Research on Network Security Situation Assessment Based on DSAE-TBSMACNN Model

Hong Dai*, Qiaochu Sun, Jiwang Sun, Baiping Sun

Abstract—**The Slime Mould Algorithm (SMA) is a novel metaheuristic algorithm inspired by Physarum polycephlum's foraging habits. It establishes a commendable equilibrium between exploitative and exploratory search, utilizing fewer parameters. However, because of its early convergence on the reference function and low computational accuracy, it quickly falls into the local optimum. We provide an enhanced Slime Mould Algorithm based on tangent flight operator and probability distribution (TBSMA) to increase the algorithm's speed and convergence accuracy. The TBSMA position update algorithm includes the tangent flight operator. Its adaptive, adjustable step size can significantly improve search capabilities and accelerate the convergence rate. Including a parameter produced by a probability distribution in the parameter calculation procedure enhances the algorithm's capacity to exit the local optimum. The paper ultimately integrates the enhanced TBSMA algorithm with the CNN to develop the TBSMA-CNN model. Preprocessing data creates the new DSAE-TBSMACNN model to lower the dataset's dimensionality using the DSAE model from the earlier study. The experiments use the CIC-IDS2017 dataset and calculate to evaluate the model's performance through accuracy, precision, recall, and F1-score. The research concludes by designing the DSAE-TBSMACNN model for network security situation assessment experiments. The model evaluation findings are thoroughly compared with the classical model to validate the model's performance. In evaluating network security situation values and levels, the experimental results verify that the model suggested performs better than others.**

Index Terms—**The Slime mould algorithm, Tangent flight operator, Network Security Situation, Graph Convolutional Network, Long Short-Term Memory, Self-Attention Mechanism**

I. Introduction

The openness, complexity, and diversity of networks have made network security more challenging in today's quickly expanding technologies. As new types of network threats keep appearing, network security situation assessment has steadily grown in popularity over time. It notifies network management of potential risks and attacks ahead of time, allowing for improved resource allocation and risk mitigation.

Network security situation assessment is an identification, understanding, and assessment of various activities in the network to grasp the security status of a whole network and support reasonable security response decisions [1]. It is primarily quantified by the situation value, thus confirming the situation level. The basic process involves extracting situational elements, assessing, and forecasting. In recent years, the primary network security situation assessment techniques include hierarchical analysis, Hidden Markov Models, and deep learning. Deep learning techniques are popular among researchers [2-5]. Wang et al. [6] used PNN for network security situation assessment. They used GA to optimize the correction factor of PNN to improve the accuracy and stability, but the assessment time is extended when the training samples are too small. Zhang et al. [7] used a simulated annealing algorithm and sparrow search algorithm to improve the BP neural network to improve the network security assessment of convergence. Hongyu et al. [8] used a bi-directional gated cyclic unit with an improved attention mechanism for situation assessment; the model improves the reliability of the results but does not refine the quantitative metrics, which may lead to less accuracy in the network security status. The fusion of intelligent optimization algorithms with deep learning algorithms plays a significant role in network security situational assessment.

Metaheuristic algorithms (MAs), intelligent optimization algorithms, become powerful tools to solve optimization problems. It is usually a general-purpose heuristic algorithm that does not rely on the specific conditions of a particular situation. It has excellent usability, randomness, and flexibility to be applied to a broader range of aspects. It has good flexibility, stronger robustness, and self-organization, which provides new ideas for many practical problems. In recent years, many advanced swarm intelligence optimization algorithms have been proposed by scholars, such as the Whale Optimization Algorithm (WOA) [9], the Manta Ray Foraging Optimization (MRFO) [10], the Artificial Bee Colony (ABC) [11], the Fruit Fly Optimization Algorithm (FOA) [12], the Gray Wolf Optimizer (GWO) [13] and other sophisticated swarm intelligence optimization algorithms. Although many experimental results demonstrate that the swarm intelligence optimization algorithm is a valuable tool for solving actual optimization problems, these algorithms have flaws, such as an imbalance between exploitation and exploratory search, low search accuracy, too many control parameters, a tendency to fall into the local optimal, and so on. The Slime

Manuscript received August 24, 2024; revised February 7, 2025. The research work was supported by the Fundamental Research Funds for the Liaoning Universities (NO. LJ212410146058) and the Graduate student Science andTechnology Innovation project of University of Science and Technology Liaoning (No.LKDYC202423).

Hong Dai* is a professor in the School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051 China. (corresponding author to provide phone: +086-186-4226-8599; fax: 0412-5929818; e-mail: dear_red9@163.com).

Qiaochu Sun is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051 China. (e-mail: 912410855@qq.com).

Jiwang Sun is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051 China. (e-mail: 1470573808@qq.com).

Baiping Sun is a postgraduate student of University of Science and Technology Liaoning, Anshan, Liaoning, CO 114051 China. (e-mail: Sunbaiping2023@163.com).

Mould Algorithm (SMA) can achieve better development while ensuring exploration performance. It balances the exploitation and exploration search excellently and uses fewer parameters. The SMA is a new MA proposed by Li et al. [14] in 2020. The algorithm simulates three unique foraging strategies of Physarum polycephalum: approaching food, wrapping food, and grabbing food. Slime mold finds food by forming a network of interconnected veins and using smells in the air to identify and surround food.

As the SMA algorithm is relatively new, it has some shortcomings, such as falling into local optimum when the solving problem dimension is too large, so scholars have begun to pay attention to improving the SMA algorithm. In terms of algorithm improvement, Chauhan et al. [15] combined the Arithmetic Optimizer Algorithm (AOA) with the SMA and introduced Lens opposition-based learning (LOBL). The randomness and development ability of the algorithm were improved. Chen et al. [16] proposed the chaotic SMA (CSMA), which alleviated the slow convergence and improved the search efficiency. Naik et al. [17] presented a leader SMA (LSMA), which used the three best candidate solutions to update the location of slime mold. It made it possible to increase efficiency and successfully strike a balance between exploration and exploitation. Ewers et al. [18] proposed an FS method based on the firefly algorithm (FA) to improve the search mode of the SMA and enhance convergence performance. The SMA algorithm also has some defects, such as low computational accuracy, premature convergence on some reference functions, and easy to fall into local optimum.

This research suggests an enhanced sticky bacteria method (TBSMA) based on probability distribution and tangent flight operator, improving the SMA algorithm. The technique improves the SMA algorithm by enhancing its global search capability and preventing it from reaching a local optimum. The CNN hyperparameters are optimized using the TBSMA algorithm under non-empirical supervision, which improves the hyperparameter performance and deceptively selects the optimal network structure. Using DSAE from the literature, the features of each assault type are deferentially retrieved by hierarchical classification, followed by layer-by-layer dimensional reduction and data fusion to produce strongly correlated features. The novel model for network security situational assessment, the DSAE-TBSMACNN, is created by combining the DSAE with the recently enhanced TBSMA-CNN model. The network intrusion detection CIC-IDS2017 [19] public dataset is used for experimental simulation analysis and verification. The experimental results show that the model achieves good accuracy, precision, recall, and F1 score results. Finally, the performance of CNN models the PSO-CNN, the MRFO-CNN, the AAA-CNN, the SVM, and the LSTM optimized by the PSO algorithm, the MRFO algorithm, and the AOA algorithm are compared with the DSAE-TBSMACNN model proposed in this paper. The model for network security situation assessment. The experimental results show that the proposed model is superior to the comparison models in terms of the accuracy, precision, recall and F1 value. The improved model is also feasible for network security situation assessment.

II. PROPOSED METHOD

A. The Slime Mould Algorithm (SMA)

Slime mold refers to Physarum polycephalum, which lives in damp, cold places. The foraging behavior of slime mold mainly inspires the Slime Mould Algorithm. In the foraging stage, the slime mold will first self-diffuse and form a vein network of varying thickness among food, whose thickness is determined by the odor concentration in the air. The higher the concentration, the more intense the bio oscillation wave, the faster the cytoplasmic flow, and the greater the flow. It causes an increase in diameter. In other words, it is the venous dilation. Slime mold can find the highest concentration and optimal food path through the vein network that resembles channels. They then surround the food and secrete enzymes to digest it. Even when slime mold has plenty of food or is already fed, it will still search for higher-quality food sources. It follows that it can exploit both resources at the same time. The algorithm simulates three unique foraging strategies of Physarum polycephalum: approaching food, wrapping food, and grabbing food. Slime mold uses the odor concentration in the air to approach food, in which the weight of positive and negative feedback is used to simulate the oscillating wave of slime mold. The oscillating wave of slime mold positively correlates with food concentration, as shown in Eq. (1).

$$\overrightarrow{X^*}=\begin{cases} rand\cdot(UB-LB)+LB, rand<z \\ \overrightarrow{X_b(t)}+\overrightarrow{vb}\cdot(W\cdot\overrightarrow{X_A(t)}-\overrightarrow{X_B(t)})\times 0.3\times\tan(k_2\times\frac{\pi}{2}), r<p \\ \overrightarrow{vc}\cdot\overrightarrow{X(t)}, r\geq p \end{cases} \quad (1)$$

Among them, $\overrightarrow{X_b(t)}$ represents the current location of the highest food concentration, namely the current individual optimal solution. $\overrightarrow{vb}$ is a random number in the range of [-a, a]. $\overrightarrow{W}$ is the weight coefficient of slime mold. $\overrightarrow{X_A(t)}$ and $\overrightarrow{X_B(t)}$ are two random individual positions. $\overrightarrow{vc}$ is a parameter that decreases linearly from 0 to 1. $\overrightarrow{X(t)}$ is the current position of slime mold. t is the current iteration number. The coefficient p is shown in Eq. (2).

$$p=\tanh\left|S(i)-DF\right| \quad (2)$$

Where, $i\in 1,2,...,n$, $S(i)$ denotes the fitness of $\overrightarrow{X(t)}$; DF is the fitness of the overall optimal solution.

The random number $\overrightarrow{vb}$ is shown in Eq. (3). And the coefficient a is shown in Eq. (4).

$$\overrightarrow{vb}=[-a,a] \quad (3)$$

$$a=\arctan h\left(-\left(\frac{t}{\max\_t}\right)+1\right) \quad (4)$$

Where, max_t is the maximum iteration. The weight coefficient of slime mold is shown in Eq. (5) which simulates the relationship between the thickness of the slime mold vein network and food concentration using weights of positive and negative feedback.

$$\overrightarrow{W(SmellIndex(i))}=\begin{cases} 1+r\cdot\log\left(\frac{bF-S(i)}{bF-wF}+1\right), condition \\ 1-r\cdot\log\left(\frac{bF-S(i)}{bF-wF}+1\right), others \end{cases} \quad (5)$$

The worst fitness for the current iteration is represented by a random value between 0 and 1, which simulates the

uncertainty of venous contraction and dilatation. Using fitness calculation can alleviate numerical changes. Eq. (6) displays the means of ranking the population of slime mold according to their fitness. This indicates which slime molds are in the upper half. The rest of the slime mold is as well.

$$SmellIndex = sort(S) \tag{6}$$

The location and search mode are modified based on the concentration of food to more accurately replicate the constriction mode of the slime mold vein network. As seen below, its location update Eq. (1) is enhanced to Eq. (7).

$$\overrightarrow{X^*} = \begin{cases} rand \cdot (UB - LB) + LB, rand < z \\ \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot \left(W \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)} \right), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \tag{7}$$

UB and *LB* represent the upper and lower bounds and denote a random number between 0 and 1; z is a parameter that maintains the balance between the exploitation and exploration search. It can be a number between 0 and 0.1, but the effect is best when the value is 0.03.

B. The Improved Slime Mould Algorithm

The tangent flight operator is a mathematical model using the tangent function, which acts as a flight function similar to the Levy flight function, facilitating the balance between development and exploration search [21]. Its mathematical expression is as follows:

$$k_1 = \tan(k_2 \times \frac{\pi}{2}) \tag{8}$$

$$k_2 = 2 \times rand() - 1 \tag{9}$$

where k_2 is a random value that is equally distributed between -1 and 1, and k_1 is the step size from tangent flight. Fig. 1 displays the flight path after 1000 iterations.

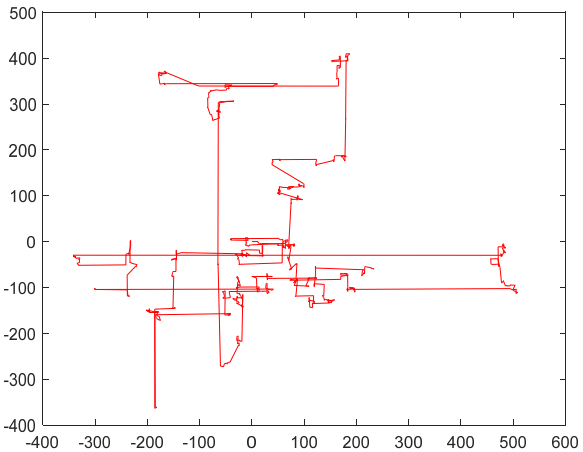


Fig. 1. Step size of tangent flight operator

The attributes of tangent flight operators comprise a blend of brief and concise processes, which can augment the algorithm's capacity for local and worldwide exploration. It is conducive to the exploitation and exploration of balanced algorithms [22]. We incorporate tangent flight operators into the SMA position update calculation, which improves the algorithm's search capability and balances exploration and exploitation. When the algorithm falls into the local optimum, it may get a more significant step size, which increases the possibility of the algorithm jumping out of the local optimum. The new position update formula obtained by adding a tangent flight operator is shown below.

$$\overrightarrow{X^*} = \begin{cases} rand \cdot (UB - LB) + LB, rand < z \\ \overrightarrow{X_b(t)} + \overrightarrow{vb} \cdot (W \cdot \overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}) \times 0.3 \times \tan(k_2 \times \frac{\pi}{2}), r < p \\ \overrightarrow{vc} \cdot \overrightarrow{X(t)}, r \geq p \end{cases} \tag{10}$$

There is another important parameter in the position update formula of the SMA. It is a linear decreasing parameter in the range of -1 to 1, and its changing trend is shown in Fig. 2.

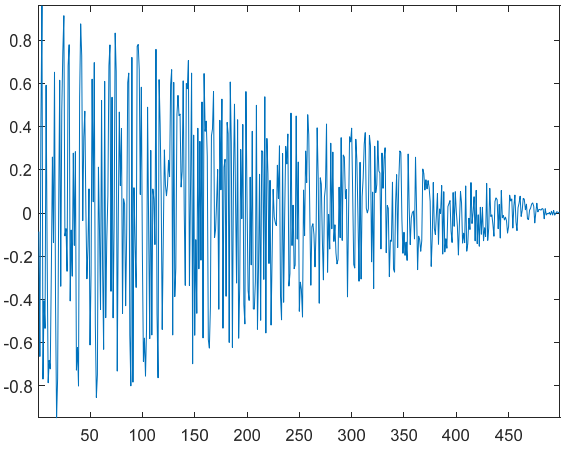


Fig. 2. The variation tendency of the original vc

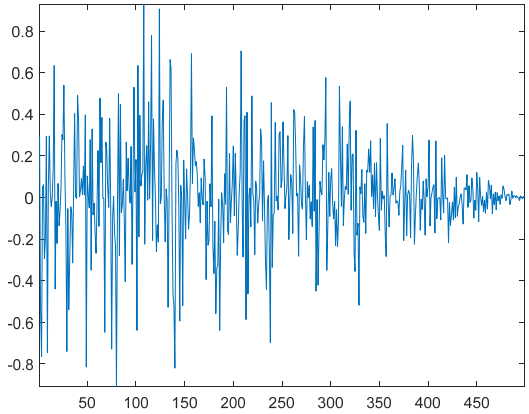
Because the original parameter has a linear decreasing trend, it decreases gradually with the algorithm's running, which may cause the algorithm to fall into the local optimum in the late running period. We change the linear decreasing trend of the original parameter into a nonlinear decreasing trend to slow down its decreasing trend. At the same time, the parameter generated by different probability distributions is added to enhance its randomness. It can expand the search step length and improve the ability to jump out of the local optimum. We choose six probability distributions: Uniform distribution, Normal distribution, Weibull distribution, Rayleigh distribution, Exponential distribution, and Beta distribution. The expressions of the six probability distributions are shown in Table I. Eq. (11) and the calculation formula in Eq. (12) show the improved mathematical expression.

$$vc = unifrnd(-b, b, 1, \dim) \tag{11}$$

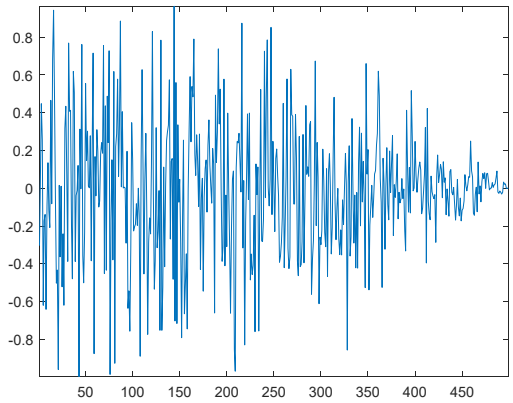
$$b = \left(1 - \left(\frac{it}{Max\_iter}\right)^2\right) \times k_3 \tag{12}$$

TABLE I
EXPRESSIONS OF SIX PROBABILITY DISTRIBUTIONS

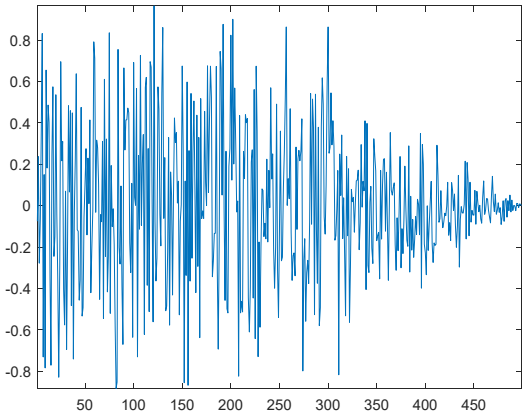
Method	k_3
Uniform distribution	$rand(\ )$
Normal distribution	$normrnd(0.8, 0.4)$
Weibull distribution	$wblrnd(1, 4)$
Rayleigh distribution	$raylrnd(0.6)$
Exponential distribution	$exprnd(0.7)$
Beta distribution	$betarnd(3.2)$



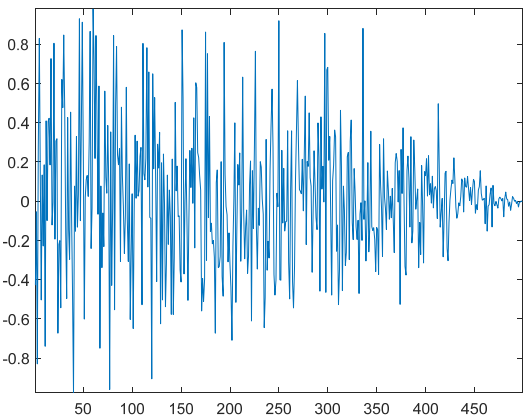
(a) Uniform distribution



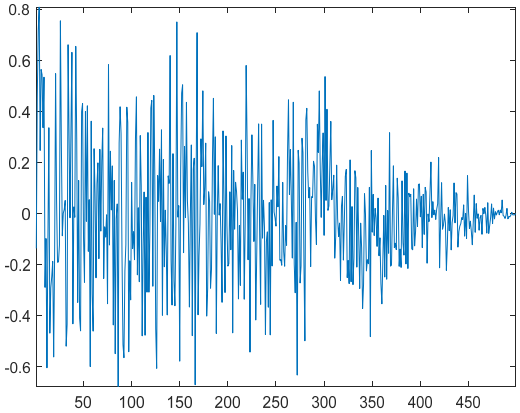
(b) Normal distribution



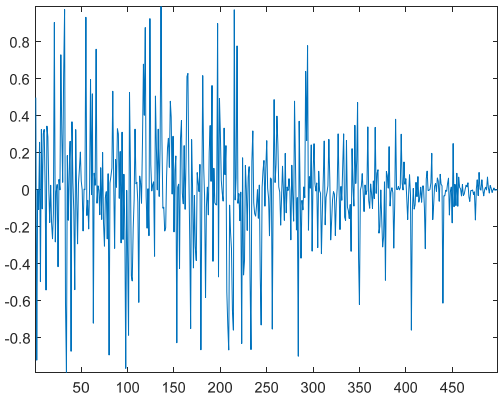
(c) Weibull distribution



(d) Rayleigh distribution



(e) Exponential distribution



(f) Beta distribution

Fig. 3. Step size of tangent flight operator

Fig. 3 shows the changing trend of the six probability distributions. According to the improved method, the pseudo-code of the improved SMA algorithm is given in Table II.

TABLE II
THE IMPROVED SMA ALGORITHM PSEUDO-CODE

Algorithm 1: The improved SMA algorithm

Input: Population size N;
maximum number of iterations Max_iter;
random candidate solutions $X_i(i-1,2,...,n)$;

Output: Best fitness value F; optimal solution X_b;

```
Init(N, Max_iter, X_i)
While t <= Max_Iteration
  For each slime_mould in population
    Calculate fitness_value(slime_mould)
  End For
    Update bestFitness
    Calculate W by Eq. (5)
    Update a by Eq. (4)
    Update b by Eq. (11)
  For each search_portion
     Update p, vb, vc
     Update positions by Eq. (10)
  End For
  t = t + 1
End While
Return bestFitness
```

C. The TBSMA-CNN Model

The paper makes improvements based on the classical LeNet-5 model, which has eight layers, including four convolutional layers, two pooling layers, and 2 fully connected layers. Firstly, a greyscale image of 7*7 size generated by image processing of network flow data is taken as input, and the first four layers are convolutional layers, in which a pooling layer is interspersed behind every two convolutional layers for maximum pooling operation. Subsequently, the data is one-dimensionalized by a flattened layer, which transitions between the convolutional and fully connected layers. The last two layers are fully connected, and the second fully connected layer applies the Softmax function for classification. A dropout layer is added between the two layers to avoid overfitting.

In the TBSMA-CNN model, the TBSMA optimization algorithm is used to find the CNN with optimal performance, which avoids the problem of the optimal hyperparameters relying on manual configuration, thus wasting a lot of resources and time. The main steps of the TBSMA-CNN algorithm are designed as follows:

1) Determine the CNN parameters that need to be optimized by the TBSMA and set the range of values;

2) Initialize the TBSMA parameters, such as the number of iterations and population size;

3) Randomly generate candidate solutions as the CNN structures;

4) Update the position of individuals according to equation 10;

5) Go through the CNN training to calculate the CNN loss function;

6) Judge the size of the fitness of the candidate solution and the fitness of the individual or the global optimal solution. If the fitness of the candidate solution is smaller than the best fitness value, then update;

7) Judge whether the current is the maximum number of iterations; if the condition is satisfied, get the optimal CNN network structure; otherwise, repeat steps 4 to 6.

III. Model Experimental Validation

A. Experiment Data and Model Parameter Settings

The CIC-IDS2017 dataset is used in this experiment for simulation, as it contains all the necessary criteria and provides eight common attack types with high authority. The CIC-IDS2017 dataset is converted into 7*7 grey scale images according to the image-based operation through data preprocessing, which provides a better initial point for the subsequent model evaluation.The experiments were conducted using PyCharm for training and testing on a platform with an Intel Core i5, 16 GB RAM, Windows 10, and a 64-bit operating system with the machine learning library TensorFlow 2.6.0.

The paper designs a network security situation assessment model based on the TBSMA-CNN model. The paper improves the classical LeNet-5 model, and the TBSMA optimization algorithm optimizes the parameters. The TBSMA-CNN model uses 7*7 greyscale image data as its input. To get the best performance out of the model, the TBSMA optimization is applied to the hyperparameters, including the number and size of convolution kernels in the convolutional layer, the size of the pooling kernels in the pooling layer, and the step size. The standard CNN parameters and the parameters optimized by the TBSMA algorithm are shown in Table III. The number of iterations of the TBSMA algorithm for this experiment is 30, and the population size is 20.

B. Experimental Results of Network Security Situation Element Extraction

The network security situation element extraction experiment mainly includes the situation element extraction and situation classification modules. They are mostly carried out on the CIC-IDS2017 dataset through the DSAE-TBSMACNN model in the paper.

The accuracy and loss values of the model are critical indicators to judge the performance of the model. In this paper, we plot the trend of accuracy and the loss function for 25 training iterations on the CIC-IDS2017 dataset, as shown in Fig. 4. From the figure, it can be observed that the accuracy rate improves through the increase in the number of iterations and levels off after five rounds and steadily improves. When the number of iterations reaches 25, the accuracy rate is 0.995, and the loss function is 0.053.

Meanwhile, the accuracy of the classification results is evaluated using the confusion matrix, as shown in Table IV. There is a misclassification between the attack flow and the normal flow mainly because some of the normal flow's eigenvalues do not differ much from the attack flow features.

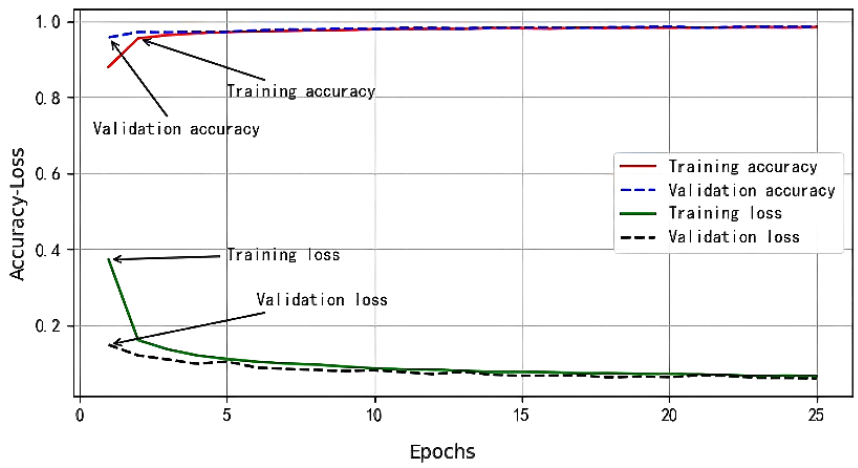


Fig. 4. Time distribution of TBSMA-CNN training model

In order to verify the performance of the models in this paper, the experiments also compare the following three models:

1) The original CNN model;

2) The SMACNN model: the initial parameters and network structure of the CNN are optimized by the original slimeball algorithm;

3) The TBSMA-CNN model: the improved slimeball algorithm based on the tangent flight operator and probability distributions is used to optimize the CNN model;

4) The DSAE-CNN model: the data set is down scaled using the directed acyclic graph SAE for dimensional reduction of the dataset.

Their performance is evaluated by the above four evaluations, namely, accuracy, precision, recall, and F1-score. The results of the comparison experiments of the five methods on the CIC-IDS2017 dataset are shown in Table IV.

TABLE III
PARAMETER OPTIMIZATIONS OF CNN AND TBSMACNN

Layers	CNN Network Layer	hyperparameter	Setting range	CNN	TBSMA-CNN
L1	Conv2D	Filters	[4,101]	32	36
		Kernel size	[3×3,5×5,7×7]	5×5	7×7
L2	Conv2D	Filters	[4,101]	16	69
		Kernel size	[3×3,5×5,7×7]	5×5	5×5
L3	MaxPooling2D	Pool size	[2,4]	2	3
		Strides	[2,4]	2	4
L4	Conv2D	Filters	[4,101]	16	68
		Kernel size	[3×3,5×5,7×7]	5×5	5×5
L5	Conv2D	Filters	[4,101]	32	72
		Kernel size	[3×3,5×5,7×7]	5×5	3×3
L6	MaxPooling2D	Pool size	[2,4]	2	2
		Strides	[2,4]	2	4
L7	Dense	Neurons	[4,201]	128	134

TABLE IV
TESTING SET CONFUSION MATRIX

Predicted \ True	2	3	4	5	6	7	8
1	0	0	0	22	0	1	0
2	1500	0	0	0	0	0	0
3	0	1500	0	0	0	0	0
4	0	0	1500	0	0	0	0
5	0	0	0	1478	0	0	0
6	0	0	0	0	1500	0	0
7	0	0	0	0	0	1499	0
8	0	0	0	0	0	0	1500

TABLE V
PERFORMANCE COMPARISON BETWEEN DSAE-TBSMACNN AND OTHER MODELS

Model	Accuracy/%	Precision/%	Recall/%	F1-score/%
CNN	93.8	93.9	94.0	93.8
SMACNN	94.6	94.7	94.5	94.5
TBSMA-CNN	95.6	95.8	95.3	95.6
DSAE-CNN	96.2	96.2	96.4	96.1
DSAE-TBSMACNN	99.5	99.4	99.5	99.2

TABLE VI
PERFORMANCE EVALUATION AFTER APPLYING DSAE-TBSMACNN

Model	Performance indicators	Model performance/%							
		Benign	DoS	PortScan	DDoS	BForce	Web Attack	Bot	Infiltration
CNN	Accuracy	77.5	99.1	98.4	96.3	93.7	97.4	99.7	89.5
	Precision	76.4	99.7	98.7	97.5	94.5	97.7	99.5	89.8
	Recall	94.8	88.5	96.7	92.0	89.1	96.1	97.9	97.0
	F1-score	85.3	93.5	97.5	94.1	91.3	96.7	98.7	93.1
TBSMA -CNN	Accuracy	78.0	91.7	93.6	92.3	92.3	94.7	95.7	91.6
	Precision	93.9	93.1	95.7	93.7	96.0	97.9	95.8	98.9
	Recall	83.3	99.0	98.0	99.0	96.3	96.7	100.0	92.7
	F1-score	88.3	95.9	96.9	96.3	96.2	97.3	97.8	95.7
DSAE-CNN	Accuracy	88.5	99.0	100.0	97.2	95.4	98.8	99.6	90.3
	Precision	96.3	97.5	98.4	96.0	95.7	98.9	97.3	90.0
	Recall	95.3	90.5	97.3	97.4	94.3	98.2	97.1	99.3
	F1-score	91.8	94.6	98.6	97.3	94.8	98.5	98.3	94.6
DSAE-TBSMA CNN	Accuracy	99.5	100.0	100.0	100.0	96.0	100.0	99.9	100.0
	Precision	98.5	100.0	100.0	100.0	97.5	100.0	100.0	100.0
	Recall	97.5	100.0	100.0	100.0	98.5	100.0	99.9	100.0
	F1-score	97.0	99.5	98.0	100.0	98.0	99.2	99.9	100.0

From Table V, it can be seen that among all the models, the DSAE-TBSMACNN model is the most effective, with an accuracy of 99.5%. This fully demonstrates that the proposed model can be advantageous in selecting feature degradation and the CNN model structure. Comparing the CNN models optimized by the slime mold algorithm, both improve by 0.8% and 1.8% compared to the original CNN accuracy, and the improved slime mold algorithm optimization performance is better. The accuracy of the DSAE-CNN model and the DSAE-TBSMACNN model is enhanced by 2.4% and 3.9%, respectively, compared with the models before improvement. Meanwhile, when optimized based on the TBSMA, the TBSMA-CNN model and the DSAE-TBSMACNN model's accuracies are better than the previous model by 1.8% and 3.3%, respectively. It shows that the DSAE-TBSMACNN model proposed can accurately and effectively predict the identification of attacks. The results of the DSAE-TBSMACNN model and two on the CIC-IDS 2017 dataset for eight classifications are shown in Table VI. It can be intuitively seen that this paper's model, the DSAE-TBSMACNN, performs well in all four evaluation metrics. This paper's model has a noticeable improvement in accuracy and F1-score and excellent performance in precision and recall. In DDoS data, the precision of this paper's model is improved by 1.5%, 6.3%, and 4.0% compared to the other three models. This paper's model improves recall by 2.5%, 1.9%, and 0.7% on Infiltration data. In summary, the DASE-TBSMACNN model in this paper extracts more representative features by feature dimensional reduction using the DSAE. Imaging the raw data can be easily observed. It can also constitute a very effective multi-dimensional feature space to improve the model accuracy. The TBSMA algorithm optimizes the CNN structure to exploit the SMA's optimization search. The tangent flight operator added to the position update formula enhances the global search capability by combining long and short steps. Meanwhile, the random numbers generated by the beta probability distribution slow down the downward trend of the original VC, thus expanding the search step length and improving the model's ability to jump out of the local optimum.

C. Model Comparison Experiment

To objectively verify the performance of the proposed model for situation extraction and classification, the paper performs network security situation extraction and classification experiments on the CIC-IDS2017 dataset using the LSTM network and support vector machine, and also the CNN model optimized with the PSO, the AOA, and the MRFO. The CNN structure is optimized by using mainstream meta-heuristic algorithms to verify the applicability of the TBSMA algorithm for the CNN hyperparameter problem. The dataset preprocessing technique is consistent across all models to achieve accurate findings. Table VII displays the comparison results among models for the four performance metrics.

TABLE VII
PERFORMANCE COMPARISON AMONG MODELS

Model	Accuracy	Precision	Recall	F1
PSO-CNN	95.6	95.6	95.7	95.5
MRFO-CNN	95.7	95.7	95.4	95.3
LSTM	95.8	95.9	95.3	95.7
SVM	96.8	96.2	96.4	95.9
AOA-CNN	97.2	96.8	97.0	97.1
TBSMACNN	99.5	99.4	99.5	99.2

From Table VII, it is found that the accuracy of the TBSMA-CNN model is 99.5%, and the accuracy of other models is 95.6%, 95.7%, 95.8%, 96.8%, and 97.2%, respectively. The accuracy of the model proposed in this paper is better than that of other models. The CNN model is optimized by meta-heuristic algorithms, and it also outperforms other models in all other evaluation criteria.

In the situation classification model, the F1-score index of the model is 3.5% and 3.3% higher than that of the LSTM and the SVM, respectively, which indicates that the multidimensional feature space obtained after data preprocessing is more suitable for the CNN model given the high-dimensional and complex characteristics of network flow. In the comparison of meta-heuristic optimization algorithms, the F1-scor of the model is 3.7%, 3.9%, and 2.1% higher than the PSO, the MRFO, and the AOA model, respectively. It is proved that for the optimal hyperparameter configuration of the CNN model, the TBSMA algorithm outperforms these three optimization algorithms, which can provide a good network structure for the CNN model, reduce the computational complexity, balance the algorithms' exploration. It also proves that the CNN model can better deal with high-dimensional image data.

IV. Network Security Situation Assessment

A. Quantification of Situational Indicators

The situational assessment uses the CVSS system and assessment algorithms to quantify network attacks. It then determines and displays the current security development state based on the quantified values.

Table VIII
Rating factor of attack types in CIC-IDS2017 dataset

Type of attack	Grade point value
Port scanning attack	0.1
BForce	0.2
DoS	0.3
Bot	0.4
Web Attack	0.5
DDoS	0.6
Internal Infiltration	0.7

According to the CIC-IDS2017 dataset standard and concerning the CVSS system, the attack types are quantified in terms of situation indicators, and the impact value of each attack type is obtained, as shown in Table VIII. We mainly focus on the availability, integrity, and confidentiality of the impact metrics and comprehensively calculate the scoring level according to the level of the metrics.

The network security situation assessment quantifies the indicators to obtain the situation assessment value, which is used to determine the current network security status against the network security situation level rating scale to achieve real-time monitoring of the network situation. The network security situation level scale is shown in Table IX.

Table IX
Rating scale of network security situation

Rating	Score	Instructions
safety	0.0-0.2	running normally
lower risk	0.2-0.4	light impact
medium risk	0.4-0.75	more serious threat
high risk	0.75-0.9	higher threat
grave danger	0.9-1.0	network is untrustworthy

B. The TBSMA-CNN Model Situation Assessment

In this paper, the attack types are divided into real proportions to simulate the real situation. 20 sets of equal amounts of data are randomly selected from the test set, and different models are used to conduct situation assessment experiments. Then, situation metrics are quantified to obtain the impact score of each attack type in the CIC-IDS2017 dataset. The results of the situation assessment experiments and the impact scores of the attack types are combined to obtain the network security situation values.

Let us extract a total set of data. There are a total of categories of data in the network flow, with regular flow designated by label one and the remaining types of attacks identified by their corresponding quantities of occurrences. The attack impact value is determined using the attack type rank score, as stated in Eq. (13).

$$I_m = \sum_{i=2}^{n} (N_i \times A_i) \tag{13}$$

Eq.(13) indicates the rank score for each attack type. The standard flow impact value is shown in Eq. (14), and the network security situation value is calculated in Eq. (15).

$$B_m = \frac{N_1}{N_m} \tag{14}$$

$$S_m = (B_m \times I_m) / \sum_{i=2}^{n} N_i \tag{15}$$

Table X
Results of the TBSMACNN model about situation assessment

Test set number	The actual situation		TBSMA-CNN model assessed value	
	Actual situational value	Situational level	Assessed situational value	Situational level
1	0.585	medium risk	0.649	medium risk
2	0.795	high risk	0.807	high risk
5	0.606	medium risk	0.638	medium risk
6	0.692	medium risk	0.653	medium risk
9	0.830	high risk	0.862	high risk
14	0.644	medium risk	0.647	medium risk
15	0.196	safety	0.020	safety
16	0.650	medium risk	0.743	medium risk
20	0.747	medium risk	0.789	high risk

Finally, the network security situation values determine the current network security status. Table X compares the assessed and actual circumstance values. Table X displays a selection of data from the experiment's 20 test groups. The table shows the evaluated and absolute scenario values and the situation classes based on those values. From Table X, it can be seen that the evaluated situation values of the TBSMA-CNN model are close to the actual situation values, and the evaluated situation grades of the majority of the test groups are the same as the real situation. When the actual situation values are close to the boundary of the current grade range, a few assessed situations will be somewhat deviated. However, the overall fit is good and meets the expectations of the network security situation assessment.

C. Experimental Comparison and Analysis

This paper's comparative experiments are primarily split into two sections to confirm the viability of the TBSMA-CNN model from two perspectives. The original CNN model, the LSTM model, and the SVM model are examples of machine learning models; the AOA algorithm, the MRFO method, the PSO algorithm, and the original SMA algorithm are examples of meta-heuristic algorithms. To verify the feasibility of the TBSMA-CNN model proposed in this paper and the necessity of improving the SMA algorithm for the super-participant optimization work, mainstream machine learning models are used for comparison experiments. All models are subjected to the same data preprocessing operations, and the results of the resulting network security situation assessment are shown in Fig. 5.

As can be seen from the folded lines in Fig. 5, the trend of the situation values of the four sets of machine learning models is broadly similar to the actual problem, and the evaluated situation values of the TBSMA-CNN model used in this paper are more closely aligned with the actual situation values.

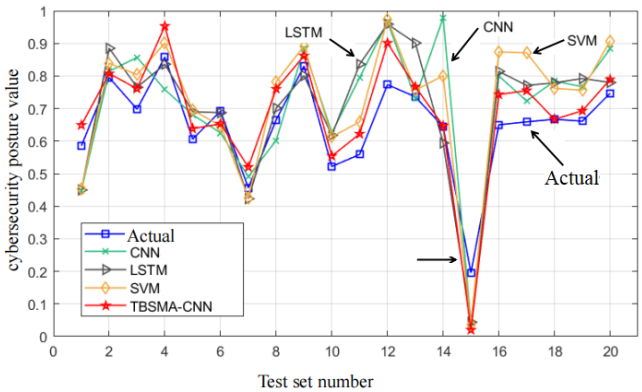


Fig. 5. Comparison of situational value in machine learning models

Analyzed from the perspective of the accuracy of situational hierarchy classification, the overall hierarchy classification of the TBSMA-CNN model in this paper is almost the same. The original CNN, LSTM, and SVM models have the problem of incorrectly dividing the grades. For example, in the 14th group of data, only the model of this paper and the LSTM model are consistent with the real situation, and the assessed situational grade is medium. In contrast, the SVM model assesses it as high, and the CNN model assesses it as severe. In the 18th data group, the remaining three models are assessed as high danger, but the TBSMA-CNN model is consistent with the actual situation, and the situation grade is low danger.

V. Conclusion

This paper suggests a slimy bacteria algorithm based on tangent flight operator and probability distribution, aiming to address the issues of low computational accuracy, poor convergence performance, slow efficiency on the benchmark function, and rapidly falling into the slimy bacteria algorithm's local optimum. The TBSMA algorithm is introduced into the convolutional neural network, relying on the TBSMA algorithm to optimize CNN hyperparameters automatically. Finding CNN with optimal performance through the TBSMA algorithm saves time and resource allocation problems and helps improve the detection performance of situational extraction and situational assessment. The actual network situation is simulated for network security assessment, and the test set data is extracted proportionally. The network attack impact value is obtained by evaluating the experimental results and combined with the CVSS system to calculate the network security situation value further. Its assessment results are compared with the classical model in detail to verify the model's performance. The experimental results confirm that the proposed model in this paper outperforms other models in assessing the situational value and situational level.

In terms of parameter optimization, although the improved TBSMA algorithm in this paper is better than the other comparative algorithms, it is still necessary to consider the problem of its parameters affecting convergence. Therefore, the SMA algorithm will be further improved for better detection performance. Regarding network security situation assessment, the accuracy of model assessment needs to be further enhanced to address the problem of poor judgment performance of critical situation values. It is necessary to improve the research of network security situation assessment systems to achieve situational prediction work for network security.

References

[1] Wen Z C, Peng L H, and Wan W Q, “An Algorithm for Network Security Situation Assessment Based on Deep Learning”, International Journal of Pattern Recognition and Artificial Intelligence, vol.37, no.2, pp1-18, 2023.

[2] Cheng M, Li S, and Wang Y H, et al., “A New Model for Network Security Situation Assessment of the Industrial Internet”, Computers, Materials and Continua, vol.75, no.2, pp2527-2555, 2023.

[3] Yang H Y, Zeng R Y, and Xu G Q, et al., “A network security situation assessment method based on adversarial deep learning”, Applied Soft Computing, vol.102, pp107096, 2021.

[4] Lin Y, Wang J, and Tu Y, et al., “Time-Related Network Intrusion Detection Model: A Deep Learning Method”, IEEE Global Communications Conference, pp1-6, 2019.

[5] Yu Y H, “A network security situation assessment method based on fusion model”, Discover Applied Sciences, vol.6, no.3, pp97-105, 2024.

[6] Wang J H, Shan Z L, and Tan H S, et al., “Networks Security Situation Assessment Base on Genetic Optimized PNN Neural Network”, Computer Science, vol.48, no.6, pp338-342, 2021.

[7] Zhang R, Pan Z H, and Yin Y F, “A Model of Network Security Situation Assessment Based on BPNN Optimized by SAA-SSA”, International Journal of Digital Crime and Forensics, vol.14, no.2, pp1-18, 2022.

[8] Yang H Y, Zhang Z X, and Zhang L, "Network security situation Assessments with parallel feature extraction and an improved BiGRU", J Tsinghua Univ(Sci & Technology), vol.62, no.5, pp842-848, 2022.
[9] Mirjalili S, and Lewis A, "The Whale Optimization Algorithm", Advances in engineering software, vol.95, pp51-67, 2016.
[10] Zhao W, Zhang Z, and Wang L, "Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications", Engineering Applications of Artificial Intelligence, vol.87, pp103300, 2020.
[11] Karaboga D, and Ozturk C, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", Application Soft Computing, vol.11, no.1, pp652-657, 2011.
[12] Pan W T, "A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example", Knowledge-Based Systems, vol.26, pp69-74, 2012.
[13] Mirjalili S, Mirjalili S M, and Lewis A, "Grey Wolf Optimizer", Advances in Engineering Software, vol.69, pp46-61, 2014.
[14] Li S, Chen H, and Wang M, et al., "Slime mould algorithm: A new method for stochastic optimization", Future Generation Computer Systems, vol.111, pp300-323, 2020.
[15] Chauhan S, Vashishtha G, and Kumar A, "A symbiosis of arithmetic optimizer with slime mould algorithm for improving global optimization and conventional design problem", The Journal of Supercomputing, vol.78, no.5, pp6234-6274, 2022.
[16] Chen Z, and Liu W, "An Efficient Parameter Adaptive Support Vector Regression Using K-Means Clustering and Chaotic Slime Mould Algorithm", IEEE Access, vol.8, pp156851-156862, 2020.
[17] Naik M K, Panda R, and Abraham A, "Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm", Computer and Information Sciences, vol.34, no.7, pp4524-4536, 2022.
[18] Ewees A, Abualigah L, and Yousri D, et al., "Improved Slime Mould Algorithm based on Firefy Algorithm for feature selection: A case study on QSAR model", Engineering with Computers, vol.38, no.3, pp2407-2421, 2022.
[19] Canadian Institute for Cybersecurity, "Intrusion detection evaluation dataset (CIC-IDS2017)", https://www.unb.ca/cic/datasets/ids-2017.html, 2024.
[20] Qiaochu Sun, Hong Dai, and Yao Xu, et al., "A Novel Network Intrusion Detection Method Based on DSAE-PSOCNN Model", Engineering Letters, vol.30, no.4, pp1306-1315, 2022.
[21] A. Layeb, "The Tangent Search Algorithm for Solving Optimization Problems", Neural Computing and Applications, vol.34, no.11, pp8853-8884, 2022.
[22] F. Zheng, and G. Liu, "An Adaptive Sinusoidal-Disturbance-Strategy Sparrow Search Algorithm and Its Application", Sensors, https://doi.org/10.3390/s22228787, 2022.