16
	Worst value	2.03E+01	3.29E-14	7.99E-15	8.56E+00	8.88E-16	8.88E-16
	average value	1.67E+01	2.10E-14	5.63E-15	5.50E+00	8.88E-16	8.88E-16
	standard deviation	6.53E+00	5.14E-15	2.53E-15	1.10E+00	0.00E+00	0.00E+00

TABLE 4. Data distribution.

Label		Quantity
BENIGN		2272688
DoS	DoS Hulk	230124
	DoS GoldenEye	10293
	DoS slowloris	5796
	DoS Slowhttptest	5499
PortScan		158930
DDos		128025
Brute Force Attack	FTP-Patator	7938
	SSH-Patator	5897
Bot		1966
Web Attack	Web Attack Brute Force	1507
	Web Attack XSS	652
	Web Attack Sql Injection	21
Infiltration		36
Heartbleed		11

of the Canadian Security Research Institute in 2017. Compared with other intrusion datasets, it has richer data categories. It includes normal network traffic data and 7 types of attacks. Each type of attack contains a variety of different types, totaling 14 types of attack data.

The original file contains five days of data traffic from Monday to Friday, divided into eight files. First, it is merged into one file. There are 2830743 traffic data in the file. Each data contains 77 features and one label column.

The data preprocessing process is as follows:

(1) NaN and Infinity exist in some characteristic columns, which will lead to model training errors. After statistics, it is found that the number of samples containing these values is very small, and Most of the useless data are in BENIGN, which accounts for the largest proportion of data. Therefore, it is directly deleted for data cleaning, and the processed data size is 2827876.

(2) In order to facilitate model classification, all Dos attacks, Web attacks and violent attacks are classified into one category, digitally code all labels from 0 to 9. Table 4 shows the distribution of processed data.

(3) Because the data difference between different features is too large, it is easy to cause some sample points in the feature space to be affected by larger eigenvalues. The characteristic values are normalized to the range of [0, 1] according to (16).

$$x_i' = \frac{x_i - x_{min}}{x_{max} - x_{min}}. \tag{16}$$

where, x_i' represents the normalized value, x_i represents the initial value, x_{min} represents the minimum value of the feature, and x_{max} represents the maximum value of the feature.

(4) Due to the imbalance of the data set, the number of normal data has reached more than 2 million, and the minimum attack type is only 11. In direct training, it is easy to over fit most class samples and ignore the features of a few class samples, which leads to poor performance of the classifier. In this experiment, SMOTE oversampling and random undersampling were used to synthesize a small number of samples. Finally, 50000 pieces of data were selected for the experiment.

In order to illustrate the applicability of HSISSA algorithm, the UNSW-NB15 dataset is additionally selected for verification.

The dataset includes 9 attack categories: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms, and contains a total of 47 features excluding the labeled columns.

The data preprocessing process is as follows:

(1) In order to prevent training errors, delete the data with the characteristic column "service" as "−", and the processed data size is 81173.

(2) The three columns of character-type features of "proto", "service" and "state" are converted into numeric features using one-hot coding. The processed dataset has 58 features in total.

(3) According to (16), the characteristic values are normalized to the range of [0, 1].

B. EVALUATION CRITERIA

These indicators are used to evaluate the performance of intrusion detection systems, either machine learning or deep learning-based [35].

(1) TN, the number of normal network traffic data correctly detected by the model as normal data.

(2) TP, the attacked network traffic data is correctly detected by the model as the number of attack data.

(3) FN, the number of attacked network traffic data that are detected as normal data by model error.

(4) FP, the number of normal network traffic data detected as attack data by model error.

Accuracy, proportion of correctly predicted network traffic data in all predicted network traffic data. The calculation process is shown in (17).

$$acc = \frac{TP + TN}{TP + TN + FN + FP}. \tag{17}$$

Recall rate, it reflects the proportion of network traffic data of a certain category that is correctly predicted. The calculation process is shown in (18).

$$recall = \frac{TP}{TP + FN}. \tag{18}$$

Precise rate, it reflects the proportion of correctly predicted network traffic data of a certain category in the predicted data of this category. The calculation process is shown in (19).

$$precision = \frac{TP}{TP + FP}. \tag{19}$$

F1 score, which is the harmonic average of recall rate and accuracy rate, it s a measure of a test's accuracy, the calculation process is shown in (20).

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}. \tag{20}$$

TABLE 5. Comparison of benchmark function optimization results/200 dimensions.

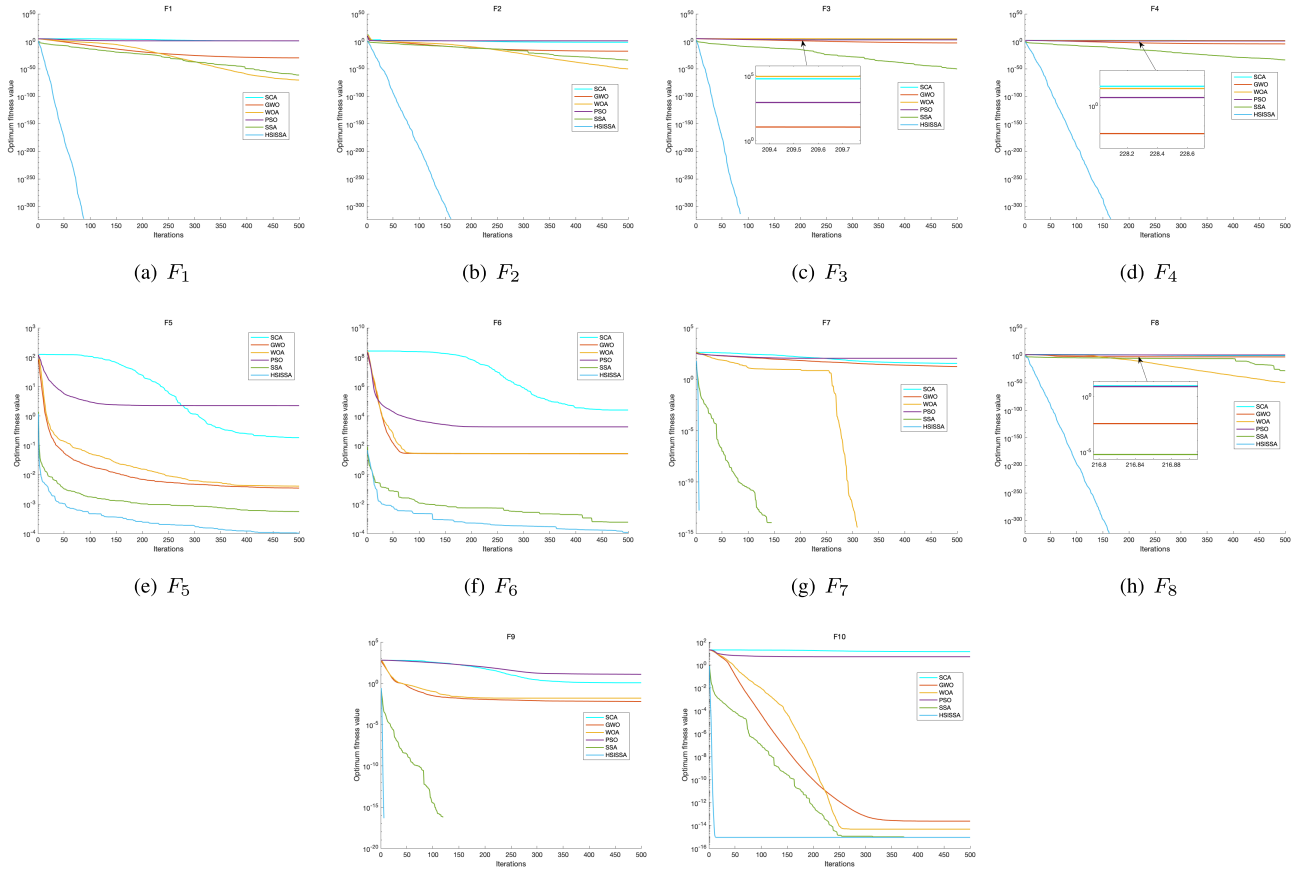
Function	Running results	SCA	GWO	WOA	PSO	SSA	HSISSA
f_1	Optimal value	1.37E+04	1.85E-08	3.77E-81	4.10E+03	0.00E+00	0.00E+00
	Worst value	8.96E+04	9.45E-08	7.68E-70	6.59E+03	5.22E-53	0.00E+00
	average value	5.09E+04	4.97E-08	3.47E-71	5.13E+03	1.74E-54	0.00E+00
	standard deviation	1.92E+04	2.17E-08	1.47E-70	6.06E+02	9.53E-54	0.00E+00
f_2	Optimal value	6.82E+00	5.84E-06	2.64E-55	2.40E+02	0.00E+00	0.00E+00
	Worst value	7.51E+01	1.75E-05	5.31E-48	4.22E+02	5.70E-33	0.00E+00
	average value	3.13E+01	1.05E-05	3.00E-49	3.11E+02	2.41E-34	0.00E+00
	standard deviation	1.80E+01	2.91E-06	1.02E-48	4.63E+01	1.06E-33	0.00E+00
f_3	Optimal value	6.36E+05	6.04E+04	2.45E+06	8.16E+04	0.00E+00	0.00E+00
	Worst value	1.80E+06	1.39E+05	9.72E+06	4.17E+05	1.21E-44	0.00E+00
	average value	1.02E+06	9.67E+04	5.07E+06	1.43E+05	4.15E-46	0.00E+00
	standard deviation	3.00E+05	1.79E+04	1.96E+06	6.39E+04	2.21E+45	0.00E+00
f_4	Optimal value	9.28E+01	5.40E+01	5.45E+01	1.62E+01	0.00E+00	0.00E+00
	Worst value	9.79E+01	4.39E+01	9.88E+01	2.14E+01	1.03E-35	0.00E+00
	average value	9.64E+01	5.36E+01	8.16E+01	1.84E+01	3.64E-37	0.00E+00
	standard deviation	1.20E+01	5.49E-05	2.62E+01	1.40E+00	1.87E-36	0.00E+00
f_5	Optimal value	5.17E+02	2.05E-02	9.92E-05	2.90E+01	3.20E-06	3.43E-06
	Worst value	3.00E+03	5.19E-02	2.56E-02	9.71E+02	8.24E-03	3.49E-04
	average value	1.50E+03	3.31E-02	5.61E-03	4.05E+02	1.73E-04	8.06E-05
	standard deviation	5.00E+02	8.48E-03	5.67E-03	3.04E+02	2.08E-04	8.27E-05
f_6	Optimal value	2.51E+08	1.97E+02	1.97E+02	1.51E+06	1.27E-07	1.64E-08
	Worst value	8.82E+08	1.98E+02	1.98E+02	5.01E+06	3.55E-02	2.68E-02
	average value	5.53E+08	1.98E+02	1.98E+02	2.64E+06	3.74E-03	1.70E-04
	standard deviation	1.74E+08	4.56E-01	1.31E-01	7.85E+05	9.13E-03	5.02E-04
f_7	Optimal value	2.07E+02	5.18E+01	0.00E+00	1.61E+03	0.00E+00	0.00E+00
	Worst value	9.66E+02	2.42E+02	0.00E+00	1.92E+03	0.00E+00	0.00E+00
	average value	5.51E+02	1.16E+02	0.00E+00	1.76E+03	0.00E+00	0.00E+00
	standard deviation	2.30E+02	4.76E+01	0.00E+00	8.28E+01	0.00E+00	0.00E+00
f_8	Optimal value	2.08E+01	1.41E-02	8.86E-58	1.42E+02	0.00E+00	0.00E+00
	Worst value	1.02E+02	4.35E-01	1.11E-50	1.79E+02	5.51E-24	0.00E+00
	average value	5.56E+01	4.12E-02	1.16E-51	1.61E+02	1.84E-26	0.00E+00
	standard deviation	1.81E+01	7.48E-02	2.82E-51	1.03E+01	1.01E-25	0.00E+00
f_9	Optimal value	1.18E+02	3.37E-09	0.00E+00	1.02E+02	0.00E+00	0.00E+00
	Worst value	1.11E+03	6.44E-02	1.11E-16	1.55E+02	0.00E+00	0.00E+00
	average value	4.61E+02	5.22E-03	3.70E-18	1.26E+02	0.00E+00	0.00E+00
	standard deviation	2.55E+02	1.62E-02	2.03E-17	1.35E+01	0.00E+00	0.00E+00
f_{10}	Optimal value	1.10E+01	7.87E-06	8.88E-16	1.08E+01	8.88E-16	8.88E-16
	Worst value	2.08E+01	2.77E-05	7.99E-15	1.21E+01	8.88E-16	8.88E-16
	average value	2.01E+01	1.44E-05	5.03E-15	1.15E+01	8.88E-16	8.88E-16
	standard deviation	2.34E+01	4.36E-06	2.30E-15	3.29E-01	0.00E+00	0.00E+00

TABLE 6. Wilcoxon rank sum test.

function	SCA		GWO		WOA		PSO		SSA	
	P	C	P	C	P	C	P	C	P	C
f_1	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+
f_2	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	4.5736E-12	+
f_3	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.6572E-11	+
f_4	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	4.5736E-12	+
f_5	3.0199E-11	+	3.0199E-11	+	5.4941E-11	+	3.0199E-11	+	2.1959E-07	+
f_6	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	1.4128E-01	-
f_7	1.2118E-12	+	4.5736E-12	+	3.3371E-01	=	1.2118E-12	+	NaN	=
f_8	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	5.772E-11	+
f_9	1.2118E-12	+	1.4552E-04	+	3.3371E-01	-	1.2118E-12	+	NaN	=
f_{10}	1.2118E-12	+	7.7774E-13	+	8.0668E-08	+	1.2118E-12	+	NaN	=

The above evaluation are biased and should not be used without clear understanding of the biases, and corresponding identification of chance or base case levels of the statistic [36]. This paper will further analyze the indicators of ROC, Informedness, Markedness and Correlation.

According to the prediction result of the learner, the threshold is changed from 0 to the maximum, that is, each sample is predicted as a positive sample at the beginning. With the increase of the threshold, the number of positive samples predicted by the learner is less and less, until no sample is positive at the end. In this process, the values of two important quantities, namely "True Positive Rate" and "False Positive Rate", are calculated each time. They are used as abscissa and ordinate for drawing, which is the ROC curve. TPR is the same as the recall rate.



(a) F_1 (b) F_2 (c) F_3 (d) F_4

(e) F_5 (f) F_6 (g) F_7 (h) F_8

(i) F_9 (j) F_{10}

FIGURE 5. Convergence curve.

FPR, it is the proportion of all inverse classes that are predicted to be positive. The calculation process is shown in (21).

$$FPR = \frac{FP}{FP + TN}. \tag{21}$$

Informedness measures the degree to which a predictor has knowledge about a particular condition, and indicates the likelihood that a prediction is accurately informed with respect to that condition. The calculation process is shown in (22).

$$Informedness = \frac{TP}{TP + FN} + \frac{TN}{FP + TN} - 1. \tag{22}$$

Markedness measures the degree to which a condition stands out for the given predictor, and indicates the likelihood that the predictor identifies the condition as exceptional. The calculation process is shown in (23).

$$Markedness = \frac{TP}{TP + FP} + \frac{TN}{TN + FN} - 1. \tag{23}$$

Correlation describes the correlation coefficient between actual classification and predicted classification, which is often regarded as a balancing measure. The calculation process is shown in (24).

$$Correlation = \sqrt{Informedness \cdot Markedness}. \tag{24}$$

C. FEATURE SELECTION

In the feature selection method of HSISSA, sparrow population initialization position corresponds to a random feature, sparrow individuals search randomly in the feature space, and evaluate the feature subset obtained. If the value of the original position is greater than 0.5, select the feature, which is represented as "1", otherwise, the feature will be discarded, which means "0". If all values are less than 0.5, the sparrow position with the highest value is selected. The number of features is combined with the F1 score as the fitness function, and the number of features is minimized under the condition that the F1 score is high, the expression is as shown in (25). Finally, the selected feature subset is classified using CatBoost to obtain the final classification accuracy, false positive rate, recall rate and F1 score.

$$fitness = 1 - \sum_{i=1}^{n} F_1 + \alpha \times features. \tag{25}$$

TABLE 7. Parameter setting.

iterations	200
max_depth	4
random_state	666
learning_rate	0.03

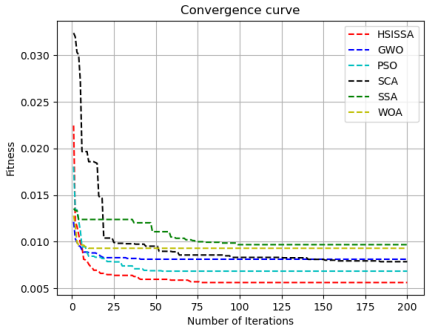


FIGURE 6. Convergence comparison of six algorithms.

TABLE 8. Accuracy of 6 algorithms.

SCA	GWO	WOA	PSO	SSA	HSISSA
94.67	96.33	96.14	95.23	97.69	99.14

where, α is a random number in the [0, 1] interval. It is verified by many experiments that the best effect can be achieved by setting it to 0.001. *features* is length of feature.

This experiment uses CatBoost algorithm for classification. It is the third improved algorithm based on GBDT after XGBoost and LightGBM. It can efficiently and reasonably handle category features, make use of the relationship between features, and use symmetric tree structure to effectively reduce the over fitting phenomenon, with good robustness [37]. Parameter settings are shown in Table 7.

Taking the classification accuracy as the fitness function, the convergence curves of the six algorithms are shown in Figure 6.

The corresponding accuracy of the six algorithms is shown in Table 8.

For the classification results of each type of attack, this paper mainly selects SSA and HSISSA for comparison. The feature selection results are shown in Table 9 and Table 10.

It can be seen from Figure 6 that when the number of iterations reaches 50, HSISSA achieves a better fitness solution, converges faster than other algorithms, and can achieve a better solution. As the number of iterations increases, all algorithms converge to the optimal solution step by step. It can be seen from Table 6 that the accuracy of feature selection results of HSISSA can reach 99.14%, which is higher than other 6 algorithms.

It can be seen from Table 9 and Table 10 that for CIC-IDS2017 dataset SSA retains an average of 13.3 features, with an average accuracy of 97.97%, average F1 score is 97.14%.

BN
ReLU
Conv 1*1,128
BN
ReLU
Conv 3*3,32

FIGURE 7. Dense block.

The HSISSA retains 7.6 features on average, with an accuracy rate of 99.5% and a F1 score of 98.48%. Compared with the original algorithm, it reduces 5.7 features, improves the accuracy rate of 1.53% and the F1 score of 1.34%. The selection results of UNSW-NB15 dataset are shown in Table 11 and Table 12. The SSA retains an average of 15.7 features, reaching an accuracy rate of 93.56%. The HSISSA retains an average of 10.1 features, reaching an accuracy rate of 96.01%, reducing 5.6 features, and improving the accuracy rate of 2.45%. HSISSA has achieved remarkable results on both datasets, so the improved algorithm has good adaptability.

Based on the above indicators, the improved algorithm can obtain better objective function value in the model, accelerate the classification efficiency, and further improve the accuracy while reducing feature dimensions.

D. OPTIMIZATION OF DenseNet DETECTION MODEL

In machine learning or deep learning algorithms, many parameters are set manually, and need to be trained many times to achieve the best results. In this paper, HSISSA is used to optimize the parameters, and the best model is obtained by virtue of its better optimization ability.

In order to adapt the data to the DenseNet model. First, take out the feature column and label column of the dataset separately, convert all features into two-dimensional data, and encode the labels with one-hot. Then, divide the data set into training set and test set according to 8:2, and then divide the training set into training set and verification set according to 8:1.

This paper establishes a DenseNet model for optimizing parameters which based on DenseNet-121 model framework. Specifically, it includes the number of bottleneck layers in DenseNet dense blocks, the compression coefficient in the transition layer, learning rate, batchsize, and epoch. First, set the optimization range, that is, the sparrow search boundary range: $l_b = [1, 1, 1, 1, 0, 1e-3, 128, 1]$, $u_b = [33, 33, 33, 33, 1, 1e-1, 512, 100]$. Secondly, set the minimum error rate of verification set during training as er, the number of iterations is n, the fitness function is shown in (26). Finally, use SSA and HSISSA to optimize the network respectively, and compare the performance of the two models.

$$fitness = 1 - \frac{\sum_{i=1}^{n} er}{n}. \tag{26}$$

TABLE 9. SSA feature selection results for CIC-IDS2017.

Label	index	Acc(%)	Fprate(%)	Recall(%)	F1score(%)
Benign	[15, (4, 5, 7, 14, 21, 28, 33, 34, 36, 42, 49, 50, 63, 65, 76)]	97.81	2.01	97.61	97.13
DoS	[13, (1, 7, 8, 13, 25, 29, 47, 53, 63, 65, 71, 72, 73)]	98.33	1.31	98.26	97.87
DDos	[12, (6, 7, 8, 17, 34, 38, 42, 44, 46, 67, 70, 71)]	96.12	3.87	96.13	96.11
PortScan	[13, (1, 17, 19, 24, 27, 34, 36, 38, 46, 47, 53, 63, 70)]	98.96	1.11	98.66	98.32
Brute Force Attack	[11, (3, 17, 27, 36, 37, 46, 62, 65, 66, 68, 75)]	97.66	2.32	97.43	97.26
Bot	[14, (0, 1, 2, 14, 19, 27, 34, 41, 44, 52, 65, 67, 70, 73)]	97.22	2.39	97.11	93.61
Web Attack	[19, (3, 8, 11, 15, 18, 23, 25, 29, 34, 37, 41, 44, 50, 53, 63, 66, 69, 70, 75)]	99.11	0.11	98.73	98.53
Infiltration	[16, (3, 8, 11, 17, 23, 24, 33, 34, 35, 36, 42, 62, 67, 71, 72, 73)]	96.77	3.14	96.60	96.33
Heartbleed	[7, (8, 9, 44, 46, 50, 54, 74)]	99.88	0.01	100.00	99.12

TABLE 10. HSISSA feature selection results for CIC-IDS2017.

Label	index	Acc(%)	Fprate(%)	Recall(%)	F1score(%)
Benign	[10, (4, 5, 28, 33, 34, 36, 42, 49, 50, 65)]	98.80	0.99	98.60	98.66
DoS	[6, (12, 18, 20, 23, 39, 65)]	99.63	0.33	99.60	99.07
DDos	[4, (4, 46, 54, 71)]	99.89	0.12	99.90	99.61
PortScan	[3, (15, 21, 41)]	99.63	0.12	99.39	99.32
Brute Force Attack	[10, (3, 17, 36, 37, 46, 62, 65, 66, 68, 75)]	99.68	0.05	99.40	99.10
Bot	[9, (0, 1, 2, 14, 19, 27, 34, 44, 65)]	99.21	0.18	98.60	94.63
Web Attack	[16, (3, 11, 15, 18, 23, 25, 29, 37, 41, 44, 50, 53, 63, 66, 69, 70)]	99.89	0.05	99.83	98.93
Infiltration	[5, (9, 11, 19, 37, 39)]	98.78	0.03	97.60	97.02
Heartbleed	[5, (11, 45, 49, 53, 54)]	100.00	0.00	100.00	100.00

TABLE 11. SSA feature selection results for UNSW-NB15.

Label	index	Acc(%)	Fprate(%)	Recall(%)	F1score(%)
Analysis	[16, (3, 4, 8, 10, 16, 17, 25, 29, 30, 32, 35, 37, 39, 43, 51, 55)]	92.71	2.09	86.52	61.41
Backdoor	[20, (7, 9, 10, 11, 14, 16, 17, 18, 19, 20, 23, 27, 29, 32, 36, 41, 44, 48, 51, 55)]	73.88	1.23	48.00	28.57
DoS	[22, (3, 4, 5, 8, 9, 10, 12, 15, 18, 19, 22, 29, 33, 34, 36, 37, 45, 47, 49, 52, 55, 56)]	95.75	3.29	94.80	81.26
Exploits	[13, (1, 3, 4, 15, 17, 18, 23, 27, 32, 37, 47, 49, 52)]	96.15	4.54	97.86	87.81
Fuzzers	[19, (2, 3, 6, 8, 16, 18, 21, 24, 27, 30, 34, 35, 37, 39, 41, 48, 50, 52, 56)]	97.43	1.49	94.95	77.95
Generic	[4, (3, 11, 20, 36)]	99.49	0.06	99.05	99.48
Normal	[11, (6, 10, 15, 24, 31, 32, 33, 37, 39, 40, 55)]	98.13	2.05	98.32	96.01
Reconnaissance	[15, (4, 11, 14, 17, 19, 20, 23, 24, 25, 29, 40, 45, 50, 55, 56)]	95.64	3.74	91.77	65.59
Worms	[21, (0, 3, 4, 8, 9, 13, 14, 18, 22, 24, 26, 30, 31, 35, 39, 43, 47, 49, 53, 56, 57)]	92.90	1.32	85.96	58.68

TABLE 12. HSISSA feature selection results for UNSW-NB15.

Label	index	Acc(%)	Fprate(%)	Recall(%)	F1score(%)
Analysis	[7, (4, 8, 25, 29, 43, 55, 56)]	98.71	1.12	99.51	75.77
Backdoor	[15, (6, 7, 10, 16, 17, 23, 29, 32, 35, 39, 41, 46, 48, 51, 55)]	86.47	0.94	82.82	35.74
DoS	[19, (3, 4, 5, 9, 10, 15, 18, 19, 20, 29, 34, 37, 45, 47, 49, 50, 52, 55, 56)]	98.49	1.14	98.12	84.36
Exploits	[11, (3, 4, 15, 17, 27, 28, 32, 33, 36, 47, 49)]	96.02	2.25	98.29	87.99
Fuzzers	[10, (2, 3, 5, 6, 16, 24, 30, 37, 48, 56)]	97.48	1.10	95.65	84.03
Generic	[2, (3, 36)]	99.51	0.05	99.07	99.51
Normal	[8, (2, 6, 10, 15, 32, 39, 52, 55)]	98.31	1.53	98.16	96.71
Reconnaissance	[7, (4, 14, 19, 23, 40, 45, 55)]	95.31	2.91	93.52	67.75
Worms	[12, (3, 5, 9, 18, 24, 26, 31, 35, 37, 43, 54, 56)]	93.79	0.13	87.71	62.11

The structure of dense block is shown in the Figure 7 and the model structure optimized by HSISSA is shown in Figure 8.

Set the sparrow population to 10 and the maximum number of iterations to 50. The network parameters and training parameters optimized by SSA and HSISSA are [8, 18, 14, 8, 0.484, 0.033, 312, 73] and [32, 7, 23, 12, 3, 0.351, 0.025, 362] respectively. The SSA-DenseNet model has 8, 18, 14, and 8 bottomlenecks for each dense block, the compression coefficient of the Transition Layer is 0.484, the learning rate is 0.033, the batch size is 312, and the epoch is 73. The HSISSA-DenseNet model has 7, 23, 12, and 3 bottomlenecks for each dense block, the compression coefficient of the Transition Layer is 0.351, the learning rate is 0.025, the batch size is 362, and the epoch is 32. The HSISSA optimized model can reach the optimal solution with half of the iterations of the SSA model, it proves that the convergence performance of HSISSA is better than that of SSA.

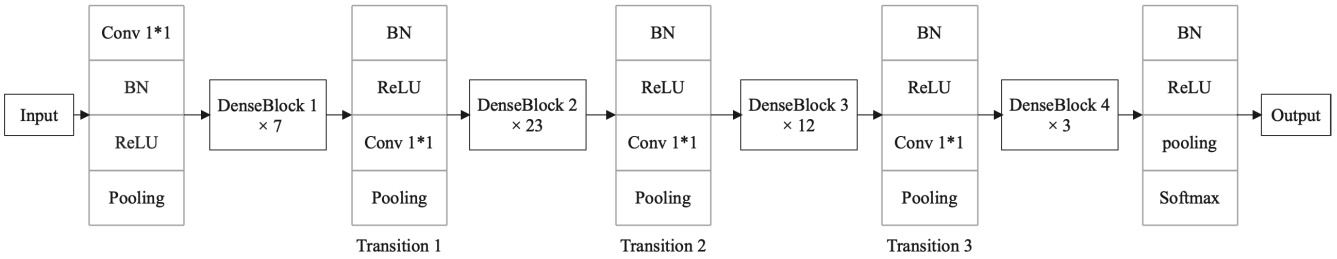


FIGURE 8. HSISSA-Densenet model.

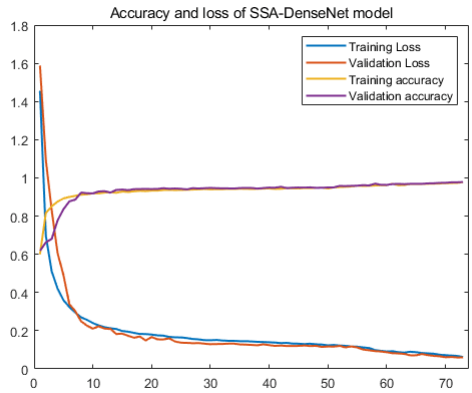


FIGURE 9. Accuracy and loss of SSA-DenseNet model.

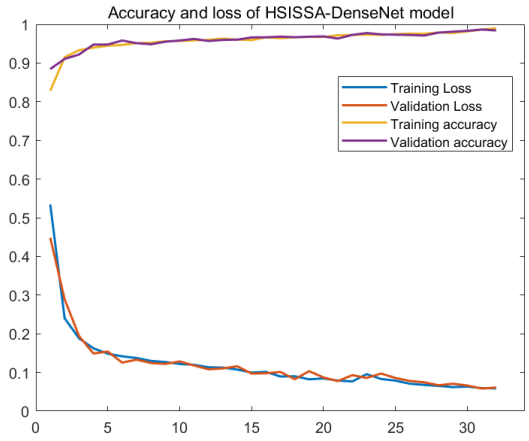


FIGURE 10. Accuracy and loss of HSISSA-DenseNet model.

The loss function and accuracy curve of the training set and verification set of the two algorithms are shown in Figure 9 and Figure 10. The initial loss value of the SSA-DenseNet model is relatively high, and it decreases significantly after about the tenth iteration. The HSISSA-DenseNet model achieves good results after about the fifth iteration. The initial accuracy rate of the HSISSA-DenseNet model is also higher than that of the SSA-DenseNet model. The accuracy rates of the two models are close to 100%, and the comparison effect on the figure is not obvious. Therefore, the test set accuracy rates of the two models are further analyzed according to Table 13 and Table 14.

TABLE 13. SSA-DenseNet model evaluation.

	precision	recall	f1-score	support
BENIGN	0.98	0.96	0.97	4800
DoS	0.96	1.00	0.98	2000
DDoS	1.00	1.00	1.00	1600
PortScan	1.00	1.00	1.00	1520
Brute Force Attack	0.93	0.94	0.94	400
Bot	0.88	0.75	0.81	200
Web Attack	0.89	0.93	0.91	240
Infiltration	0.99	0.97	0.98	100
Heartbleed	0.99	1.00	0.99	88
accuracy			0.98	10498
macro avg	0.96	0.95	0.95	10498
weighted avg	0.98	0.98	0.98	10498

TABLE 14. HSISSA-DenseNet model evaluation.

	precision	recall	f1-score	support
BENIGN	0.99	0.96	0.97	4800
DoS	0.97	0.99	0.98	2000
DDoS	1.00	1.00	1.00	1600
PortScan	1.00	1.00	1.00	1520
Brute Force Attack	0.97	0.94	0.95	400
Bot	0.96	0.94	0.95	200
Web Attack	0.98	0.97	0.97	240
Infiltration	1.00	1.00	1.00	100
Heartbleed	1.00	1.00	1.00	88
accuracy			0.99	10498
macro avg	0.99	0.98	0.98	10498
weighted avg	0.99	0.99	0.99	10498

It can be seen from the above table, for BENGING, DoS, Infiltration, and Heartbleed, the HSISSA-DenseNet model is one percentage point higher than the SSA-DenseNet model. For Brute Force Attack, Bot, and Web Attack, the HSISSA-DenseNet model is 4%, 8%, and 9% higher respectively. The overall accuracy rate increased by one percentage point. For the macro coverage of the multi classification problem, the HSISSA-DenseNet model is 3% higher than the SSA-DenseNet model, it can better deal with complex and changing network attack environments.

Based on the above analysis, for the model optimization of intrusion detection, HSISSA can not only achieve faster convergence, but also greatly improve the performance of the model, which further proves the effectiveness and practicability of HSISSA.

TABLE 15. SSA-RF model evaluation.

	precision	recall	f1-score	support
BENIGN	0.92	0.96	0.94	4800
DoS	0.94	0.98	0.96	2000
DDoS	0.99	1.00	0.99	1600
PortScan	1.00	0.99	0.99	1520
Brute Force Attack	0.89	0.77	0.82	400
Bot	0.82	0.58	0.68	200
Web Attack	0.91	0.83	0.87	240
Infiltration	0.97	0.83	0.89	100
Heartbleed	0.97	1.00	0.98	88
accuracy			0.93	10498
macro avg	0.93	0.88	0.90	10498
weighted avg	0.98	0.98	0.98	10498

TABLE 16. HSISSA-RF model evaluation.

	precision	recall	f1-score	support
BENIGN	0.96	0.97	0.97	4800
DoS	0.94	0.99	0.96	2000
DDoS	1.00	1.00	1.00	1600
PortScan	1.00	1.00	1.00	1520
Brute Force Attack	0.96	0.92	0.94	400
Bot	0.95	0.82	0.88	200
Web Attack	0.96	0.94	0.95	240
Infiltration	0.99	0.95	0.97	100
Heartbleed	1.00	1.00	1.00	88
accuracy			0.97	10498
macro avg	0.97	0.95	0.96	10498
weighted avg	0.99	0.99	0.99	10498

TABLE 17. Comparison results of all models.

model\(%)	**acc**	**precision**	**recall**	**f1score**
DenseNet-121	91.12	92.43	90.66	91.54
RF	88.89	86.78	83.21	84.96
SSA-DenseNet	98.04	95.78	94.98	95.33
HSISSA-DenseNet	99.34	98.56	97.78	98.11
SSA-RF	92.91	93.44	88.22	90.22
HSISSA-RF	97.22	97.33	95.44	96.33

E. OPTIMIZATION OF RANDOM FOREST MODEL

In order to further illustrate the applicability of HSISSA, this paper uses the classical random forest algorithm for validation and proposes a HSISSA-RF model to demonstrate the better optimization performance of the improved algorithm.

In this paper, HSISSA is used to optimize the number of decision trees, the minimum number of samples that can be divided into nodes, the maximum number of leaf nodes and the maximum depth of the decision tree in the random forest algorithm. The search range is set as follows: $l_b = [20, 5, 30, 10]$, $u_b = [100, 20, 50, 100]$. Set the minimum error rate of verification set during training as er, the number of iterations is n, the fitness function is shown in (26). Then use SSA and HSISSA to optimize the parameters respectively, and compare the optimization results of the two algorithms.

TABLE 18. Comparison results of all models.

model\(%)	**Informedness**	**Markedness**	**Correlation**
DenseNet-121	87.76	90.88	89.31
RF	81.13	84.91	82.99
SSA-DenseNet	94.24	94.85	94.54
HSISSA-DenseNet	97.67	98.31	97.99
SSA-RF	87.13	92.37	89.71
HSISSA-RF	94.98	96.96	95.96

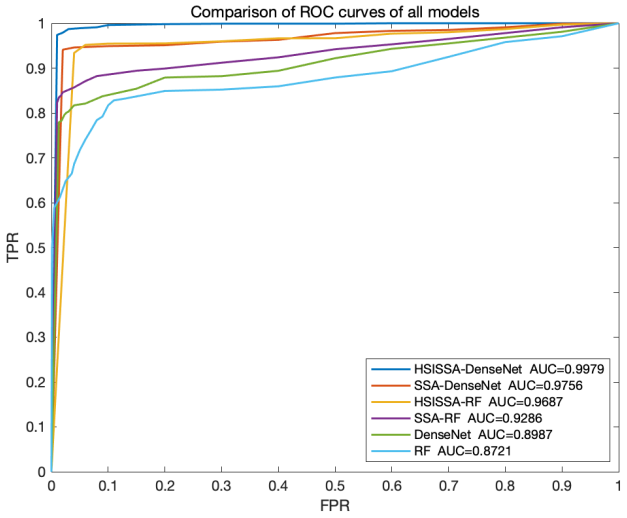


FIGURE 11. Comparison of ROC curves of six models.

The optimized parameters of the HSISSA-RF model are: [76, 10, 49, 72], the optimized parameters of the SSA-RF model are: [93, 9, 46, 32]. Table 15 and Table 16 further analyze the test set accuracy of the two models. The average accuracy of HSISSA-RF model is 4% higher than that of SSA-RF model. For BENIGN, DDos, Brute Force Attack, Bot, Web Attack, Infiltration and Heartbleed, HSISSA-RF model is 4%, 1%, 7%, 13%, 5%, 2% and 3% higher. It can be seen that HSISSA has better optimization performance.

Table 17 and Table 18 shows the comparison of detection performance between the four optimized models and the unoptimized models in this paper. It can be seen from the Table 17 that the optimization results of HSISSA are better than those of SSA, and both are better than those of the original model. The accuracy and F1 score of HSISSA-DenseNet model are higher than other models. It can be seen from Table 18 that the model optimized by HSISSA is also the highest for both informedness and markedness, and the correlation is the closest to 1, and the prediction is the most accurate. Figure 11 shows the ROC curve comparison of six models. The curve of HSISSA-DenseNet is closest to the upper left corner, with the highest accuracy, the largest AUC area and the best performance. The ROC curve and AUC area of the model optimized by HSISSA algorithm are better than those before optimization. Therefore, the HSISSA algorithm proposed in this paper can improve the performance

of the intrusion detection model, and has a certain degree of adaptability.

VII. CONCLUSION

To solve the problem of slow convergence speed and the possibility of falling into local optima of SSA, this paper proposes a population initialization method for a hybrid circle piecewise map, which enhances the randomness of the population distribution. A spiral exploration method and Levy flight operator were introduced to update the positions of discoverers and scouters in the population, thereby enhancing the global exploration performance of the algorithm. The simplex method and pinhole imaging reverse strategy were used to optimize the position of poor and optimal sparrows, avoiding stagnation of population search and reducing the possibility of falling into local optimization. To prove the superiority of the improved algorithm, we selected ten test functions and five popular algorithms (SCA, GWO, WOA, PSO, and SSA) for comparison. From the convergence curve and the table of function operation results, it can be seen that HSISSA converges faster and has a higher accuracy than the other algorithms. Therefore, the HSISSA is a more efficient, accurate, and robust algorithm.

In the application research of intrusion detection, the CIC-IDS2017 dataset and UNSW-NB15 dataset are preprocessed first, and then two different objective functions are designed for optimization. The feature selection experiment and model parameter optimization experiment are carried out respectively. On the CIC-IDS2017 dataset and UNSW-NB15 dataset, HSISSA retained 7.6 features and 10.1 features on average, with the accuracy rate of 99.5% and 96.01%, 5.7 features and 5.6 features less than that of SSA, and improved the accuracy rate of 1.53% and 2.45%. The accuracy of HSISSA-DenseNet and HSISSA-RF models has reached 99.34% and 97.22%, and other metrics including F1 score, correlation, etc. are all optimal, which are improved to a certain extent compared with the SSA-optimized model and the original model, especially for the classification of attacks of Brute Force Attack, Bot, and Web Attack has been significantly improved. These experimental results show that HSISSA algorithm can efficiently perform feature selection and model optimization, and has considerable applicability in the field of intrusion detection.

Although some achievements were made in this study, there are still some work to be improved in the future. One is to study the impact of appropriate parameters on the performance of the algorithm, and further optimize the algorithm by combining the advantages of other swarm intelligence algorithms. The other is to study the processing of unbalanced data sets. The data generated by using SMOTE oversampling cannot perfectly fit the real data. Finally, for models with deep network layers, the algorithm optimization process is slow. How to improve the optimization rate is also a major focus of future research.

REFERENCES

[1] M. H. Nasir, S. A. Khan, M. M. Khan, and M. Fatima, "Swarm intelligence inspired intrusion detection systems—A systematic literature review," *Comput. Netw.*, vol. 205, no. 14, Mar. 2022, Art. no. 108708.
[2] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern., B, Cybern*, vol. 26, no. 1, pp. 29–40, Dec. 1994.
[3] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, pp. 69–74, Feb. 2012.
[4] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[5] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.
[6] M. Khishe and M. R. Mosavi, "Chimp optimization algorithm," *Exp. Syst. Appl.*, vol. 149, Jul. 2020, Art. no. 113338.
[7] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *J. Bionic Eng.*, vol. 18, no. 3, pp. 674–710, May 2021.
[8] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat swarm optimizer," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 8457–8482, Oct. 2020.
[9] N. Chopra and M. M. Ansari, "Golden jackal optimization: A novel nature-inspired optimizer for engineering applications," *Exp. Syst. Appl.*, vol. 198, Jul. 2022, Art. no. 116924.
[10] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, Jan. 2020.
[11] Y. Q. Tang, L. Chenghai, S. Yafei, and C. B. Chenchen, "Adaptive mutation sparrow search optimization algorithm," *J. Beijing Univ. Aeronaut. Astronaut.*, vol. 6, p. 5425, Jul. 2021, doi: 10.13700/j.bh.1001-5965.2021.0282.
[12] J. Yuan, Z. Zhao, Y. Liu, B. He, L. Wang, B. Xie, and Y. Gao, "DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm," *IEEE Access*, vol. 9, pp. 16623–16629, 2021.
[13] X. Ren, S. Chen, K. Wang, and J. Tan, "Design and application of improved sparrow search algorithm based on sine cosine and firefly perturbation," *Math. Biosciences Eng.*, vol. 19, no. 11, pp. 11422–11452, 2022.
[14] J. Ma, Z. Hao, and W. Sun, "Enhancing sparrow search algorithm via multi-strategies for continuous optimization problems," *Inf. Process. Manage.*, vol. 59, no. 2, Mar. 2022, Art. no. 102854.
[15] Y. Zhang, P. Li, and X. Wang, "Intrusion detection for IoT based on improved genetic algorithm and deep belief network," *IEEE Access*, vol. 7, pp. 31711–31722, 2019.
[16] T. Dash, "A study on intrusion detection using neural networks trained with evolutionary algorithms," *Soft Comput.*, vol. 21, no. 10, pp. 2687–2700, May 2017.
[17] X. Li, P. Yi, W. Wei, Y. Jiang, and L. Tian, "LNNLS-KH: A feature selection method for network intrusion detection," *Secur. Commun. Netw.*, vol. 2021, Jan. 2021, Art. no. 8830431, doi: 10.1155/2021/8830431.
[18] Y. Du, Z. M. Wang, and M. H. Li, "Industrial control intrusion detection method based on optimized kernel extreme learning machine," *Netinfo Secur.*, vol. 21, no. 2, pp. 1–9, 2021.
[19] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 6125–6137, 2019.
[20] S. Malliga, P. S. Nandhini, and S. V. Kogilavani, "A comprehensive review of deep learning techniques for the detection of (distributed) denial of service attacks," *Inf. Technol. Control*, vol. 51, no. 1, pp. 180–215, Mar. 2022.
[21] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: A systematic review," *IEEE Access*, vol. 9, pp. 101574–101599, 2021.
[22] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci.*, vol. 513, pp. 386–396, Mar. 2020.
[23] J. Yoo, B. Min, S. Kim, D. Shin, and D. Shin, "Study on network intrusion detection method using discrete pre-processing method and convolution neural network," *IEEE Access*, vol. 9, pp. 142348–142361, 2021.

[24] E. A. Shams, A. Rizaner, and A. H. Ulusoy, "A novel context-aware feature extraction method for convolutional neural network-based intrusion detection systems," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 13647–13665, Oct. 2021.
[25] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107315.
[26] C. Pak, K. An, and P. Jang, "A novel bit-level color image encryption using improved 1D chaotic map," *Multimedia Tools Appl.*, vol. 78, no. 9, pp. 12027–12042, May 2019.
[27] Y. F. Wang, Q. Liu, and J. W. Sun, "Multistrategy improved sparrow search algorithm optimized deep neural network for esophageal cancer," *Comput. Intell. Neurosci.*, vol. 2022, Sep. 2022, Art. no. 1036913.
[28] B. Gao, W. Shen, H. Guan, L. Zheng, and W. Zhang, "Research on multistrategy improved evolutionary sparrow search algorithm and its application," *IEEE Access*, vol. 10, pp. 62520–62534, 2022.
[29] H. A. Alsattar, A. A. Zaidan, and B. B. Zaidan, "Novel meta-heuristic bald eagle search optimisation algorithm," *Artif. Intell. Rev.*, vol. 53, no. 3, pp. 2237–2264, Mar. 2020.
[30] S. M. Wang, Z. Q. Ren, and J. T. Song, "Transient electromagnetic method inversion based on Lévy flight-particle swarm optimization," *Chin. J. Geophys.-Chin.*, vol. 65, no. 4, pp. 1482–1493, Apr. 2022.
[31] C. H. Liu and H. E. Qing, "A simplex method guided sparrow search algorithm based on improved search mechanism," *Comput. Eng. Sci.*, vol. 44, no. 12, pp. 1–9, Dec. 2021.
[32] D. M. Zhang et al., "Whale optimization algorithm for embedded circle mapping and one-dimensional oppositional learning based small hole imaging," *Control Decis.*, vol. 36, no. 5, pp. 1173–1180, May 2021.
[33] J. Derrac et al., "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 1–3, 2011.
[34] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.
[35] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "Machine learning and deep learning approaches for cybersecurity: A review," *IEEE Access*, vol. 10, pp. 19572–19585, 2022.
[36] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Int. J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2020.
[37] A. Jumabek, S. Yang, and Y. Noh, "CatBoost-based network intrusion detection on imbalanced CIC-IDS-2018 dataset," *J. Korean Inst. Commun. Inf. Sci.*, vol. 46, no. 12, pp. 2191–2197, Dec. 2021.

LIU TAO received the Ph.D. degree in security technology and engineering from the Xi'an University of Science and Technology, in 2009. His research interests include cyber security, artificial intelligence, and big data analytics.

MENG XUEQIANG received the bachelor's degree in engineering from the North China Institute of Science and Technology, in 2018. He is currently pursuing the master's degree in electronic information with the Xi'an University of Science and Technology. His current research interests include biologically inspired algorithms, cyber security, and deep learning.