International Journal of Digital Crime and Forensics
Volume 14 • Issue 2

A Model of Network Security Situation Assessment Based on BPNN Optimized by SAA-SSA

Ran Zhang, Zhengzhou University of Light Industry, China
Zhihan Pan, Zhengzhou University of Light Industry, China*
Yifeng Yin, Zhengzhou University of Light Industry, China
Zengyu Cai, Zhengzhou University of Light Industry, China

ABSTRACT

In order to address the problem that the accuracy and convergence of current network security situation assessment models need to be improved, a model of network security situation assessment based on SAA-SSA-BPNN is proposed. Comparative experimental results show that this assessment model has higher accuracy and faster convergence than other situation assessment models based on improved BP neural network.

KEYWORDS

BP Neural Network, Local Extreme Value, Network Security, Simulated Annealing Algorithm, Situation Assessment, Sparrow Search Algorithm, Threshold, Weight

INTRODUCTION

In the wake of the speedy progress of Internet techniques, cyberspace security issues have become increasingly complicated. Network attacks occur frequently, and the scale is expanding. The number of public security threats on the Internet is increasing, and global cybersecurity is facing severe challenges. Traditional cybersecurity countermeasures can no longer satisfy complex network security requirements, and more modern technology and methods should be adopted to stop the emergence of security events in nework. In these circumstances, a new security technology known as awareness of the network security situation emerged.

In previous work, we used the sparrow search algorithm (SSA) to optimize the back propagation neural network (BPNN) and applied it to network security situation assessment (Zhang, Pan & Yin, 2021). Compared with other assessment models based on BPNN, this model raises the efficiency and accuracy of situation assessment to a certain extent, but the SSA algorithm often falls into a local optimum due to fast convergence. To address this problem, a simulated annealing algorithm (SAA) is introduced to improve SSA, an assessment network security situation model based on BPNN improved by SAA-SSA is proposed, and the accuracy and validity of the model are proven by experiments.

DOI: 10.4018/IJDCF.302877 *Corresponding Author

BACKGROUND

The definition of awareness of the network security situation was first proposed by Bass (1999) and contains three stages: perception, assessment and prediction. The network security situation assessment system aims to integrate and analyze situation factors and data information extracted from the network, model and assess the present network security situation, obtain the values of situation through the assessment model, dynamically represent the present operating state and the overall severity of threats of the network system, and predict and forewarn its development trend to provide decision support for network security management. As an essential component of the new technology of next generation network security and new cybersecurity defense system, assessment of network security situation means a lot in research and practical value.

Since a definition called the awareness of network security situation was proposed, many specialists have done extensive studies about the technology of awareness on network security situation. (Tao, Kong, Zhao, Cheng & Wang, 2020) used the stacked self-coding network to decrease the dimension of situation data to decrease the data storage overhead and enhance the operation efficiency. (Wang, Zhao & Li, 2020) used fuzzy c-means, hybrid hierarchical genetic algorithm and least square method to optimize the parameters and structure of traditional RBF to assess the security situation of network. (Chang, Tian, Zhang, Qian & Hu, 2021) aimed at the problem that single-point network data cannot effectively analyze network security and introduced a multisource heterogeneous data fusion strategy. (Feng, Wang, Ma & Li, 2011) designed the empirical function of evidence theory by using arctangent and correction functions and applied evidence theory to the assessment of network security situations. Smith (2012) developed and improved security detection tools to address increasingly complex Internet attacks. (Yegneswaran, Barford & Paxson, 2005) proposed a situation assessment approach based on honeynets, which builds the safety condition curve for analyzing the current network security situation; however, the curve cannot display evident effects at every attack time; thus, it is not comprehensive enough. Kotenko & Novikova (2014) used visualization technology to show a group of security indices, which were employed to evaluate the network security situation and the efficiency of the network safeguard mechanism. Although these situation assessment models or algorithms have improved the original technology in part, there is still room for optimization in the accuracy of assessment and the convergence of the algorithm.

TO BUILD A NETWORK SECURITY SITUATION INDICES SYSTEM

Whether the network security situation index system is scientific and reasonable directly affects the ultimate result of network security situation assessment, so it will be very important to construct a rational and objective situation index system according to certain principles (Wang, Zhang, Fu & Chen, 2007) before assessing the network security situation.

Principles for Building the Indices System of Network Security Situation

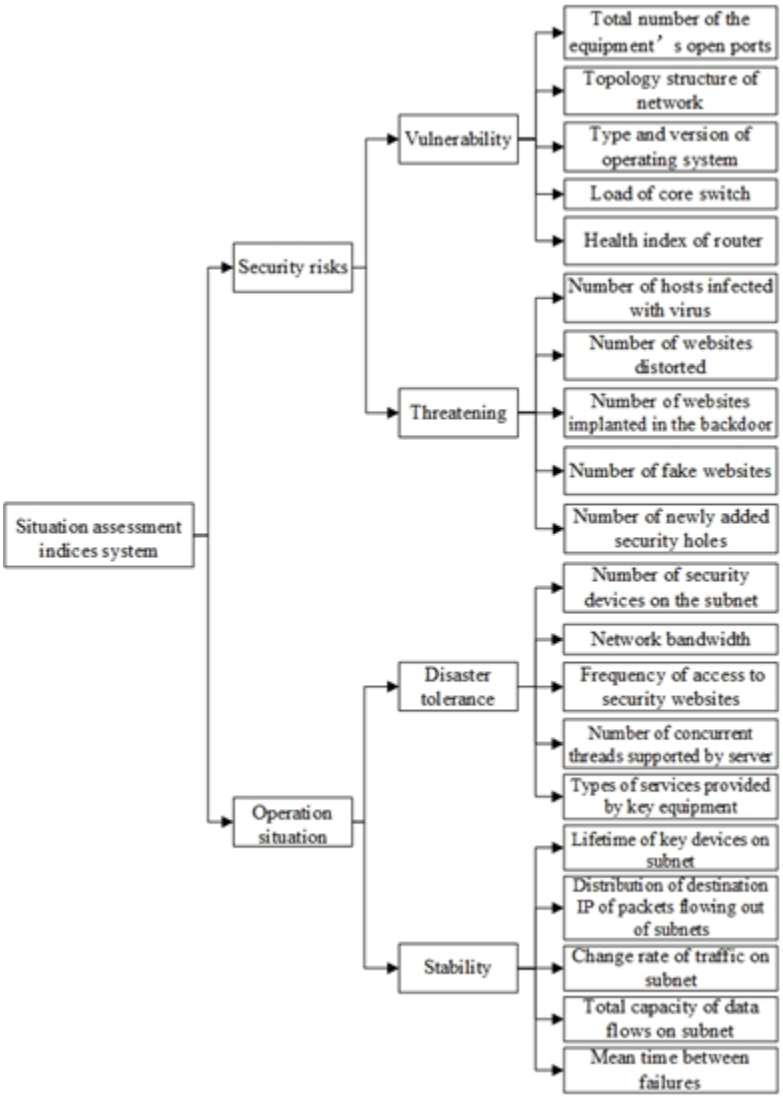
Many reasons affect the network situation, and they restrict each other. Therefore, it is quite complex to establish an objective and reasonable situation index system. Certain principles need to be followed, appropriate methods and steps need to be adopted, and statistical analysis, comprehensive trade-off and induction need to be repeated to construct a scientific and reasonable situation indices system. The establishment of a network security situation index system mainly follows the following principles: the principle of similarity, the principle of hierarchy and the principle of dynamic-static combination. The principle of similarity requires that indices with similar characteristics be considered as one category, such as the distribution of data packets and the distribution of packet size. The hierarchical principle requires that the indices that have different degrees of impacts on the network be considered separately, such as those indices for subnets and those for macro networks. The dynamic-static combination principle requires that indices with different characteristics, such as traffic and network topology, be considered separately.

The Construction of Network Security Situation Indices System

The network security situation index system describes the network security situation as a whole. Under the construction rules of the situation index system, a tree hierarchical security situation index system is constructed (Zhang, Pan & Yin, 2021), as shown in Figure 1. There are two first-level indices, four secondary indices and twenty tertiary indices in the indices system, which, respectively describe the security risks and operating status characteristics of the network system from four aspects: vulnerability, threatening, disaster tolerance and stability.

Based on this index system, this paper assesses security threats to the network system. Five tertiary indices under the threat index are used, which are the number of hosts infected with virus, the number of websites distorted, the number of websites implanted in the backdoor, the number of fake websites and the number of newly added security holes, and each index is given equal weight.

Figure 1. Network security situation indices system



The experimental data were collected from the Weekly Report of Network Security Information and Dynamics of the Chinese National Internet Emergency Center.

NETWORK SECURITY SITUATION ASSESSMENT MODEL BASED ON SAA-SSA-BPNN

We introduce a sparrow search algorithm (SSA) optimized by a simulated annealing algorithm (SAA) to improve the BP neural network (BPNN) and apply it in the assessment of network security situations. First, the corresponding data are acquired and preprocessed according to the constructed indices system, and then the BPNN improved by the SAA-SSA algorithm is trained. Finally, the generated evaluation model is used for situation assessment, and its output results are analyzed. The whole situation assessment procedure is composed of three parts, and the specific situation assessment model can be seen from Figure 2.

Data Collection and Processing

Establish an indices system and collect 370 pieces of data from July 2012 to December 2021 from Chinese National Internet Emergency Center (CNCERT) as experimental data, normalize the data and take 360 pieces of data as training dataset and 10 pieces as test dataset.

Situation Assessment Model Generation

The structure of the BPNN is decided by the features of the input–output data. The SAA-SSA algorithm is adopted to find the optimal initial weights and thresholds for the BPNN. After assigning the optimal parameter combination to the BPNN, the training data are input to train it, and finally, the SAA-SSA-BPNN situation assessment model with assessment ability is obtained.

ASSESSMENT AND RESULT ANALYSIS

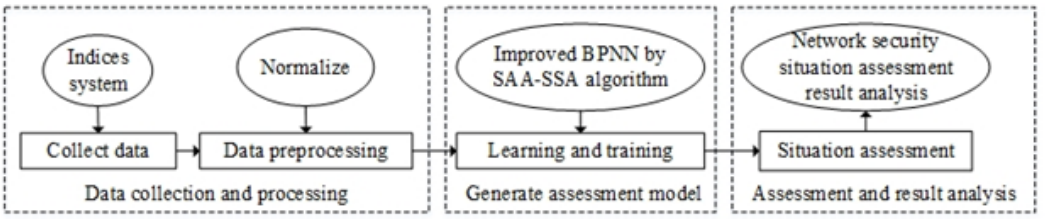
The test data are input into the SAA-SSA-BPNN situation assessment model with assessment capability to obtain situation assessment values. Based on the classification at the situation level, the results of situation assessment are analyzed, and the situation and level of present network security are judged, enabling administrators to fully master the present network security situation and take measures in time.

ASSESSMENT OF NETWORK SECURITY SITUATION BASED ON SAA-SSA-BPNN

BP Neural Network

The back propagation neural network (BPNN) is one of the most typical multilayer feedforward neural networks, which was proposed by Rumelhart and McClelland in 1986 and trained by the error back

Figure 2. Network Security Situation assessment model



propagation algorithm (Rumelhart, Hinton & Williams, 1986). BPNN reduces errors by continuously returning error signals from the back layer to the front layer to adjust the size of weights and thresholds, stops learning and training unless the global error falls below the expected error or when the learning times achieve the maximum. BPNN has a simple structure, which is particularly applicable to resolve complicated internal mechanism problems and has powerful self-learning capability. The specific topology of BPNN is shown from Figure 3.

In Figure 3, $X_1, X_2, \cdots, X_n$ are the input data of the BPNN, Y is the output value of the BPNN, ω_{ij} represents a connection weight between the input and hidden layers of the BPNN, and ω_{jk} stands for a connection weight between the hidden and output layers of the BPNN. There is one output value in the next experiment, but in different experiments, the number of output values may be different, so it is necessary to design the topology of the BPNN according to the actual situation.

The training steps of the BPNN are as follows (MATLAB Chinese Forum, 2010):

Step 1: Initialize BPNN. According to the characteristics of the dataset to decide n, l and m, which separately represent the number of input layer nodes, the number of hidden layer nodes and the number of output layer nodes. Then, the rands() function is used to initialize ω_{ij} and ω_{jk}, the hidden layer threshold a and the output layer threshold b, and the learning rate η and the neuron excitation function sigmoid are set.

Step 2: Calculate the hidden layer output based on the input data, ω_{ij} and a, and mark it as H:

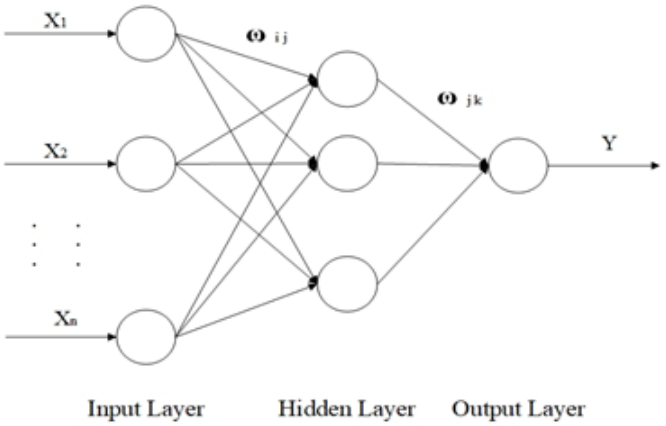
$$D_j = \sum_{i=1}^{n} \omega_{ij} x_i - a_j, j = 1,2,3,\ldots,l \tag{1}$$

$$H_j = f\left(D_j\right) \tag{2}$$

In formula (2), the hidden layer excitation function is f. The functions selected in this paper are as follows:

$$f\left(x\right) = \frac{1}{1+e^{-x}} \tag{3}$$

Figure 3. BPNN topology



Step 3: Calculate the output of the output layer based on, H, ω_{jk} and b, and mark it as Y:

$$Y_k = \sum_{j=i}^{l} H_j \omega_{jk} - b_k,\ k = 1,2,...,mY \tag{4}$$

Step 4: Calculate the prediction error based on and the expected output S and mark it as e:

$$e_k = S_k - Y_k \tag{5}$$

Step 5: Renew the weights. Renew ω_{ij} and ω_{jk} based on e:

$$\omega_{ij} = \omega_{ij} + \eta H_j \left(1 - H_j\right) \sum_{k=1}^{m} \omega_{ij} e_k,\ i = 1,2,...,n \tag{6}$$

$$\omega_{jk} = \omega_{jk} + \eta H_j e_k \tag{7}$$

Step 6: Renew the thresholds. Renew a and b based on e:

$$a_j = a_j + \eta H_j \left(1 - H_j\right) \sum_{k=1}^{m} \omega_{ij} e_k \tag{8}$$

$$b_k = b_k + e_k \tag{9}$$

Step 7: Judge if the iteration is completed; otherwise, return to **Step 2**.

However, the initial weights and thresholds of BPNN are generated by the rands () function, it performs too many iterations, the convergence speed is low, and it cannot guarantee convergence to the global optimum every iteration. For this reason, intelligent optimization algorithms are usually introduced to solve this problem; for example, the genetic algorithm (GA) (Gao, Luo, Wang, Yang, Sun & Wang, 2021) or particle swarm optimization (PSO) (Duan, 2019) are used to optimize the BPNN, but the convergence speed of these two algorithms is not ideal. Therefore, we adopt the sparrow search algorithm (SSA) for improving BPNN, which uses the fast convergence speed and strong local search ability of SSA. However, the weakness of SSA is its poor ability to search globally and break the local optimum state, whereas the simulated annealing algorithm (SAA) has a strong ability to search globally and break the local optimum state. Therefore, SAA should be introduced to optimize the SSA algorithm, the optimized SSA is used to improve the BPNN, and the SAA-SSA-BPNN is applied to network security situation assessment.

Sparrow Search Algorithm

The sparrow search algorithm (SSA) has become a newly developed intelligence optimization technique that was proposed in recent years. Inspired by the characteristics of sparrow predation, the algorithm divides sparrows in the process of predation into two roles: discoverer and participant. The discoverers are in charge of searching for food for sparrow groups and giving the whole colony a foraging direction, whereas the participants follow the discoverers and compete with them to obtain food (Xue, 2020). According to the relationship between them and the response of sparrows when they meet the predator, a mathematical model is established, which is an efficient intelligence

optimization algorithm. The advantage of the SSA algorithm is that it has good stability and fast convergence speed. The algorithm includes the steps as follows:

Step 1: The group of sparrows needs to be initialized, and the related parameters of the sparrow population need to be defined. We set n as the sparrow population scale, d as the variable dimension, f is the individual fitness value and t as the present iteration of the number. The quantity of sparrows can be represented as the following matrix:

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,d} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,d} \end{bmatrix} \tag{10}$$

Step 2: According to the fitness function, the fitness value f_i of each sparrow is calculated, and the present optimal fitness value f_g, the worst fitness value f_w and their counterpart positions X_{best} and X_{worst} are selected by sorting the fitness values.

Step 3: In the SSA, due to the initiative of discoverers, they could gain a wider foraging area and greater fitness value. If the alarm value R_2 is less than the safety value ST, then the surrounding environment is safe at this time. If the alarm value R_2 is higher than the safety value ST, then some alert sparrows have recognized that they are in dangerous situations, and all sparrows should quickly move to a secure place for foraging. The position update of the discoverers can be represented by formula (11):

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\dfrac{i}{\alpha \cdot iter_{\max}}()_2\right) & \\ X_{i,j}^{t} + Q \cdot L & R_2 \geq ST \end{cases} \tag{11}$$

In the above formula, $j = 1, 2, 3, \cdots, d$; $iter_{max}$ indicates the maximum number of iterations; $X_{i,j}$ represents the position information of the i-th sparrow in the j-th dimension; α indicates a randomly selected number, and its value range is (0,1]; Q expresses a randomly distributed number subject to a normal distribution; and L represents a matrix of $1 \times d$ in which every component is one.

Participants always monitor the discoverers throughout the whole process of foragers. When they realize that the discoverers have seen something better, they will fly away instantly to snatch new food from the discoverers. If they win, they will immediately obtain the discoverers' food; otherwise, they will repeat the above operations. As $i > n/2$, this means that f_i of the i -th participant is low. At this time, the sparrow is very hungry, so it has to fly elsewhere to find food and obtain more energy. The position of the participants can be updated according to formula (12):

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{X_{worst} - X_{i,j}^{t}}{i^2}\right) & i > \dfrac{n}{2} \\ X_{p}^{t+1} + \left|X_{i,j}^{t} - X_{p}^{t+1}\right| \cdot A^{+} \cdot L & otherwise \end{cases} \tag{12}$$

Within formula (12), X_p expresses the optimal position that discoverers occupy; A stands for a $1 \times d$ matrix in which every component is given a random value of 1 or -1, and $A^+ = A^T(AA^T)^{-1}$.

During sparrow foraging, if predators appear, those sparrows in the outermost part of the population will first realize the existence of danger. When they are aware of the danger, that is, when $f_i > f_g$, sparrows in the outermost part of these groups will attempt to fly to a relatively safe foraging area. Otherwise, it indicates the sparrows in the group center realize dangerous that they have to fly for safety to reduce the risk of their arrest. According to formula (13), we can upgrade the position of sparrow that is concerned about danger:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \left| X_{i.j}^t - X_{best}^t \right| & f_i > f_g \\ X_{i,j}^t + K \left(\dfrac{\left| X_{i,j}^t - X_{worst}^t \right|}{\left(f_i - f_w \right) + \varepsilon} \right) & f_i = f_g \end{cases} \tag{13}$$

In the above formula, β expresses the step length control parameter; K indicates a randomly selected number in [-1,1]; ε is the smallest constant:

Step 4: Get the present optimal value. Executing the update operation when the present optimal value exceeds the optimal value of the earlier iteration, otherwise no updating operation is performed, and the iterations will continue until the requirements are satisfied, resulting in obtaining the optimal fitness value f_g and its counterpart position X_{best}.

Although SSA is strong in local search and fast in convergence, it is weak in searching globally and breaking the local optimum. Therefore, in this paper, a simulated annealing algorithm (SAA) is introduced to improve SSA shortcomings.

Simulated Annealing Algorithm

In 1953, the simulated annealing algorithm (SAA) was proposed by Metropolis (Metropolis, 1953). However, SAA is generally not used alone but is usually used for combinatorial optimization. SAA's principle is to simulate the annealing and cooling process of high-temperature solids, which goes through three steps: heating, waiting and cooling. Generally, in practical applications, the system energy is expressed as f, and the temperature of the system is expressed as a control parameter T. f decreases as the internal energy decreases with the temperature; when the temperature drops to normal temperature, the internal energy decreases to a minimum. The system state $T = 0$ corresponds to the global optimal solution of the optimization problem (Zhang, Ye & Hu, 2004). The SAA controls the temperature change process according to the Metropolis criterion, which has a strong global search ability, can accept inferior solutions with a certain probability, and effectively prevents the algorithm from strucking at local extrema (Wang, Zhao & Li, 2020). The algorithm steps are as below:

Step 1: Initialize annealing temperature.

Initializing annealing temperature T_k (let k =0), T_k is expressed as follows:

$$T_k = -\frac{f_g}{\ln(\alpha)} \tag{14}$$

where f_g represents the global optimal fitness value, α stands for the initial acceptance probability and its value interval is [0.2,0.5].

Step 2: Calculate the annealing rate:

$$T_{t+1} = \gamma T_t \tag{15}$$

where γ is the cooling rate and t is the iteration number.

Step 3: Judge according to Metropolis acceptance criteria:

$$P = \begin{cases} 1 & \Delta f < 0 \\ \exp(-\dfrac{\Delta f}{T_k}) & \Delta f \geq 0 \end{cases} \tag{16}$$

$$\Delta f = f_i - f_g \tag{17}$$

where P is the sudden jump probability and f_i in formula (17) is the current individual fitness value. If $\Delta f < 0$, receive the new answer with probability 1; otherwise, receive the new answer with probability $\exp(-\Delta f / T_k)$.

Network Security Situation Assessment Algorithm Based on SAA-SSA-BPNN

The sparrow search algorithm (SSA) has advantages in local search ability and convergence speed, but its ability in global searching and breaking local optima is weak, while the simulated annealing algorithm (SAA) has a strong ability in global searching and breaking the local optimal state. Therefore, in this paper, we use SAA to overcome the shortcomings of SSA, and the improved SSA is adopted to find the optimal initial weights and thresholds for BPNN.

To illustrate the SAA-SSA-BPNN network security situation assessment algorithm more clearly, the algorithm steps for finding the optimal initial weights and thresholds for BPNN are represented as Algorithm 1 below. Before implementing this algorithm, it is imperative to establish the security indices system and preprocess the situation data used in the experiment.

Every individual in the group includes all the weights and thresholds of the network. Therefore, the optimal individual corresponding to X_{best} and f_g obtained by the SAA-SSA algorithm is assigned to the BPNN. Next, the preprocessed training data are put into the improved BPNN model for training according to formulas (1)-(9). Then, the SAA-SSA-BPNN situation assessment model with situation assessment capabilities is obtained, and finally, the test data are put into the model to obtain the assessment values of the situation. Administrators analyze the current network security situation in line with the obtained situation values and the network security situation assessment level table. The complete SAA-SSA-BPNN algorithm process is shown in Figure 4.

EXPERIMENT AND RESULT ANALYSIS

We adopt a simulated annealing algorithm (SAA) to overcome the shortcomings of the sparrow search algorithm (SSA) in this experiment and then use the improved SSA to modify the BP neural network (BPNN), thereby increasing the convergence speed and efficiency of the BPNN and avoiding the SSA-BPNN algorithm from falling into local optimization. To facilitate the analysis of the network

Algorithm 1. Find the optimal values for BPNN

Input:
$Iter_{max}$: the maximum number of iterations
ND: the number of the discoverers
SD: the number of the sparrows conscious about dangers
R_2: the warning value
Establish fitness function $f(x)$, where variable $x = (x_1, x_2, x_3, \ldots, x_d)$;
Initialize a sparrow group with n sparrows and set the parameters associated with the sparrow population;
Output: X_{best}, f_g
1: **while** the $Iter_{max}$ *is not met* **do**
2: Sort the fitness values to search out the current optimal individual, the current worst individual and their corresponding positions;
3: R_2 = rands (1);
4: **for** $i = 1$: *ND*
5: Using formula (11) update the sparrow's position;
6: **end for**
7: **for** $i = (ND+1)$: n
8: Using formula (12) update the sparrow's position;
9: **end for**
10: **for** $i = 1$: *SD*
11: Using formula (13) update the sparrow's position;
12: **end for**
13: Update the global optimal position according to Metropolis criterion;
14: Cooling treatment;
15: If the new position is preferable to the previous iteration result, the update operation is performed;
16: $t = t + 1$;
17: **end while**
18: **return** X_{best}, f_g

situation, the network security situation assessment level is quantified into a specific situation value range and divided into five levels: excellent, good, medium, poor and dangerous, as shown in Table 1.

SIMULATION ENVIRONMENT

This experiment takes the security threat of the network system as the assessment target and verifies the effectiveness and feasibility of the SAA-SSA-BPNN model. It collects five three-level index elements under the threat index in the indices system established before: the number of hosts infected with virus, the number of websites distorted, the number of websites implanted in the backdoor, the number of fake websites and the number of newly added security holes. Here, the weight of each index is set to be equal. At present, most of the current research uses experimental data from the KDDCUP99 dataset or the attack dataset published by the HoneyNet project, but the KDDCUP99 dataset has a long history and too many redundant data, and the impact factors on the attack dataset are too single. Therefore, this experiment uses 370 groups of data from the Weekly Report of Network Security Information and Dynamics published by CNCERT from July 2012 to December 2021, and the experimental sample data are separated into two categories: a) 360 training samples data are designed to train the BPNN improved by SAA-SSA; b) 10 test samples data are designed to test the effectiveness of SAA-SSA-BPNN model. MATLAB R2019a is used for simulation. The hardware environment used a 1.8 GHz CPU and 8 GB memory, and the operating system was Windows 10.

DATA PREPROCESSING

Before the experiment, it was essential for us to normalize the data first. Usually, there are two ways of normalization: putting all the numbers in the [0,1] interval or changing the dimensional expression

Figure 4. Process diagram of SAA-SSA-BPNN

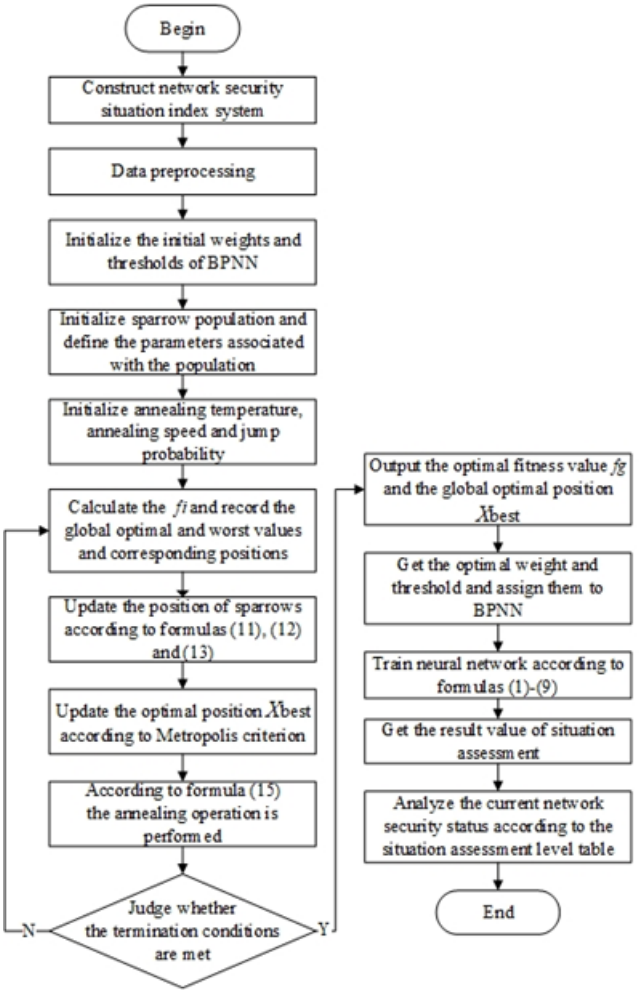


Table 1. Network security situation assessment level

Excellent	Good	Medium	Poor	Dangerous
[0,0.2]	(0.2,0.4]	(0.4,0.6]	(0.6,0.8]	(0.8,1]

into a dimensionless expression. Here, the first approach is used to normalize the experimental data to the [0,1] range according to formula (18):

$$Y_i = \frac{X_i - X_{min}}{Xmin_{max}} \tag{18}$$

Within formula (18), X_{min} is the minimum data, X_{max} is the maximum data in the experimental dataset, and Y_i is the final normalized result.

Determine the Structure and Parameters of BPNN

Setting the Number of Nodes

In BPNN, the dimension of I/O data decides the quantity of I/O layer nodes. As the dataset used in this experiment includes five input parameters and an output parameter, there are five nodes in the input layer and one node in the output layer. The quantity of hidden layer nodes is determined by many factors, including the number of input and output layer nodes and the complexity of practical problems. If there are too few nodes in the hidden layer, network training results are often unsatisfactory; if the nodes are numerous, network training time will increase. At present, the number of hidden layer nodes is generally decided by the trial and error method, with reference to the following three empirical formulas:

$$l < n - 1 \tag{19}$$

$$l < \sqrt{\left(m + n\right)} + a \tag{20}$$

$$l = \log_2 n \tag{21}$$

Within the above formulas, a represents a positive integer less than 10. Based on the dimension of input and output data and the trial and error method, we determine that there are 8 hidden layer nodes in this experiment.

Selection of Parameters

BPNN generally initializes weights and thresholds through the rands function. In this paper, the SSA optimized by SAA can be applied to search for the optimal weights and thresholds of the BPNN.

In the training process, the learning rate determines the convergence state of the BPNN, so we need to have an appropriate learning rate. In most experiments, the learning rate is generally set between 0.01 and 0.8, which is set to 0.1 in this paper.

Analysis of Experimental Results

Comparative Analysis of Assessment Results

A comparison between the ten situation assessment values obtained by the improved BPNN using GA, PSO, SSA and SAA-SSA with the ten situation values provided by the Chinese National Internet Emergency Center (CNCERT) is shown in Table 2. Among them, the situation value of the CNCERT is quantified by extracting the intermediate value of the corresponding situation assessment level value range.

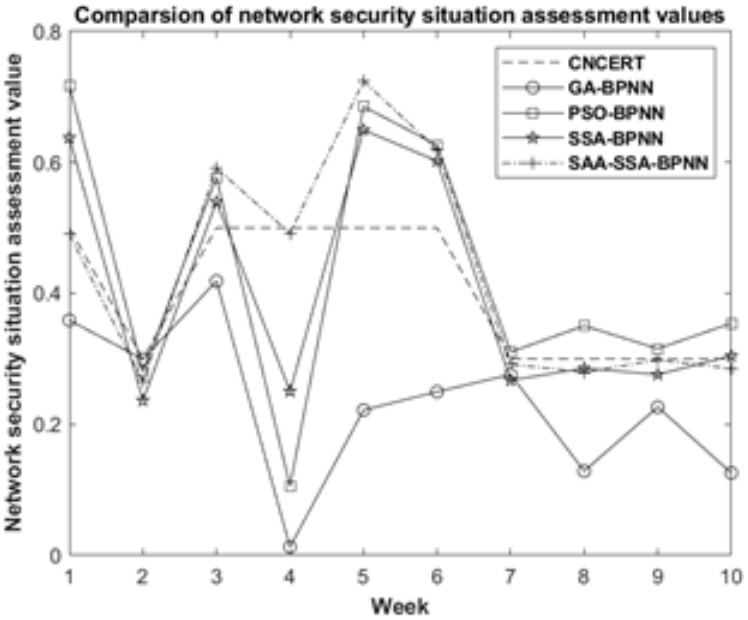
To more intuitively analyze the situation assessment results, the assessment values in the above table are shown in a line chart, as shown in Figure 5. As shown in Figure 5, the trends of the situation assessment value curves for the four evaluation models are approximately the same. They all fluctuate in the 2nd, 3rd, 4th, 5th and 7th weeks. Except for the assessment values of SAA-SSA-BPNN, the assessment values of the other three assessment models in the 4th week are far from CNCERT.

Next, the lines of the four models are analyzed one by one. The assessment value of the GA-BPNN model is close to that of CNCERT in the 2nd and 7th weeks and unstable at other time points. The assessment value of the PSO-BPNN model is close to that of CNCERT in the 2nd, 7th and 9th weeks, but the results at other time points fluctuate greatly. The SSA-BPNN assessment model and

Table 2. Assessment data analysis

CNCERT Assessment Value	GA-BPNN Assessment Value	PSO-BPNN Assessment Value	SSA-BPNN Assessment Value	SAA-SSA-BPNN Assessment Value
0.5	0.3590	0.7173	0.6377	0.4908
0.3	0.2993	0.2809	0.2363	0.2651
0.5	0.4187	0.5767	0.5397	0.5912
0.5	0.0128	0.1057	0.2505	0.4917
0.5	0.2217	0.6853	0.6495	0.7242
0.5	0.2497	0.6263	0.6020	0.6197
0.3	0.2760	0.3104	0.2673	0.2918
0.3	0.1286	0.3505	0.2855	0.2794
0.3	0.2264	0.3155	0.2760	0.2983
0.3	0.1257	0.3543	0.3045	0.2852

Figure 5. Comparison of assessment results of four models



PSO-BPNN assessment model have basically the same trend, but the assessment values of the SSA-BPNN model in the first three weeks are lower than those of the PSO-BPNN assessment model, and its assessment values in the fourth to tenth weeks are closer to CNCERT than those of the PSO-BPNN. Figure 5 shows that the assessment value of the SAA-SSA-BPNN assessment model only fluctuates slightly in the 3rd, 5th and 6th weeks and is basically consistent with CNCERTs at other times. This indicates that the assessment accuracy of the SAA-SSA-BPNN assessment model is the highest among the four models.

Table 3 shows the levels of situation obtained by applying the four assessment models, GA-BPNN, PSO-BPNN, SSA-BPNN and SAA-SSA-BPNN, and compares them with the levels of CNCERT. From Table 3, it shows that the situation levels obtained by GA-BPNN assessment model for six weeks is different from CNCERT, the PSO-BPNN model have four situation levels that are inconsistent with CNCERT, and the four situation levels from the SSA-BPNN model are inconsistent with CNCERT. However, the SAA-SSA-BPNN assessment model only has two situation levels that are inconsistent with the CNCERT. Therefore, the SAA-SSA-BPNN assessment model can most realistically describe the real network security situation.

Error Analysis

Table 4 shows the comparison of absolute error values between the ten situation assessment results of GA-BPNN, PSO-BPNN, SSA-BPNN and SAA-SSA-BPNN and the situation values from CNCERT.

Here, the absolute error values in Table 4 are represented by a more intuitive line chart, as shown in Figure 6. By analyzing Figure 6, it can be found that only the absolute error line of the SAA-SSA-BPNN situation assessment model has the smallest fluctuation range. Except for the 3rd, 5th and 6th data points, other data points are very close to the 0 error value guide line. Therefore, the SAA-SSA-BPNN assessment model is closest to the situation value of CNCERT in the four models.

To further validate the accuracy and superiority of the SAA-SSA-BPNN situation assessment model, the mean square error (MSE), mean absolute error (MAE) and mean absolute percent error

Table 3. Comparison of assessment levels

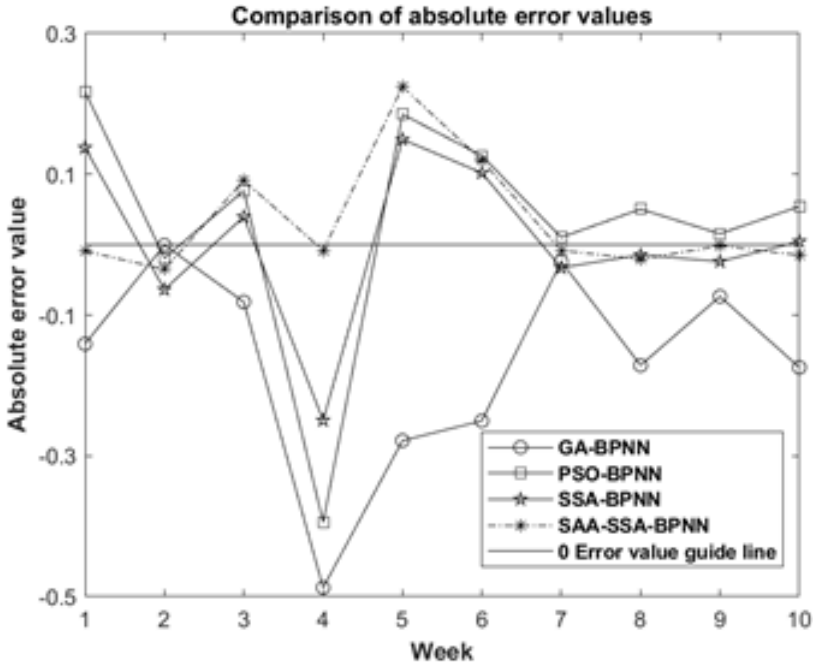
Week	1	2	3	4	5	6	7	8	9	10
CNCERT	M	G	M	M	M	M	G	G	G	G
GA-BPNN	G	G	M	E	G	G	G	E	G	E
PSO-BPNN	P	G	M	E	P	P	G	G	G	G
SSA-BPNN	P	G	M	G	P	P	G	G	G	G
SAA-SSA-BPNN	M	G	M	M	P	P	G	G	G	G

(E represents excellent; G represents good; M represents medium; P represents poor.)

Table 4. Comparison of absolute error values

GA-BPNN Absolute Error Value	PSO-BPNN Absolute Error Value	SSA-BPNN Absolute Error Value	SAA-SSA-BPNN Absolute Error Value
-0.1410	0.2173	0.1377	-0.0092
-0.0007	-0.0191	-0.0637	-0.0349
-0.0813	0.0767	0.0397	0.0912
-0.4872	-0.3943	-0.2495	-0.0083
-0.2783	0.1853	0.1495	0.2242
-0.2503	0.1263	0.1020	0.1197
-0.0240	0.0104	-0.0327	-0.0082
-0.1714	0.0505	-0.0145	-0.0206
-0.0736	0.0155	-0.0240	-0.0017
-0.1743	0.0543	0.0045	-0.0148

Figure 6. Comparison of absolute error values



(MAPE) are used to measure the errors of the four models' assessment values from CNCERT. The three formulas are as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y-\hat{y}\right)^2 \tag{22}$$

$$MAE = \frac{1}{n}\left(\sum_{i=1}^{n}\left|y-\hat{y}\right|\right) \tag{23}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y-\hat{y}}{y}\right| \tag{24}$$

Within the above formulas, y indicates the value from CNCERT, $\hat{y}$ is the situation assessment value, and n represents the number of test data.

Table 5 shows the MSE, MAE and MAPE between the network security situation assessment values obtained by the GA-BPNN model, PSO-BPNN model, SSA-BPNN model and SAA-SSA-BPNN model and the values from CNCERT. As clearly shown in Table 5, the error between the situation assessment values obtained by the SAA-SSA-BPNN model and the values from CNCERT are obviously smaller than those of the other three assessment models, which also shows that the SAA-SSA-BPNN situation assessment model has higher assessment accuracy.

Table 5. Accuracy comparison

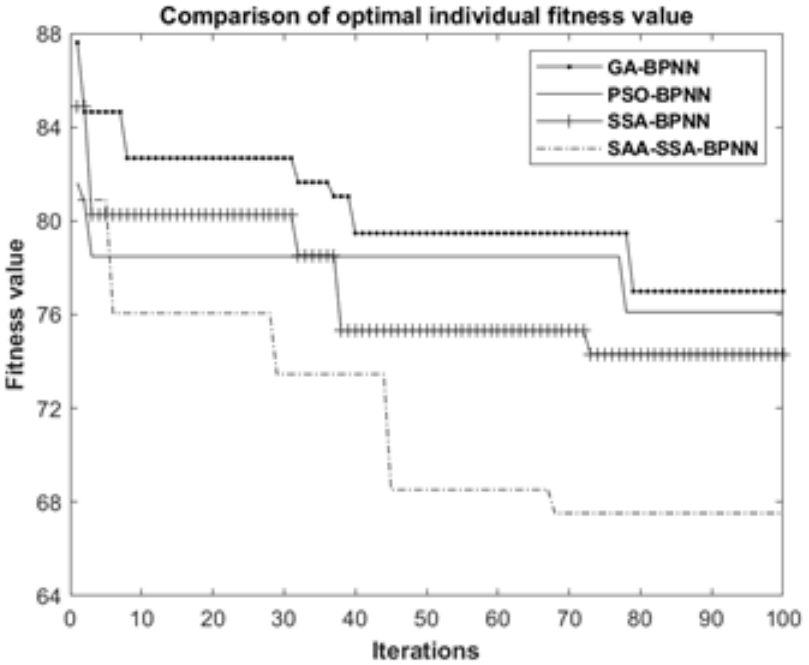
Assessment Index	GA-BPNN	PSO-BPNN	SSA-BPNN	SAA-SSA-BPNN
MSE	0.0470	0.0265	0.0121	0.0075
MAE	0.1682	0.1150	0.0818	0.0533
MAPE	0.3956	0.2499	0.1821	0.1173

Convergence Analysis

The experiment takes the sum of the absolute values of the assessment errors of the training data as the individual fitness value. Changes in fitness values demonstrate the convergence situation of the assessment model. The lower the fitness value, the better the individual. We compare the convergence of the GA-BPNN assessment model, PSO-BPNN assessment model, SSA-BPNN assessment model and SAA-SSA-BPNN assessment model, which is illustrated in Figure 7.

In Figure 7, the fitness value of the GA-BPNN assessment model is relatively high at the beginning. Although it jumps out of the local optimum at the 7th, 31st, 36th and 39th iterations, it traps a long-term local optimum at the 40th iteration and converges to 76.9932 at the 79th iteration. The PSO-BPNN assessment model sinks into the longest local optimum at the 3rd iteration and does not jump out of the local optimum by the 78th iteration; it eventually converges to the minimum at 76.1043. The SSA-BPNN assessment model also reaches a local optimum at the 3rd and 38th iterations, but it is in the local optimum for a shorter time than GA-BPNN and PSO-BPNN, and it converges to 74.3174 at the 73rd iteration. Compared with the GA-BPNN assessment model and PSO-BPNN assessment model, the SSA-BPNN assessment model has a faster convergence speed. Among the four models, the fitness value of the SAA-SSA-BPNN model initially shows the minimum. In the iterative process, SAA-SSA-BPNN jumps out of the local extremum several times, and at the 68th iteration,

Figure 7. A comparison of the changes in optimal individual fitness values



it tends to be stable and converges to the minimum of 67.5243. It has the fastest convergence speed and the minimum fitness value when a fitness value curve becomes steady, which cannot easily sink into a local optimum as the fitness value becomes steady. Therefore, the convergence effect of the SAA-SSA-BPNN situation assessment model is superior to that of the three other assessment models.

As shown in Figure 7, both the GA and PSO algorithms are trapped in a long-term local optimum. BPNN optimized by SSA improves the convergence speed. However, the SSA-BPNN model also falls into a local optimum, although the time to fall into a local optimum is shorter than that of GA-BPNN and PSO-BPNN. SSA is optimized by SAA to solve the problem that SSA easily sinks into a local optimum, and SSA-BPNN optimized by SAA has the smallest convergence value and the fastest convergence speed among the four algorithms. Obviously, the SAA-SSA-BPNN algorithm shows significant advantages in convergence.

Time Complexity Analysis

Time complexity is used to qualitatively describe the running time of the algorithm. The time complexity of the BPNN is influenced by the maximum number of iterations $iter_{max}$, the sample size n and the spatial dimension d, and its time complexity is $O\left(iter_{max} \times n \times d^2\right)$, which is approximately $O\left(d^2\right)$ when the spatial dimension d is high. After using the SSA algorithm to improve the BPNN, the complexity increases by $O\left(iter_{max} \times n \times d\right)$. Thus, the time complexity of the SSA-BPNN algorithm is $O\left(iter_{max} \times n \times d^2 + iter_{max} n \times d\right)$. When the spatial dimension d is high, it is approximate to $O\left(d^2\right)$, which is similar to the time complexity of the BPNN algorithm. For fear of the SSA algorithm sinking into a local optimum, the SAA algorithm is introduced to optimize it. However, the optimization process is completed in the iteration cycle of SSA without increasing the number of cycles, so it does not increase the computational load, and the time complexity is still $O\left(d^2\right)$. Therefore, the two improvements of the BPNN algorithm basically do not increase its time complexity.

CONCLUSION

To improve the BPNN, this paper introduces the SSA optimized by SAA and adopts the improved BPNN in the field of assessment of the network security situation. Then, the network security situation assessment model based on SAA-SSA-BPNN is presented, which solves the problems that the SSA easily falls into a local optimum. The optimal weights and thresholds of the BPNN are difficult to determine, and the convergence speed of the BPNN is slow to significantly enhance the accuracy and convergence speed of assessment. Future research will compare our model with other intelligent assessment models to find a situation assessment model with higher accuracy and assessment efficiency.

ACKNOWLEDGMENT

The authors like to thank all the reviewers for their valuable comments. This research was supported by the Henan Province Key Scientific Research Project for Higher Education Institutions [No. 21B520021] and the Henan Province Natural Science Foundation Project [No. 202300410508].

REFERENCES

Bass, T. (1999). Multisensor data fusion for next generation distributed intrusion detection system. In *Proceedings of 1999 IRIS National Symposium on Senor and Data Fusion* (pp. 1-6). The Johns Hopkins University.

Chang, L. W., Tian, X. X., Zhang, Y. Q., Qian, Y. H., & Hu, Z. G. (2021). Network security situation assessment system based on multisource heterogeneous data fusion. *Journal of Intelligent Systems*, *16*(01), 28–47.

Duan, X. J. (2019). Design of network security risk assessment system based on BP neural network optimized by chaotic particle swarm optimization. *Science Technology and Engineering, 19*(16), 251-255.

Feng, X. W., Wang, D. X., Ma, G. Q., & Li, J. (2011). Research on key technology of situation assessment in network situation awareness. *Computer Engineering and Applications, 47*(19), 88-92.

Gao, L., Luo, Z., Wang, W., Yang, X. D., Sun, Q., & Wang, F. (2021, April). *An improved BP neural network security situation assessment algorithm based on genetic algorithm*. Academic Press.

Kotenko, I., & Novikova, E. (2014). Visualization of security metrics for cyber situation awareness, In *2014 9th International Conference on Availability, Reliability and Security (ICARS), 2014 International Conference on* (pp. 506-513). IEEE. doi:10.1109/ARES.2014.75

MATLAB Chinese Forum. (2010). *Analysis of 30 cases of MATLAB neural network*. Beijing University of Aeronautics and Astronautics Press.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast com puting machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092. doi:10.1063/1.1699114

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(3), 533–536. doi:10.1038/323533a0

Smith, S. E. (2012). *Tightening the net: Examining and demon strating commonly available network security tools*. Faculty of the Department of Computing and Mathematical Sciences Texas A&M University.

Tao, X., Kong, K., Zhao, F., Cheng, S., & Wang, S. (2020). An efficient method for network security situation assessment. *International Journal of Distributed Sensor Networks*, *16*(11). Advance online publication. doi:10.1177/1550147720971517

Wang, H. B., Zhao, D. M., & Li, X. X. (2020). Research on network security situation assessment and forecasting technology. *Journal of Web Engineering*, *19*, 7–8.

Wang, J., Zhang, F. L., Fu, C., & Chen, L. S. (2007). Research on index system in network situation awareness. *Computer Application, 8*, 1907-1909+1912.

Wang, L. Y., Xu, R. R., Jin, X. B., & Ding, R. (2021, September). Comparison of nonlinear inversion comprehensive assessment among simulated annealing algorithm, genetic algorithm and neural network algorithm. *Journal of Wuhan University (Information Science Edition)*, 1-16. .10.13203/j.whugis20200217

Xue, J. K. (2020). *Research and application of a new swarm intelligence optimization technology*. Donghua University.

Yegneswaran, V., Barford, P., & Paxson, V. (2005). Using honeynets for intemet situational awareness. In *Maryland: Proceedings of the 4th Workshop on Hot Topics in Networks (WHTN), 2005 International Conference on* (pp. 762). IEEE.

Zhang, B., Ye, J. W., & Hu, Y. C. (2004). Application of simulated annealing algorithm in path optimization. *China Journal of Highway and Transport*, *01*, 83–85. doi:10.19721/j.cnki.1001-7372.2004.01.020

Zhang, R., Pan, Z. H., & Yin, Y. F. (2021). Research on assessment algorithm for network security situation based on SSA-BP neural network, In *2021 7th International Symposium on System and Software Reliability (ISSSR), 2021 International Conference on* (pp. 140-145). IEEE. doi:10.1109/ISSSR53171.2021.00024