Application of Improved CSA Algorithm-Based Fuzzy Logic in Computer Network Control Systems

Jianxi Yu
School of Engineering & Economics, Henan Institute of Economics and Trade, Zhengzhou, 450046, Henan, China

***Abstract*—In the past few years, with the high-speed popularization of computers and the widespread use of smart phones and mobile devices, the Internet has gradually become an indispensable part of people's daily lives. The Internet is constantly driving the process of digital society and providing people with more convenient and innovative applications. However, the internet industry also faces challenges such as runtime ambiguity, instability, large data volume, and difficulties in network situational awareness. In response to the above issues, this study combines the standard cuckoo algorithm with a fuzzy neural network to design a computer network situational awareness system. It uses principal component analysis to deduct the dimensions of the original data and then adds Gaussian noise to introduce appropriate randomness. The test proved that the improved model had a significant optimization effect on real network data, with an improvement of about 81.2% compared to the standard cuckoo algorithm. In the 220th iteration of the test set, the Loss function value was 0, which could accurately predict the network situation, with an accuracy rate of 98%. The designed system identification has higher recognition accuracy and less time consumption and has certain application potential in computer networks.**

Keywords—CSA; computer network; fuzzy logic; principal component analysis method; network operation; situation awareness

I. INTRODUCTION

Since the beginning of the 21st century, information technology has rapidly advanced, and the Internet has played an increasingly important role in social production and life. The average monthly traffic volume of the population has been increasing year by year [1]. The application of this technology is becoming increasingly diversified, giving rise to new industries such as 5G technology, the Internet of Things, edge computing, cloud computing, and data center optimization. This is constantly changing people's way of life and work, but it also brings new challenges and opportunities. It is widely acknowledged that the network runtime state exhibits several characteristics that present significant challenges to the development of the internet industry. These include ambiguity, instability, a considerable volume of data, and a lack of situational awareness within the network. In response to the above issues, experts and scholars in the field of the Internet have applied the cuckoo algorithm to computer network control systems. This algorithm is a heuristic optimization algorithm inspired by the reproductive behavior of cuckoo birds [2]. The algorithm realizes the optimization of the computer network control system by initializing the cuckoo group, generating new solutions, evaluating and selecting, updating and iterating, judging the convergence conditions, and analyzing the results [3]. However, the related research precision is not high, the generalization ability is low, the training time is long, and the rate of convergence is slow. In this study, the Principal Component Analysis (PCA) is first used to reduce the dimensions of the huge and changeable network data, and then Gaussian noise is introduced to lift the rate of convergence of the algorithm. Based on the standard Cuckoo Search Algorithm (CSA), a computer Network Situational Awareness Model (NSAM) is designed by integrating a Fuzzy Neural Network (FNN). The paper mainly consists of five sections. Section II summarizes the research status of scholars in the industry on the difficulties of Internet situational awareness. Section III establishes a computer NSAM that integrates CSA and fuzzy logic. Section IV conducts comparative experiments and efficiency verification on the optimization effect of the model. Section V is a summary of the research and an explanation of the direction for improvement.

II. RELATED WORKS

As a result of the growing use of mobile internet in a range of sectors, predictive models with enhanced network situational awareness capabilities are increasingly attracting interest from businesses and researchers. Liu C et al. introduced cloud control middleware to manage service requests to meet constraints, aiming at the problem that traditional cloud computing mode makes it difficult to provide real-time computing resources. They developed a conceptual computing framework built on cloud and mist combination, which has better performance in energy consumption and response time [4]. Abed Algoni B H et al. used a special type of opposition-based learning ECS model to address the problem of CSA being prone to suboptimal situations. The experiment showed that ECS exhibited better performance than all tested variants [5]. Cheng P et al. proposed Particle Swarm Optimization (PSO) - CSA to predict local comfort and global comfort by artificially solving the problems of motion state and being unable to be directly used for model analysis. This model had a high prediction accuracy [6]. Eltamally A M et al. proposed PSO and CSA to capture global peaks in the P curve of Photovoltaic (PV) arrays, which have advantages in optimizing control parameters [7]. Fan J et al. designed a chaotic CSA image segmentation model to address the issue of difficulty in improving accuracy in noisy images. This model improved accuracy and reduced uncertainty [8]. Li J et al. designed a balanced learning differential CS extension algorithm to solve the problem of CSA easily falling into local optima, which lifted the algorithm's global search capacity and accuracy [9].

CSA has unique advantages in group optimization problems and provides a certain reference for Internet network situational awareness. Chen S Y et al. designed a variable order fuzzy fractional proportional integral differential control system to address the issue of the inability of integral differential (PID) controllers to achieve high-precision control. This system could achieve better control response and anti-interference characteristics than Integer Order (IO) controllers [10]. Muhammad K et al. designed a television camera monitoring system based on fuzzy logic to continuously monitor the phenomenon of a large amount of data generated by television cameras every day. This model could handle data uncertainty in the real-world domain [11]. To solve the problem of autonomous decision-making of mobile robots to overcome obstacles, Ben Jabeur C et al. established a decision model based on an intelligent PID optimization neural network and fuzzy logic controller. The mobile robot applying this model could quickly execute tasks and adapt to constantly changing environmental conditions [12]. Costa R et al. designed a mountain flood prediction model based on classification and regression trees, deep learning neural networks, and fuzzy logic to identify slopes with a high probability of mountain flood outbreaks. The prediction accuracy exceeded 84% [13]. In response to the issue of insufficient drone controllers to cope with weather disturbances, Ulus Ş designed a drone control model by integrating classic PID and fuzzy logic controllers. This controller had better performance than other controllers [14]. Katsikis V N et al. designed a multi-objective evolutionary network framework to address the lack of flexibility in fuzzy logic neural networks, which has advantages in effectiveness and interpretability [15].

In summary, the application of fuzzy logic and CSA in computer network control systems has sufficient theoretical and practical foundations, but relevant research rarely combines the two to solve the problem of network situational awareness difficulties. Therefore, this study improves CSA and combines fuzzy logic to design NSAM to promote further development of the internet industry.

III. Overall Plan Design for Network Operational Situation Awareness

This chapter is mainly divided into five sections. The first section is divided into establishing an indicator model and NSAM, while the second section improves the CSA and integrates fuzzy logic. In the third section, the computer network control system based on fuzzy logic is established, and PCA is taken to decrease the dimension of the original data.

A. *Establishment of NOSA Model*

Network Operational Situation Awareness (NOSA) refers to the real-time monitoring, analysis, and identification of various activities, events, and resources in the network to obtain a comprehensive understanding of the network's operational status [16]. NOSA can help organizations or network administrators detect abnormal activities, attack attempts, or system failures promptly, and take corresponding measures to protect the security and stability of the network. Hence, this system needs to include the whole links from data to situation analysis to users. By the above requirements, this manuscript proposes the NOSA system, as exhibited in Fig. 1.

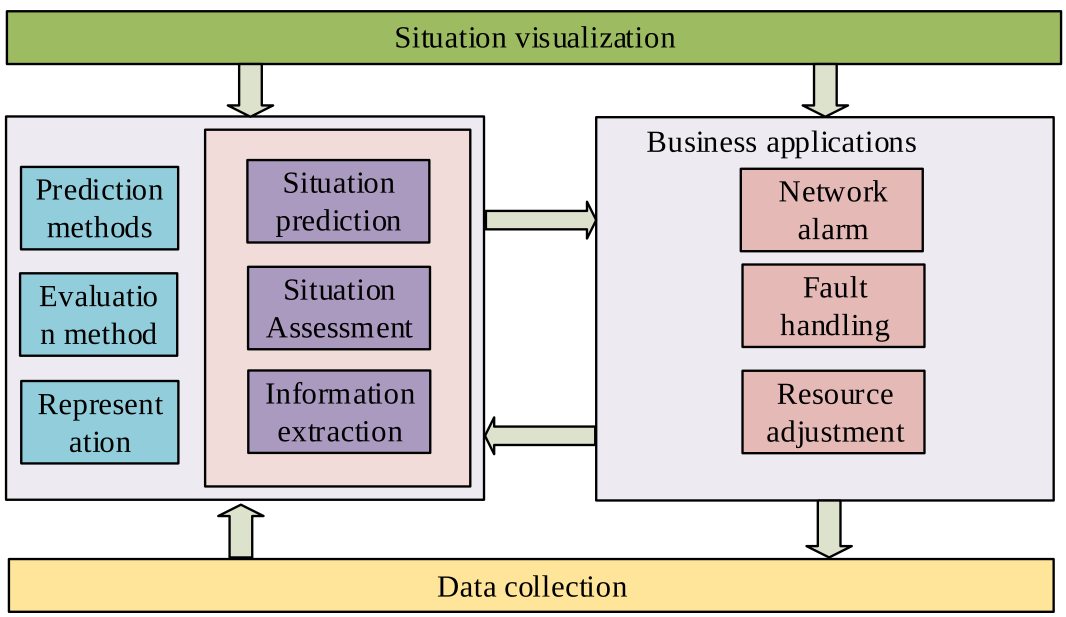

Fig. 1. Schematic diagram of network operation situational awareness system.

Fig. 1 shows a system that includes four modules: data, data fusion, situation visualization, and business application. Specifically, the results of the data fusion module can direct the management operations of the business application module. After the business application module manages the network, it will transmit new analysis results to the data fusion module through it. This cyclic process enables the system to continuously optimize and improve to better meet user needs. With the guidance of the above NOSA system model, a network operation situation indicator system can be constructed. This research presents a network operation situation indicator system, which is constructed from the perspectives of network performance and network traffic. The system is designed to integrate the TCP/IP five-layer model, as illustrated in Fig. 2.

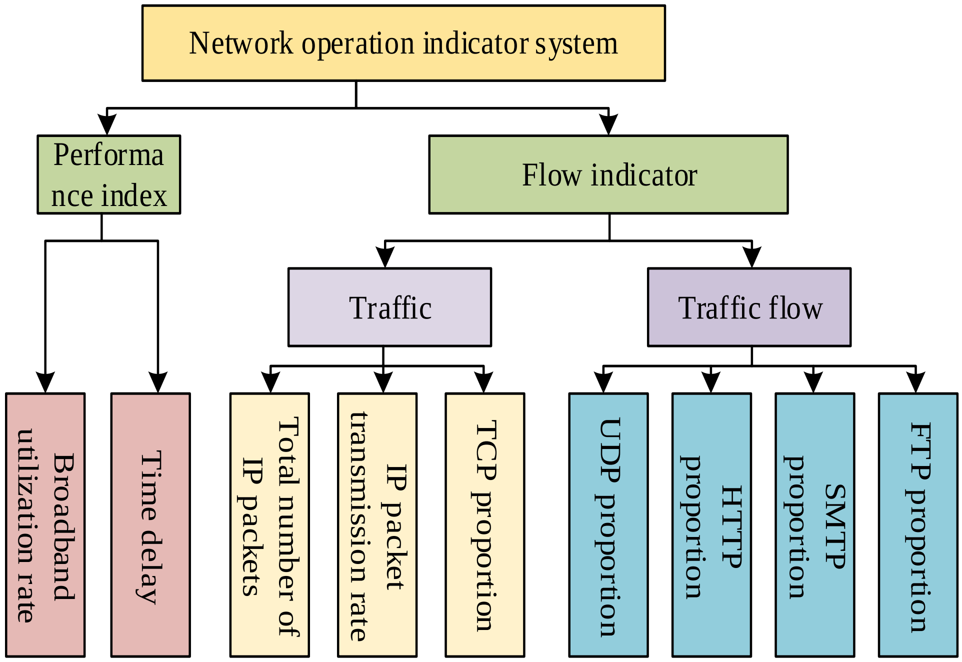

Fig. 2. Network operation situation indicator system.

Fig. 2 shows the network traffic indicators, including IP throughput, link utilization, the size and proportion of traffic based on network protocols, and the size and proportion of traffic. The network protocol traffic mainly covers protocols such as TCP, UDP, ICMP, etc; Business traffic mainly includes protocols i.e. HTTP, FTP, and SMTP. The goal of this study is to reflect the operational situation of the network from a business perspective. Therefore, it is necessary to integrate multiple protocol layer indicators [17]. The transport layer protocol traffic data, namely TCP, UDP and ICMP, are added to the network traffic indicator. After establishing the network operation situation indicators, it is also essential to sort out the original data according to the relevant calculation formulas. Broadband utilization is an important performance indicator in a network, which represents the current load level and Resource Utilization Efficiency (RUE) of the network. This characteristic indicator is calculated by Eq. (1).

$$L = \frac{T}{B} \tag{1}$$

In Eq. (1), T is the average transmission rate of the network, while B represents the maximum transmission rate of data packets in the network. In addition, this indicator system is one-way delay. Delay is used to represent the network Transmission delay. The calculation formula is Eq. (2).

$$Delay = \frac{\sum d}{sum} \tag{2}$$

In Eq. (2), d means the delay of all data packets transmitted by the network, and sum is the sum of the number of transmitted data packets. The quantity of data packets transmitted by the network per unit time is called the IP packet transmission rate, and its calculation formula is Eq. (3).

$$V_{IP} = \frac{sum}{t} \tag{3}$$

In Eq. (3), sum represents the total number of IP packets transmitted by the network, while t represents the total transmission time. In this indicator system, the proportion of protocol traffic between the application layer and the network layer is introduced. This type of indicator is represented by acc, and its calculation formula is Eq. (4).

$$acc = \frac{sum\_protocol}{sum} \tag{4}$$

In Eq. (4), $sum\_protocol$ represents the gross of protocol packets per unit time, and sum represents the total number of IP packets during that period.

B. Fuzzy Logic Model based on CSA Algorithm

CSA heuristic swarm intelligence optimization algorithm simulates the process of cuckoo's foraging and nest protection to achieve global optimization. The basic idea of the algorithm is to represent the candidate solutions of the problem as nests of cuckoo birds, with each nest corresponding to a solution vector. Then, the quality of each nest is evaluated based on the fitness function of the problem. CSA has good global search ability and high parallelism. It is suitable for various optimization problems, especially continuous optimization problems. The performance of the algorithm is still affected by factors such as parameter settings, nest protection strategy, and nest elimination strategy, and needs to be adjusted and optimized according to specific problems. Fig. 3 shows the algorithm flowchart.

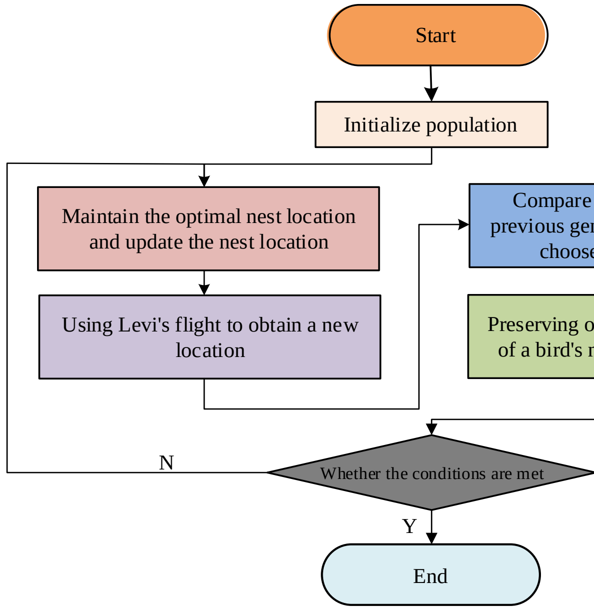

Fig. 3. Flowchart of cuckoo algorithm.

Fig. 3 is the CSA flowchart. Firstly, a set of feasible solutions is randomly produced as the initial population, and the optimal nest position is retained by adjusting the positions of the parents and parasitic birds. This step can use some heuristic methods, such as random walk or local search algorithms. Then, the position is updated by adjusting the position of the parent and parasitic birds. The next step is to eliminate solutions with lower fitness with a certain probability. Finally, to determine whether the termination condition is met, i.e. reaching the max-iterations or finding a solution that meets the requirements. Step 2 is repeated to 5 until the termination conditions are met. Although CSA has certain advantages and application value, there are also some shortcomings. For example, the rate of convergence is slow, the parameter selection is difficult, and the dependence on problem characteristics is strong. To solve the above problems, the idea of fuzzy logic is used to improve traditional CSA. The schematic diagram is shown in Fig. 4.

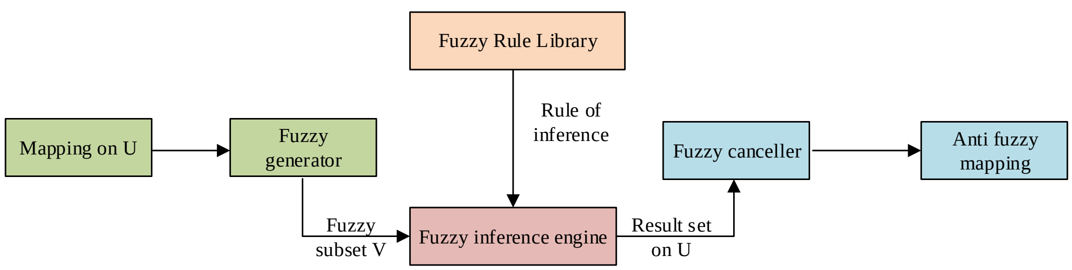

Fig. 4. Fuzzy mapping diagram.

Fig. 4 is a fuzzy inference system, which mainly consists of four parts. The first part is a fuzzy generator, mainly responsible for converting the precise input values into fuzzy membership values, that is, mapping the input to the corresponding membership functions. The second part is the fuzzy rule library, which defines a set of fuzzy rules, each containing a series of prerequisites and a conclusion. The preconditions and conclusions are described by fuzzy sets. The third part is the inference engine, which uses the inference mechanism to calculate the fuzzy output according to the given rules and the fuzzy input. The final part is to convert the fuzzy output back to precise values through anti-fuzzification. Unlike traditional binary logic, which only has true and false values, fuzzy logic allows variables to have a continuous range of values, with fuzziness between 0 and 1. Assuming the input variable is $x=[x_1,x_2,...,x_n]^T$, each component is a fuzzy variable. Each fuzzy variable is segmented into n fuzzy sets, and the fuzzy set of each component in the input variable is Eq. (5).

$$T(x_i)=\{A_i^1,A_i^2,...,A_i^k\},k=1,2,3,...,n \tag{5}$$

In Eq. (5), A_i^k is the k-th variable value of the i-th input component. The fuzzy outputs vector $y=[y_1,y_2,...,y_n]^T$ in this fuzzy model, if x_1 is A_i^k, the output fuzzy output vector is equation (6).

$$y_{it}=p_{0r}^i+p_{1r}^i x_1+\cdots+p_{nr}^i x_n \qquad 0<r<k^n \tag{6}$$

In Eq. (6), p_{0r}^i represents the output of the i-th output vector under rule r. This fuzzy rule can be represented as an IF THEN statement, as shown in Eq. (7).

IF x_1 A_i^k, THNE

$$\begin{cases} y_{1r} = p_{0r}^1 + p_{1r}^1 x_1 + \cdots + p_{nr}^1 x_n \\ \cdots \\ y_{ir} = p_{0r}^i + p_{1r}^i x_1 + \cdots + p_{nr}^i x_n \end{cases} \quad (7)$$

In Eq. (7), x_1 represents the k-th variable value of the i-th input component. By using single point fuzzification to represent input variables, the applicability of each rule can be calculated, as shown in Eq. (8).

$$T_r = \mu_{1k} \wedge \mu_{2k} \cdots \wedge \mu_{nk} = \mu_{1k} \cdot \mu_{2k} \cdots \mu_{nk} \quad (8)$$

In Eq. (8), T_r represents the applicability of rule r. The output value of the fuzzy system is the weighted average of each rule, and its formula is Eq. (9).

$$y_i = \sum_{q=1}^{q=r} T_q y_{iq} / \sum_{q=1}^{q=r} T_q \quad (9)$$

In Eq. (9), y_i is the i-th component of the output vector. Fuzzy logic is widely used in control systems, artificial intelligence, decision support systems, and other fields, especially suitable for problems with fuzziness and uncertainty. Through the reasoning and processing of fuzzy logic, incomplete information and fuzzy concepts in the real world can be better handled, improving the effectiveness of decision-making and control.

C. Computer Network Control System Based on Fuzzy Logic

The large amount of data in computer network control systems has uncertainty and fuzziness, so it is necessary to apply fuzzy theory to solve these problems. In addition, network data also have the characteristics of being massive and multidimensional. Neural network is an effective method to deal with big data. When it is combined with fuzzy theory, it can solve NOSA problem in complex data environment. Therefore, an improved CAS algorithm, CSA-FNN, is designed by combining CAS and FNN, and its structure is Fig. 5.

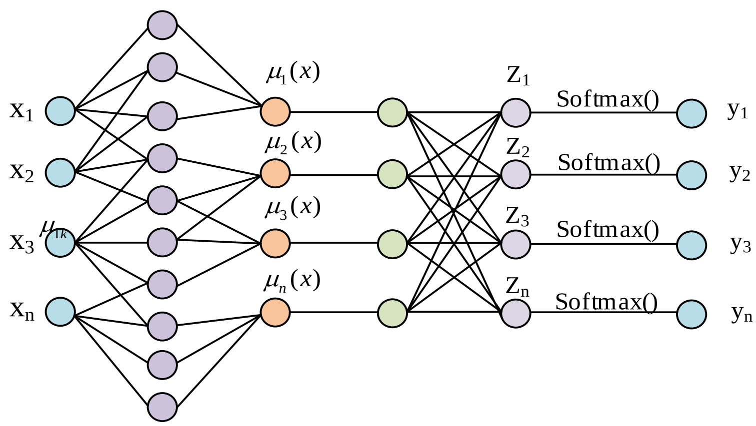

Fig. 5. CSA fuzzy neural network.

In Fig. 5, the first is the input layer, which completes the input of training data. The second is the rule mapping layer, which fuzzily divides each input component. The third is the rule fitness layer, where each node represents a fuzzy rule. The fourth is the normalization layer, which normalizes the fitness of each rule. The fifth is the anti-fuzzy layer, which completes the mapping from the fuzzy rule space to the output space through the Activation function. The last layer is the output layer, which outputs the network operational situation level. In the above CSA-FNN model, the network operational situation features include 10 indicators, that is, the input data is 10 dimensions. The number of nodes in each layer increases exponential type with the number of nodes in the input layer, resulting in too many nodes. Therefore, it is necessary to reduce the dimensions of the original data. This study uses PCA to reduce and reconstruct the dimensions of the original data. Fig. 6 is the process steps.

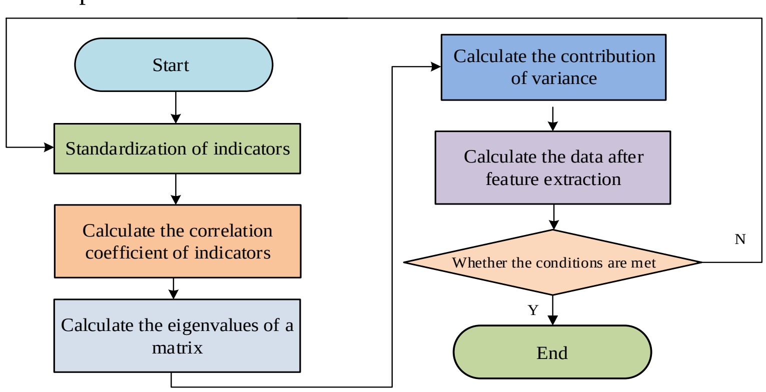

Fig. 6. CSA-FNN.

The first is to standardize the indicators. The dimensions of the original indicator data in computer networks are different, so it is necessary to normalize the indicator data and convert them into data under the same dimension. This article adopts the maximum-minimum normalization method, and its expression is shown in Eq. (10).

$$I=\begin{cases} I_{11} & I_{12} & \cdots & I_{1m} \\ I_{21} & I_{11} & \cdots & I_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ I_{n1} & I_{n2} & \cdots & I_{nm} \end{cases}=(I_1,I_2,...,I_m) \tag{10}$$

In Eq. (10), I represents the normalized network indicator data, and m represents that each data has m indicators. $\mathrm{I_m}$ represents the m-th dimensional vector of the data, normalized as Eq. (11).

$$i_{jp}'=\frac{i_{jp}-\overline{I_p}}{\sqrt{\sigma_p}} \tag{11}$$

In Eq. (11), $\overline{I_p}$ represents the average value of the p-th indicator. σ_p represents the mean square deviation of the p-th indicator. The correlation coefficient of each indicator in I is calculated to obtain the correlation coefficient matrix, as shown in Eq. (12).

$$A=\frac{1}{m}I^T I\begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1m} \\ R_{21} & R_{11} & \cdots & R_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ R_{m1} & R_{m2} & \cdots & R_{mm} \end{pmatrix} \tag{12}$$

In Eq. (12), $R_{ij}(i,j=1,2,...,m)$ represents the correlation coefficient between the i-th and j-th indicators. The eigenvalue of matrix A and Orthogonalization is calculated, as shown in Eq. (13).

$$a=(a_1,a_2,...,a_m) \tag{13}$$

In Eq. (13), a is the matrix obtained by Orthogonalization the characteristic matrix of matrix A. Then the contribution of each feature vector is calculated to the square difference, and the contribution calculation formula is Eq. (14).

$$a_i=\frac{\lambda_i}{\sum_{k=1}^{m}\lambda_k} \tag{14}$$

In Eq. (14), $a_i(i=1,2,...,m)$ is the contribution rate of the mean square deviation of the i-th eigenvector. Finally, the data after feature extraction based on the formula are calculated. After the above 5 steps of processing, the dimension of the data is reduced to 4 dimensions, and the number of nodes is significantly reduced, solving the problem of massive and multidimensional network data. Therefore, the improved CSA algorithm based on fuzzy logic has been successfully applied to computer network control systems.

IV. CSA-FNN Model Performance Testing

This chapter mainly verifies the optimization effect of the model. The first section mainly compares and analyzes CSA-FNN with other algorithms to verify the comparative advantages of the algorithm. Then, distinctive datasets are used to verify the generalization ability of the model. The second section mainly conducts simulation experiments to test the efficiency in practical environments.

A. *Comparative Analysis of Algorithms and Validation of Generalization Ability*

This study uses CSA combined with FNN to establish a computer network control system model, solving the problems caused by the uncertainty and fuzziness of computer network data. The original data dimension is reduced by PCA, which significantly reduces the number of nodes and complexity of the neural network [18]. To evaluate the optimization ability of fuzzy inference systems for computer network control systems, the experiment uses Python 3.8 on the Windows 10 platform and uses the Cooperative Association for Internet Data Analysis (CAIDA) dataset to perform 1000 iterations on the traditional CSA and CSA-FNN models, respectively. Fig. 7 shows the relationship between its training error and the iterations.

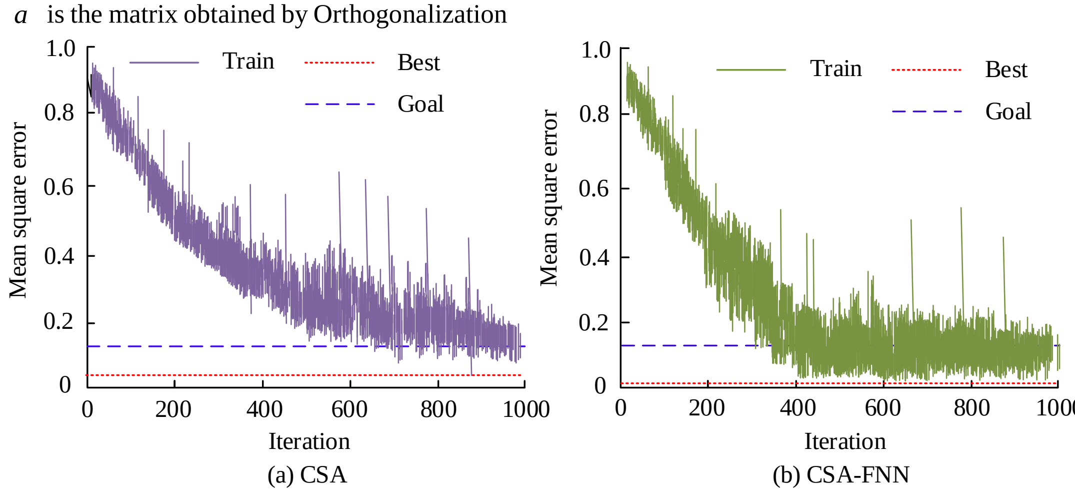

Fig. 7. Comparison of CSA and CSA-FNN algorithm errors.

Fig. 7 shows the comparison of the effects of CSA before and after improvement on the training dataset. Compared to the standard CSA model, the rate of error reduction in the first 200 iterations of CSA-FNN is not significantly different. However, when the number of iterations reaches the range of [200, 400], the error of the FNN structure rapidly decreases and converges by the 400th iteration. The error of traditional CSA tends to converge after 700 iterations. Therefore, the CSA-FNN model proposed in the experiment has a faster convergence rate and a lower error in the final convergence. To eliminate the impact of dataset selection on experimental results and verify the generalization ability of the model, it is necessary to apply the Measurement and Analysis on the WIDE Internet (MAWI) dataset to train the above algorithms. Table I is the MAWI dataset parameter table.

Table I shows that the MAWI dataset is a very large dataset, including a large amount of network traffic data. Therefore, appropriate preprocessing and sampling are required when using this dataset for analysis and research [19]. The MAWI dataset usually contains more complex network traffic patterns, while the CAIDA dataset focuses more on reflecting the traffic characteristics of the actual Internet. For these two datasets, researchers preprocess the data, including data cleaning and normalization, to reduce noise and eliminate dimensional differences between features. Afterwards, a comparison is made between CSA-FNN and Whale Optimization Algorithm (WOA), Ant Colony Optimization (ACO), PSO, and Artificial Fish Swarm Algorithm (AFSA) to observe their training performance on different datasets. The results are shown in Fig. 8.

TABLE I. BASIC PARAMETERS OF THE ACTION DATASET

Parameter type	Parameter scale
Traffic data	735000
Time stamp	6900min
Source IP Address and Destination IP Address	900
Protocol	4
Packet size	512bit
Packet Marking	5

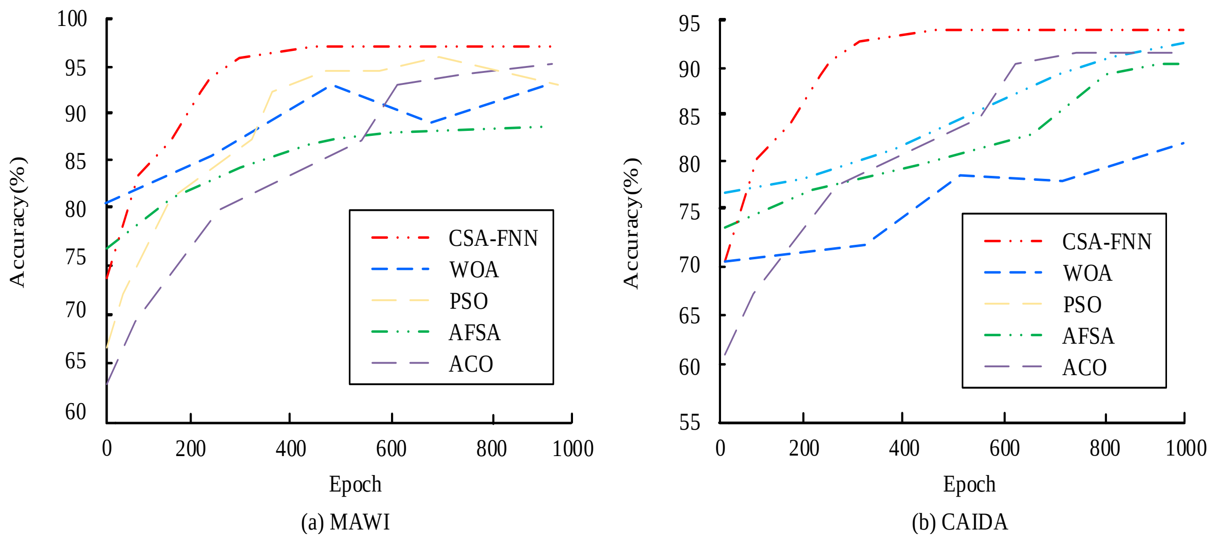

Fig. 8. Comparison of MAWI and CAIDA data set.

Fig. 8 shows the trend curve of the accuracy of various algorithms in two datasets as a function of the amount of training rounds. From Fig. 8 (a), the CSA-FNN model performs best, with accuracy tending to converge after 300 iterations, while the other four models only converge after training 600 times. In Fig. 8 (b), the accuracy of all algorithm models decreases when using the CAIDA dataset. However, the convergence of CSA-FNN accuracy does not change much, and it tends to converge after 400 iterations, ultimately converging to around 94%. The above results demonstrate that CSA-FNN has advantages over the other four algorithms, such as fast convergence, high accuracy, high stability, and strong generalization ability.

B. *NOSA Simulation Experiment*

The comparative analysis of algorithms has successfully verified the comparative advantages of CSA-FNN compared to other algorithm models, providing a solid theoretical foundation for its application in real computer network control systems. Although the superiority of the CSA has been verified, further simulation experiments are still necessary to evaluate the scalability, robustness, and stability of the algorithm. The experiment utilizes real traffic data from MAWILAB and obtains network traffic data through steps such as data cleaning and normalization [20]. Link utilization, IP packet rate, total number of IP packets, and TCP ratio are used as evaluation indicators. Link utilization refers to the degree to which network links are occupied by valid data. The IP packet rate and total number reflect the strength of network traffic. The TCP ratio represents the proportion of Transmission Control Protocol (TCP) packets to the total number of packets. Table II shows some data.

TABLE II. SOME NETWORK TRAFFIC DATA

Serial Number	Link Utilization	IP Packet Rate	Total Number of IP Packets	TCP Proportion
734	0.0113659	0.009562344	0.011231245	0.442123226
735	0.0132411	0.009878563	0.016324534	0.442661626
736	0.0114534	0.009758231	0.012313122	0.412336123
737	0.0164553	0.009431312	0.011229807	0.412286126
738	0.0174554	0.009341123	0.011123145	0.512326112
739	0.0142432	0.009234133	0.011212343	0.123146126
740	0.0111311	0.009124313	0.011224344	0.642666126
741	0.0141233	0.009413444	0.011224523	0.412312313
742	0.0143133	0.009512312	0.011253451	0.482626126
743	0.0143566	0.009413123	0.011231312	0.144567435
744	0.0163234	0.009434546	0.011212343	0.734341231
745	0.0178621	0.009223431	0.011213217	0.423123126
746	0.0115456	0.009254323	0.011221203	0.442231226
747	0.0112567	0.009256723	0.011231207	0.431234112

Normalizing the original data is helpful to eliminate the dimensional difference between features, improve the training effect and stability of the model, and improve the rate of convergence of the model. Then the normalized real network traffic data are used to verify the improved CSA-FNN structure. The simulation software used in the study is Pychar, and the simulation environment is Python 3.6. 1000 iterative training sessions are performed on CSA and CSA-FNN using the data shown in Table II. The link weights of the normalization layer and the anti-fuzzification layer are randomly generated by a Gaussian function to circumvent the issue of gradient disappearance or explosion due to an excessive data volume of the model. Furthermore, the introduction of randomness serves to enhance the stability of the model's learning and optimization process. The results are displayed in Fig. 9.

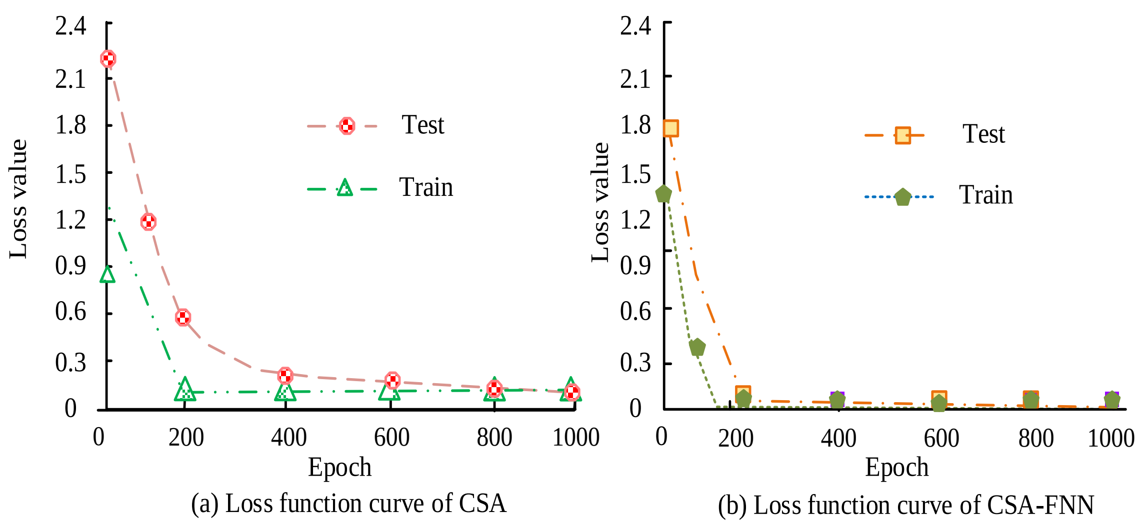

Fig. 9. Loss function curve of CSA and CSA-FNN.

In Fig. 9, the data gradient after the normalization operation is significantly reduced, and the stability of the model is significantly improved compared with the MAWI dataset, and the rate of convergence is accelerated. From Fig. 9(a), the CSA training set curve converges after about 200 iterations, while the test set converges after about 400 iterations. From Fig. 9(b) that during the first 120 iterations of the training set of the CSA-FNN model, the value of the Loss function rapidly decreases from 1.5 to 0.35, a decrease of about 76.7%. In the 220th iteration of the test set, the Loss function value is basically 0. Comparing the two figures, it can be found that CSA-FNN has increased by about 81.2% compared to CSA. Finally, Capsa software is used to capture real-time data traffic in the current local area network for analysis, intercepting network data traffic information for two consecutive hours. The experimental results are exhibited in Fig. 10.

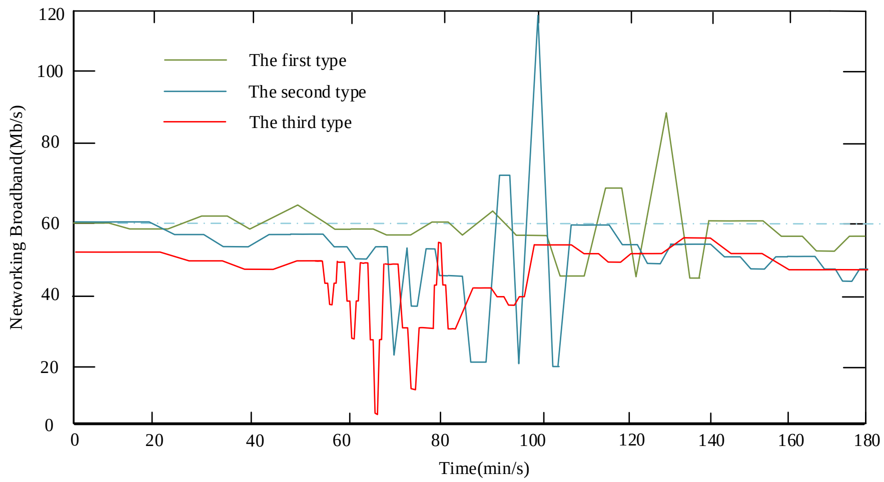

Fig. 10. Map of changes in network operational situational awareness.

Fig. 10 shows the changes in three types of NOSA. From the graph, the network bandwidth of the first type of network operation situation is mostly in a state of 40-60Mb/s, with occasional small fluctuations. This situation may be caused by accidental network changes. The second type of network operation situation is that the network is normal in the first 60 minutes, and severe fluctuations begin to occur in the first 60 minutes. The reason for this situation may be due to unstable factors in the local area network. The third type of network operation situation curve shows that the network is normal in the first 50 minutes, and then the network quality rapidly decreases. This situation may be due to the sudden addition of new tasks and drastic fluctuations in the network. Compared with the actual data, the network awareness model can accurately predict the network situation, with an accuracy rate of 98%. In addition, the study also records the changes in perception accuracy of the CSA-FNN model in different network environments, as shown in Fig. 11.

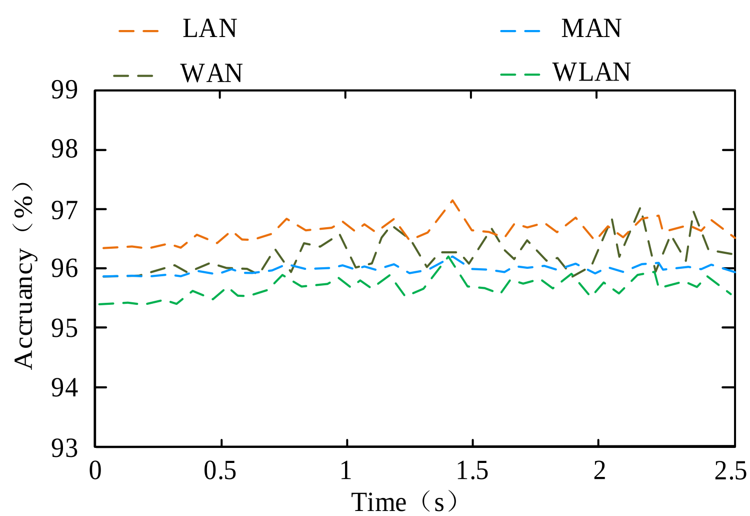

Fig. 11. Changes in perception accuracy under different network environments.

Fig. 11 shows the temporal variation of perception accuracy of the CSA-FNN model in different types of network environments (LAN, MAN, WAN, and WLAN). In Fig. 11, with the increase of time, the accuracy of various network environments shows an upward trend, among which LAN and WLAN have higher accuracy, maintaining above 98% and 99% respectively, while MAN and WAN have relatively lower accuracy, but also exceeding 97%. This phenomenon may be due to the relatively simple network environment of LAN and WLAN, which are easy to predict and classify, while MAN and WAN, due to their complex network structure, may have more uncertain factors leading to slightly lower accuracy. In addition, the fluctuation of each line type in the figure is relatively large, which may be caused by the dynamic changes in the network environment and the irregularity of the data flow. Overall, the CSA-FNN model can maintain high accuracy in various network environments, demonstrating its good adaptability and robustness. Finally, to verify the effectiveness of the algorithm in practical applications, the WSSD system proposed in study [21], the WEA-SA NSAM proposed in study [22], and the OWS-WOA model proposed in study [23] are introduced to compare with the CSA-FNN. The implementation phase comprises the deployment of infrastructure, the implementation of security controls, and the execution of security processes.

These include the installation of network and security hardware, the configuration of network devices and firewalls, the deployment of intrusion detection systems and SIEM systems, and the execution of key security processes such as security auditing, data backup and recovery, patch management, etc. The experimental results are shown in Table III.

TABLE III. COMPARISON BETWEEN CSA-FNN AND OTHER NETWORK SITUATIONAL AWARENESS SYSTEMS

Experimental Group	Detection Rate (%)	False Alarm Rate (%)	Response Time (ms)	Computing Resource Consumption (%)
CSA-FNN	94.2	2.5	35	12
WSSD	91.8	3.1	40	14
WEA-SA	93.0	2.8	38	13
OWS-WOA	92.5	3.0	42	15

According to the data in Table III, the CSA-FNN algorithm exhibits high performance in network situational awareness. Specifically, the detection rate of CSA-FNN reaches 94.2%, which is the highest among the four experimental groups, indicating that it can more accurately identify abnormal behaviors or threats in the network. Meanwhile, the false positive rate of CSA-FNN is 2.5%, which is also the lowest among the four experimental groups, indicating that it performs better in reducing unnecessary alarms and thus minimizing interference with normal network activity. In terms of response time, CSA-FNN is 35 milliseconds, faster than WSSD and WEA-SA, but slightly slower than OWS-WOA. However, a response time of 35 milliseconds is still very fast, which is already fast enough for real-time network situational awareness. In terms of computational resource consumption, CSA-FNN consumes 12% of resources, which is the lowest among all models, demonstrating its advantage in RUE. In contrast, OWS-WOA has the highest resource consumption, reaching 15%. Overall, CSA-FNN performs well in the three key indicators of detection rate, false positive rate, and computational resource consumption, especially in terms of detection rate and resource consumption, indicating that CSA-FNN is an efficient and accurate tool for network situational awareness.

V. CONCLUSION

To solve the problems of fuzziness, instability, large data volume, and difficulty in network situational awareness in network operation, this study designed a computer NSAM based on CSA and fuzzy logic fusion. The experiment used Python 3.8 on the Windows 10 platform and trained CSA and CSA-FNN using the CAIDA dataset. The results showed that the error of the proposed FNN structure rapidly decreased and converged at the 400th iteration. The error of the standard CSA only converged after 700 iterations, and the convergence rate increased by 75%. Then, the performance of CSA-FNN algorithm was evaluated through horizontal comparative experiments with algorithms such as WOA, ACO, PSO, AFSA, etc. The data showed that the accuracy of all algorithm models has decreased when using the CAIDA dataset. However, the accuracy convergence of the CSA-FNN model did not change much, and it tended to converge after 400 iterations, ultimately converging to around 94%. This proved that CSA-FNN had advantages over the other four algorithms such as fast convergence, high accuracy, high stability, and strong generalization ability. Finally, simulation experiments were conducted using MAWILAB: During the first 120 iterations of CSA-FNN, the testing accuracy improved with the improvement of training accuracy, and the speed was very significant. During this period, the value of Loss function decreased rapidly from 1.5 to 0.35, with a decrease of about 76.7%. Therefore, this model has good practical application capabilities. However, the model still requires a considerable amount of training and a longer training time, which is also an area that can be further improved in future research. The performance of the proposed model is an important consideration when the network size and complexity increase. The scalability of the model is key to ensuring its effective operation in larger or more complex network environments. As the scale of the network expands, models may need to handle larger amounts of data. Therefore, optimizing the data processing flow and algorithm efficiency is necessary to maintain or improve the response speed and accuracy of the model. In addition, the system for monitoring and measuring network status needs to effectively utilize computing resources. In the future, the concept of RUE can be studied and implemented to ensure that the system is efficient in resource utilization, especially in situations where multiple resources are limited.

REFERENCES

[1] Amin S N, Shivakumara P, Jun T X. An Augmented Reality-Based Approach for Designing Interactive Food Menu of Restaurant Using Android.Artificial Intelligence and Applications. 2023, 1(1): 26-34.

[2] Nsugbe E. Toward a Self-Supervised Architecture for Semen Quality Prediction Using Environmental and Lifestyle Factors.Artificial Intelligence and Applications. 2023, 1(1): 35-42.

[3] Oslund S, Washington C, So A, Multiview Robust Adversarial Stickers for Arbitrary Objects in the Physical World. Journal of Computational and Cognitive Engineering, 2022, 1(4): 152-158.

[4] Liu C, Wang J, Zhou L. Solving the multi-objective problem of IoT service placement in fog computing using cuckoo search algorithm. Neural Processing Letters, 2022, 54(3): 1823-1854.

[5] Abed-alguni B H, Alawad N A, Barhoush M, et al. Exploratory cuckoo search for solving single-objective optimization problems. Soft Computing, 2021, 25(15): 10167-10180.

[6] Cheng P, Wang J, Zeng X,Zeng, X. , Bruniaux, P. , & Tao, X..Motion comfort analysis of tight-fitting sportswear from multi-dimensions using intelligence systems.Textile Research Journal, 2022, 92(11-12):1843-1866.

[7] Eltamaly A M. Optimal control parameters for bat algorithm in maximum power point tracker of photovoltaic energy systems.International Transactions on Electrical Energy Systems, 2021, 31(4):1-22.

[8] Fan J, Xu W, Huang Y. Application of chaos cuckoo search algorithm in computer vision technology. Soft Computing, 2021, 25(18): 12373-12387.

[9] Li J, Yang Y H, Lei H, & Wang, G. G. Solving logistics distribution center location with improved cuckoo search algorithm. International Journal of Computational Intelligence Systems, 2021, 14(1): 676-692.

[10] Chen S Y, Yang M C. Nonlinear Contour Tracking of a Voice Coil Motors-Driven Dual-Axis Positioning Stage Using Fuzzy Fractional PID Control with Variable Orders.Mathematical Problems in Engineering, 2021, 2021(4):1-14.

[11] Muhammad K, Obaidat M S, Hussain T, Ser, J. D., & Doctor, F. Fuzzy Logic in Surveillance Big Video Data Analysis: Comprehensive Review, Challenges, and Research Directions.ACM Computing Surveys, 2021, 54(3):1-33.

[12] Ben Jabeur C, Seddik H. Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot. Asian Journal of Control, 2021, 23(1): 23-41.

[13] Costache R, Arabameri A, Moayedi H, Pham, Q. B., Santosh, M., Nguyen, H, Pham, B. T. Flash-flood potential index estimation using fuzzy logic combined with deep learning neural network, naïve Bayes, XGBoost and classification and regression tree. Geocarto International, 2022, 37(23): 6780-6807.

[14] Ulus Ş, Eski I. Neural network and fuzzy logic-based hybrid attitude controller designs of a fixed-wing UAV. Neural Computing and Applications, 2021, 33(14): 8821-8843.

[15] Katsikis V N, Stanimirović P S, Mourtas S D,Xiao, L., Karabašević, D., & Stanujkić, D. Zeroing neural network with fuzzy parameter for computing pseudoinverse of arbitrary matrix. IEEE Transactions on Fuzzy Systems, 2021, 30(9): 3426-3435.

[16] Dong S, Yu T, Farahmand H, et al. A hybrid deep learning model for predictive flood warning and situation awareness using channel network sensors data. Computer-Aided Civil and Infrastructure Engineering, 2021, 36(4): 402-420.

[17] Tan L, Yu K, Ming F, Cheng, X., & Srivastava. Secure and resilient artificial intelligence of things: a HoneyNet approach for threat detection and situational awarenes. IEEE Consumer Electronics Magazine, 2021, 11(3): 69-78.

[18] Huang D, Jiang F, Li K,Tong, G., & Zhou, G.. Scaled PCA: A new approach to dimension reduction. Management Science, 2022, 68(3): 1678-1695.

[19] Mawi H, Narine R, Schieda N. Adequacy of unenhanced MRI for surveillance of small (clinical T1a) solid renal masses. American Journal of Roentgenology, 2021, 216(4): 960-966.

[20] Aytekin A. Comparative analysis of the normalization techniques in the context of MCDM problems. Decision Making: Applications in Management and Engineering, 2021, 4(2): 1-25.

[21] Qidong Y, Jianbin G, Shengkui Z, et al. A dynamic Bayesian network-based reliability assessment method for short-term multi-round situation awareness considering round dependencies. Reliability engineering & system safety, 2024, 243(Mar.): 1109838.1-1109838.17.

[22] Zhitao C, Xiaodong Y, Yiyong Z. Research on hierarchical network security situation awareness data fusion method in big data environment. Journal of Cyber Security Technology, 2024, 8(1): 31-52.

[23] Guo, X, Jianing, Y, Zhanhui, G, et al. Research on Network Security Situation Awareness and Dynamic Game Based on Deep Q Learning Network. Journal of Internet Technology, 2023, 24(2): 549-563.