Fuzzy Neural Network Algorithm Application in User Behavior Portrait Construction

Peisen Song[1*], Bengcheng Yu[2], Chen Chen[3]
School of Economics and Management, China University of Mining and Technology, Xuzhou, 221000, China[1, 3]
School of Information Engineering, Xuzhou College of Industrial Technology, Xuzhou, 221000, China[1, 2]
School of Business Administration, Xuzhou College of Industrial Technology, Xuzhou, 221000, China[3]

***Abstract*—With the increasing number of online users, constructing user behavior profiles has received widespread attention from relevant scholars. In order to construct user behavior profiles more accurately, the research first designed an adaptive fuzzy neural network algorithm based on the momentum gradient descent method. It uses momentum gradient descent to optimize and learn the parameters adjusted by error backpropagation algorithm and least squares estimation method and optimizes the structure of the fuzzy neural network through subtraction clustering. Finally, the improved algorithm is applied to the construction of user behavior profiles. The results showed that in error analysis, the error range of the improved algorithm was within [-0.10, 0.10], and the accuracy was relatively high. In indicator calculation, the improved algorithm had a recall rate of 0.07 and 0.09 higher than the other two algorithms, an accuracy rate of 0.03 and 0.07 higher than the other two algorithms, and an F1 score of 0.07 and 0.08 higher than the other two algorithms, indicating good overall performance. In the ROC curve, the average detection rate of the designed user behavior profiling model was 0.065 and 0.155 higher than the other two models, respectively, with higher detection accuracy. These results demonstrated the effectiveness of improved algorithms and design models, providing certain reference value for the development of related fields.**

Keywords*—User behavior profiling; momentum gradient descent method; adaptive fuzzy neural network; error backpropagation algorithm; least squares estimation method; subtractive clustering

I. Introduction

In today's increasingly data-driven society, user behavior profiling is an important research object in data analysis [1]. User behavior profile construction can help enterprises gain a deeper understanding of users and develop more personalized and refined service and marketing strategies. In the process of constructing user behavior profiles, traditional user behavior analysis methods often struggle to handle such large-scale and complex data. Meanwhile, user behavior data often contains a large amount of uncertainty and ambiguity, making it a challenge to accurately analyze and predict user behavior. Fuzzy Neural Network (FNN) is an important modeling tool, which combines the adaptive learning ability of neural networks and the ability of fuzzy logic to handle uncertain and fuzzy information and can effectively learn and process nonlinear and complex data patterns [2]. Therefore, FNN is widely regarded by researchers as a powerful tool for dealing with various data problems in user behavior profiling, such as the nonlinearity and dynamic changes of user behavior, as well as the fuzzy relationship between user attributes and behaviors. However, although the theory and technology of FNN have developed to a considerable extent, its application in constructing user behavior profiles is still in the exploratory stage. On the one hand, due to the complexity of user behavior, applying FNN to practical user behavior profiling still faces many challenges; On the other hand, compared to other machine learning and data mining technologies, the superiority and adaptability of FNN in constructing user behavior profiles have not been fully demonstrated [3].

In this context, the study first utilizes the Back Propagation (BP) and Least Squares Estimation (LSE) to adjust the network parameters of FNN and then utilizes the Gradient Descent with Momentum (GDM) method to optimize and learn the adjusted parameters. The structure of FNN is optimized using the Subtractive Clustering Method (SCM) to shorten training time, and an MGD-ANFIS algorithm is designed for the model construction of user behavior profiling, in order to have a positive driving effect on the theory and practice of FNN. Compared to traditional methods, this algorithm can better process and parse user behavior data, thereby more accurately constructing user behavior models. The innovation of this study lies in the use of the BP algorithm and LSE algorithm to optimize FNN, which can more accurately characterize user behavior characteristics and provide strong support for enterprises to develop more refined service marketing strategies. The value of this study lies in providing a new type of user behavior analysis method, which has higher accuracy and predictive ability compared to traditional analysis methods and can better meet the needs of modern data-driven society for user behavior analysis.

The research content consists of six sections. Section I introduce the background of the research and propose methods. Section II is a review of online user behavior research at home and abroad, summarizing and summarizing existing research, and pointing out the shortcomings of existing research. Section III mainly constructs a network user profile model using MGD-ANFIS. Section III (A) is the design of the MGD-ANFIS algorithm, which provides a detailed introduction to the design ideas and implementation process of the MGD-ANFIS algorithm and Section III (B) is based on the MGD-ANFIS algorithm and constructs a user profile model. Section V and Section VI provides the conclusion and acknowledgment respectively.

II. Related Works

With the acceleration of the Internet process, network user behavior has been digitized, and constructing user behavior profiles through complex algorithms is currently a hot research topic. Kumar S and other researchers designed a semi-local algorithm based on the correction degree centrality exclusion ratio to maximize influence through user behavior. It used the correction degree centrality exclusion ratio idea to ensure minimal overlap between the regions affected by selected diffusion nodes. Results showed that the algorithm output value outperformed other methods [4]. Zhao and other scholars designed a spatiotemporal gated network method to recommend interest points to network users. It can use interest point context prediction to assist the next interest point through joint learning and jointly train interest point context prediction and next interest point recommendation. The results showed that this method had high accuracy [5]. Wu et al. designed a recommendation algorithm to predict user behavior using anonymous sessions. The algorithm constructed sequences, captured them using graph neural networks, and combined session representations that met conditions through attention networks. The results showed that this algorithm outperformed other methods [6]. Kumar designed a speed learning-based classifier method to predict children's behavior based on their current emotions. The probability model was introduced into the deep learning classifier and multiple sample emotions were used for prediction. The results showed that the method had high recognition rate and prediction accuracy [7]. Chen and other researchers designed a graph convolutional network method based on linear residual to model the interaction behavior between users and the network. This method can alleviate the over-smooth problem in graph convolutional aggregation operations of sparse user and network project interaction data, and the results showed that this method was highly efficient [8]. Adam et al. designed an artificial intelligence-based chat system for real-time communication with users in an e-commerce environment. Through random online experiments, they empirically tested the impact of verbal anthropomorphic design prompts and entry techniques on user request compliance. The results showed that the system had good interaction effects with users [9].

Zhang and other scholars designed a multi-scale application programming interface graph sequence model to detect the dynamic behavior of users using malicious software. It concatenates graph features from different time periods and graph scales to detect whether the software is malicious. The results showed a good performance [10]. Zhang and other researchers designed a network attack detection method that combines traffic calculation and deep learning to detect unknown attacks in high-speed networks. It utilized sliding window flow data processing to achieve real-time detection and improved classification accuracy through deep trust networks and support vector machines. The results showed that this method had high efficiency and accuracy [11]. Boone et al. designed a data-driven technology using big data technology and the Internet of Things to better execute user management and meet user needs. It can collect and analyze large amounts of data in real time, and results showed high technology accuracy [12]. Ullah and other researchers designed an Apache web server intelligent intrusion detection system using machine learning methods to enhance the security of communication between suppliers and users. It utilized naive Bayesian machine learning algorithms for training, and results showed that this system had a high validation accuracy [13]. Chen et al. designed an attention evaluation method based on multimodal data and multi-scene modeling to evaluate users' psychological states and provide early warnings. The method analyzed the relationship between emotional data and attention in-depth and corrected labels with emotional data. Results showed a high prediction efficiency [14]. Scholars such as Cui designed a combined model using the time correlation coefficient and improved K-means clustering with cuckoo search to help users obtain real-time information. Through K-means clustering, similar users were gathered together and their behavior was analyzed. Results showed a high model accuracy [15].

In summary, many scholars have improved their understanding and predictive ability of user behavior through different algorithms and models, utilizing data on online user behavior. At the same time, they have also provided new solutions for areas such as network security, e-commerce interactivity, and mental health warning. However, these methods still have certain shortcomings in terms of algorithm generalization ability and model robustness. Therefore, the study utilizes GDM to optimize and learn the network parameters after BP and LSE optimization and then uses SCM to optimize the network structure, and design the MGD-ANFIS algorithm to construct a user behavior portrait model.

III. Method

The first section of this chapter is to improve FNN and design the MGD-ANFIS algorithm. The second section is to construct a user behavior profiling model using the MGD-ANFIS algorithm.

A. MGD-ANFIS Algorithm Design

FNN is a hybrid model that combines fuzzy logic and neural networks [16-17]. Its goal is to handle fuzzy and uncertain problems through the reasoning ability of fuzzy logic and the learning ability of neural networks. In traditional neural networks, both input and output are fixed values, while in the real world; many problems have ambiguity and uncertainty. FNN can better handle these problems by introducing the concepts of fuzzy sets and fuzzy reasoning. The basic structure of FNN includes the input layer, hidden layer, and output layer. Its inputs and outputs can be fuzzy sets, rather than just fixed values. Fuzzy sets are mathematical tools used to represent fuzziness and uncertainty, which can describe the degree of membership of a value within a certain range. When training FNN, fuzzy inference, and fuzzy set methods are usually used to define the objective function and error function of the network. The BP algorithm can update the weights and biases of the network to minimize the error function [18]. FNN has extensive applications in fields such as fuzzy control, pattern recognition, and decision support systems. The FNN structure is shown in Fig. 1.

In Fig. 1, the first four layers are the precursor network,

the first layer is the input layer, the second layer is the fuzzification layer, the third layer is the fuzzy rule calculation layer, and the fourth layer is the normalization layer. The last two layers are the post network, the fifth layer is the fuzzy rule output layer, and the sixth layer is the output single node layer. FNN can handle fuzzy and uncertain inputs, providing more flexible and robust decision-making and reasoning capabilities. However, due to the complexity and computational overhead of FNN, its training and inference process may be more time-consuming than traditional neural networks [19]. Therefore, the study used the BP algorithm and LSE algorithm to adjust the parameters of the FNN's antecedent and consequent networks to minimize errors. At the same time, GDM was used to optimize and learn the adjusted parameters, and SCM was used to optimize the structure of the FNN to shorten training time. MGD-ANFIS algorithm was designed. Firstly, define a fuzzy set, and the calculation formula is shown in Eq. (1).

$$A = \left\{ (x, \mu_A(x)) \middle| x \in X \right\} \tag{1}$$

In Eq. (1), A represents a fuzzy set, x represents any feature, X represents a set of all features, and $\mu_A(x)$ represents the membership function. In the input layer, components of the input vector are directly connected to the nodes. The commonly used membership function shapes are shown in Fig. 2.

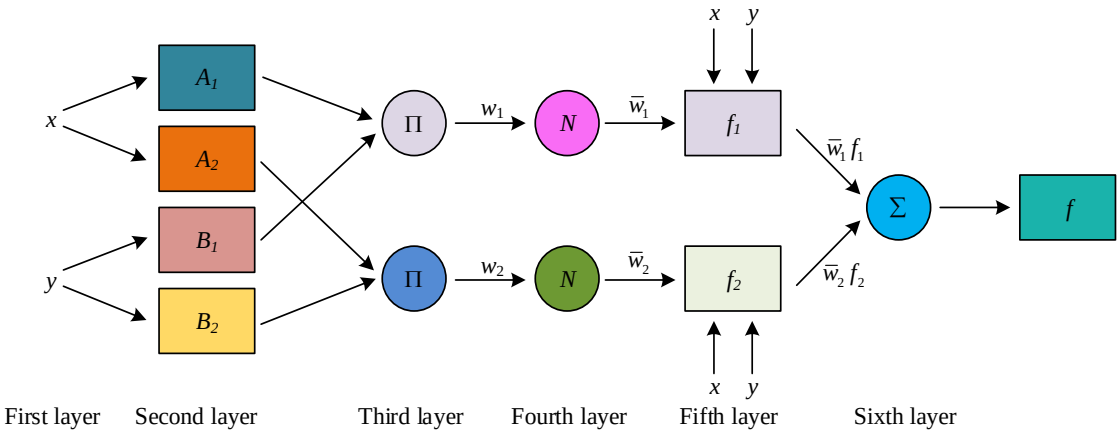

Fig. 1. Fuzzy neural network structure diagram.

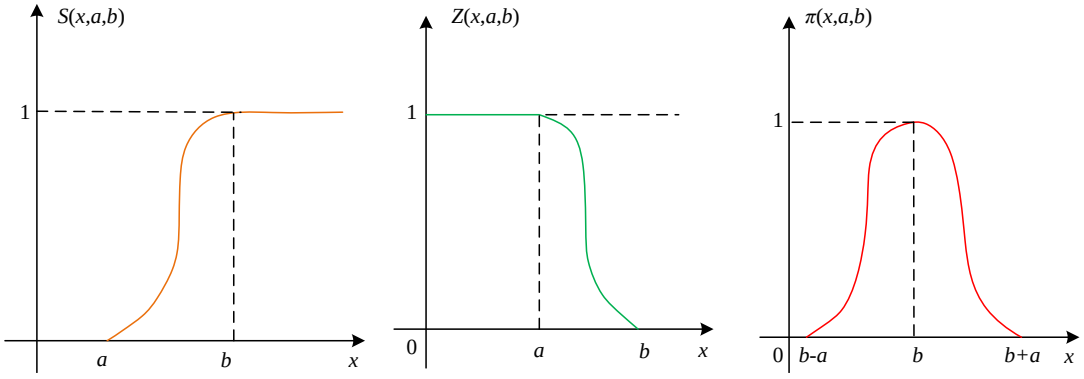

Fig. 2. Common membership function shapes.

In Fig. 2, common membership functions are divided into S-type, Z-type, and bell-type. The study adopts a Gaussian function with a bell shape as the membership function, and its expression is shown in Eq. (2).

$$f(x, \varepsilon, c) = e^{\frac{(x-c)^2}{2\varepsilon^2}} \tag{2}$$

In Eq. (2), c represents the membership function center value, and ε represents the membership function width. The next step is to map the input data onto a fuzzy aggregation through a fuzzification layer and describe the degree of membership of the input data through a membership function. The specific calculation method is shown in Eq. (3).

$$O_{1,i} = \begin{cases} \mu_{A_i}(x_1), i = 1,2 \\ \mu_{B_{i-2}}(x_2), i = 3,4 \end{cases} \tag{3}$$

In Eq. (3), $O_{1,i}$ represents the degree of membership of the input variable, x_1 and x_2 represent node i's input, and A_i and B_{i-2} represent two different fuzzy sets. The next step is to use the BP algorithm to adjust the antecedent parameters, where the momentum gradient descent method is used for BP propagation, as shown in Eq. (4).

$$\begin{cases} \dfrac{\partial E}{\partial w_{iq}} = -\delta_k^4 \times N_h^3 \\ \delta_h^3 = \delta_h^4 \times \dfrac{\sum_{s=1,s\neq h}^{m} N_h^3}{(\sum_{n=1}^{m} N_h^3)^2} \quad ,h=1,2,...,m, m=\prod_{j=1}^{n} m_j \\ \delta_{ij}^2 = \delta_h^3 \times s_{ij} \times e^{\frac{(x_j - c_{ij})^2}{\varepsilon_{ij}^2}} \end{cases} \tag{4}$$

In Eq. (4), ∂ represents the derivative, w_{iq} represents the point where the membership function value is 1, N_h^3 represents fuzzy rule calculation layer input value, δ represents the variance, s_{ij} represents the number of rules, k represents a certain rule, i and j represent nodes, and e represents natural constants. The specific expression form of the updated parameters is shown in Eq. (5).

$$\begin{cases} V_{dc} = \lambda \times dc + (1 - V_{dc}) \times dc, c = c + \alpha \times dc \\ V_{db} = \lambda \times db + (1 - V_{db}) \times db, b = b + \alpha \times db \\ V_{dw} = \lambda \times + (1 - V_{dw}) \times dw, w = w + \alpha \times dw \end{cases} \tag{5}$$

In Eq. (5), b represents the bias term, dc, db, and dw represent the differentiation of the parameters, V_{dc}, V_{db}, and V_{dw} represent the exponentially weighted average of each parameter, λ represents the momentum coefficient, and α represents the learning rate. Then the LSE algorithm can adjust consequent parameters, and the basic function of the LSE algorithm is shown in Eq. (6).

$$f(x) = a_1\varphi_1(x) + a_2\varphi_2(x) + ... + a_m\varphi_m(x) \tag{6}$$

In Eq. (6), $\varphi_m(x)$, $m=1,2,...,m$ represents a set of linearly independent functions, and a_m represents the undetermined coefficients. The calculation method for optimizing parameters using the least squares method is shown in Eq. (7).

$$\min F(x) = \sum_{i=1}^{m} f_i^2(x) \tag{7}$$

In Eq. (7), min represents the minimum value and $F(x)$ represents the objective function. The next step of the SCM algorithm is to cluster the input feature values to find the clustering center, thereby determining fuzzy rules and membership functions. The density index calculation method for feature data is shown in Eq. (8).

$$D_i = \sum_{j=1}^{n} \exp\left(\frac{\left\|x_i - x_j\right\|^2}{(\gamma_\alpha / 2)^2}\right) \tag{8}$$

In Eq. (8), D_i represents the density index, exp represents the exponential function with a base, and γ_α represents a positive number that can define a neighborhood of feature point x_i. The density index calculation method for each feature point is shown in Eq. (9).

$$\begin{cases} D_i = D_i - D_{cl} \times \exp\left[-\left(\dfrac{\left\|x_i - x_j\right\|^2}{(\gamma_\beta / 2)^2}\right)\right] \\ \gamma_\beta = k\gamma_\alpha \end{cases} \tag{9}$$

In Eq. (9), D_{cl} represents the density index corresponding to each cluster center x_{cl}, while γ_β and k both represent a positive number. The SCM clustering process is shown in Fig. 3.

In Fig. 3, the parameters are first initialized, then the density indicators of each sample point are calculated. Next, the density indicators of the remaining sample points are corrected, and whether the termination condition is met can be finally determined. If it is met, the density indicator can be output. Otherwise, recalculate. Fuzzy rules are calculated after SCM clustering, as shown in Eq. (10).

$$O_{2,i} = \mu_{Ai}(x_1)\mu_{Bi}(x_2), i = 1,2 \tag{10}$$

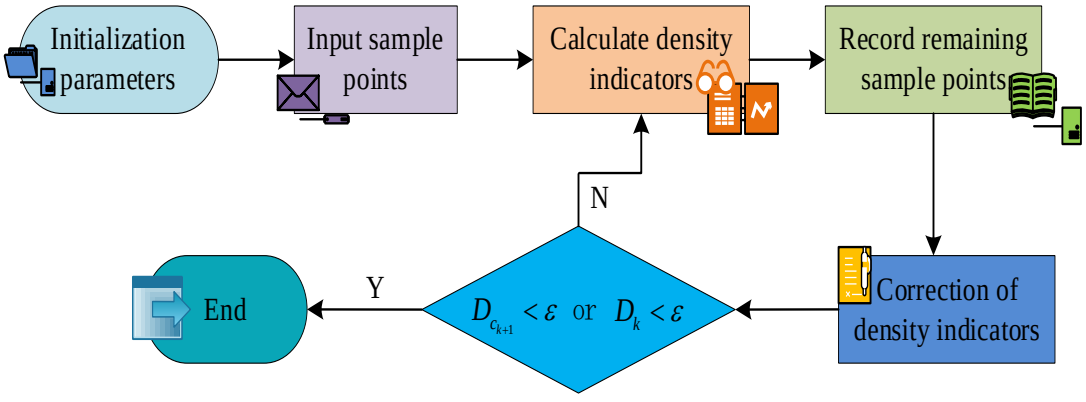

Fig. 3. SCM clustering process.

In Eq. (10), $O_{2,i}$ represents the strength of the fuzzy rule. The next step is to normalize, and the calculation method is shown in Eq. (11).

$$y = \frac{x - MinValue}{MaxValue - MinValue} \tag{11}$$

In Eq. (11), x represents the original value, y represents the converted value, $MaxValue$ represents the maximum feature value, and $MinValue$ represents the minimum feature value. The normalization results are shown in Eq. (12).

$$\begin{cases} O_{3,i} = \dfrac{w_i}{w_1 + w_2}, i = 1,2 \\ w_i = O_{2,i} \end{cases} \tag{12}$$

In Eq. (12), $O_{3,i}$ represents the normalized fuzzy rule, and w represents the numerical value of the fuzzy rule. The next step is to output the fuzzy rule, as shown in Eq. (13).

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x_i + q_i x_2 + r_i), i = 1,2 \tag{13}$$

In Eq. (13), f_i represents the output function, and p_i, q_i, and r_i represent the node parameters. The total output can be obtained from the node output as shown in Eq. (14).

$$O_{5,i} = \sum \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{14}$$

In Eq. (14), $O_{5,i}$ represents the total output of MGD-ANFIS. When determining the parameters of the current component network, the output expression is shown in Eq. (15).

$$O'_{5,i} = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + \bar{w}_1 r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + \bar{w}_1 r_2 \tag{15}$$

In Eq. (15), $O'_{5,i}$ represents the final output value.

B. Construction of Network User Behavior Portrait Model

The construction of a network user behavior profiling model aims to identify unknown intrusions and profile user behavior. The MGD-ANFIS algorithm is used to construct a user behavior profiling model. This model can efficiently collect network user behavior data, and through filtering and compression, recombine feature vectors to generate records with various meanings to accurately identify malicious users and provide early warnings. The specific framework of the model is shown in Fig. 4.

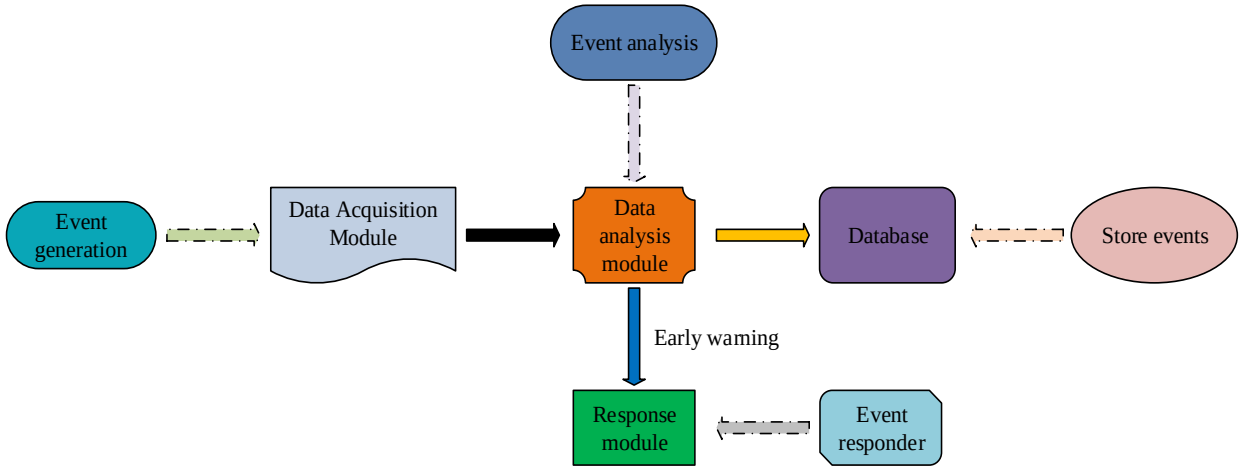

Fig. 4. User behavior portrait model framework.

In Fig. 4, the model includes four parts: data collection, data analysis, response module, and database. The data collection module is to build a user behavior database, and the integrity of the obtained database is related to the accuracy of user behavior judgment. It refers to the collection and processing of user behavior data on the network, and the use of the MGD-ANFIS algorithm for modeling and learning, by adjusting network structure and parameters to adapt to different user behavior characteristics. Data analysis is the most important component of user behavior profiling models, which obtains data from the data collection module and analyzes it, including data cleaning, data preprocessing, and feature extraction. Data cleaning refers to denoising, deduplication, recombination, and restoration of collected data to ensure data quality. Data preprocessing involves normalizing the data, and feature extraction involves extracting an appropriate number of features from the original data to distinguish whether user behavior is normal. The response module is to record, prevent, and alarm users with abnormal behavior, and can also be expanded based on the current user's input or environmental variables. The database mainly records users with abnormal behavior. Meanwhile, in user behavior profiling models, statistical feature quantity is an important factor affecting the performance of user behavior detection [20]. The research aims to determine whether a user's behavior is normal or malicious through a small number

of statistical features, and statistical features need to be designed for different types of attacks. The current common types of attacks include attacking through network vulnerabilities, exploiting network protocol flaws, and using illegal and irregular operations to enter the system. Therefore, the study designs feature quantities from three aspects: content, time traffic, and host traffic, as shown in Fig. 5.

In Fig. 5, feature statistics are in the data restoration section of data collection, which includes basic feature quantities, content feature quantities, time traffic characteristics, and host traffic characteristics. The basic feature quantity mainly counts the connection time, bytes, and network protocol types between users and the cloud. The content feature quantity mainly counts the number of access privacy and login failures. The time traffic characteristic mainly counts the connections to the same host and server. The host traffic characteristic mainly counts the connections to the same host and server in a relatively small number of connections. After designing each module, parameter settings are required. The parameters studied in this study mainly include input layer nodes, membership function, number of fuzzy subsets, and number of output layer nodes, training frequency, and radius of subtractive clustering. Finally, the obtained samples are trained and tested, and the specific process is shown in Fig. 6.

In Fig. 6, the training process and testing process correspond to the set training data and testing data, respectively. During the training process, the data consists of users with normal behavior and users with abnormal behavior, and the expected output is sent to the data analysis module as its input. Before training, it is necessary to initialize the parameters, LSE algorithm is used to identify the subsequent parameters, and continuously adjust precursor network parameters through the BP algorithm to minimize the difference between the predicted and actual output. During the detection process, data composition is consistent with the training process, but the test data is directly sent to the data analysis module as its input. After analysis, the detection rate and false alarm rate are calculated separately and compared with the results of the training data, to evaluate user behavior profile model performance based on the MGD-ANFIS algorithm.

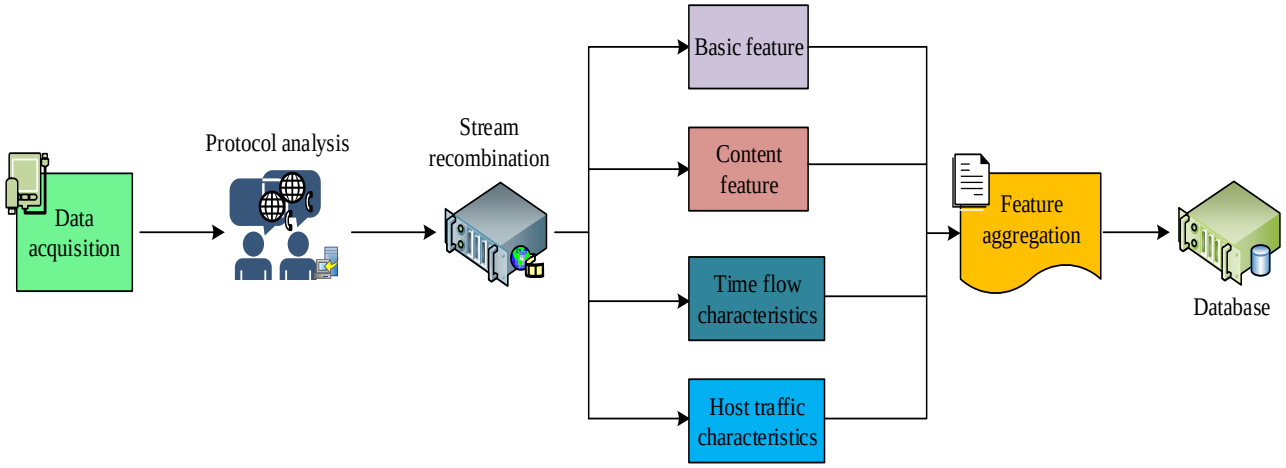

Fig. 5. Overall composition of characteristic quantities.

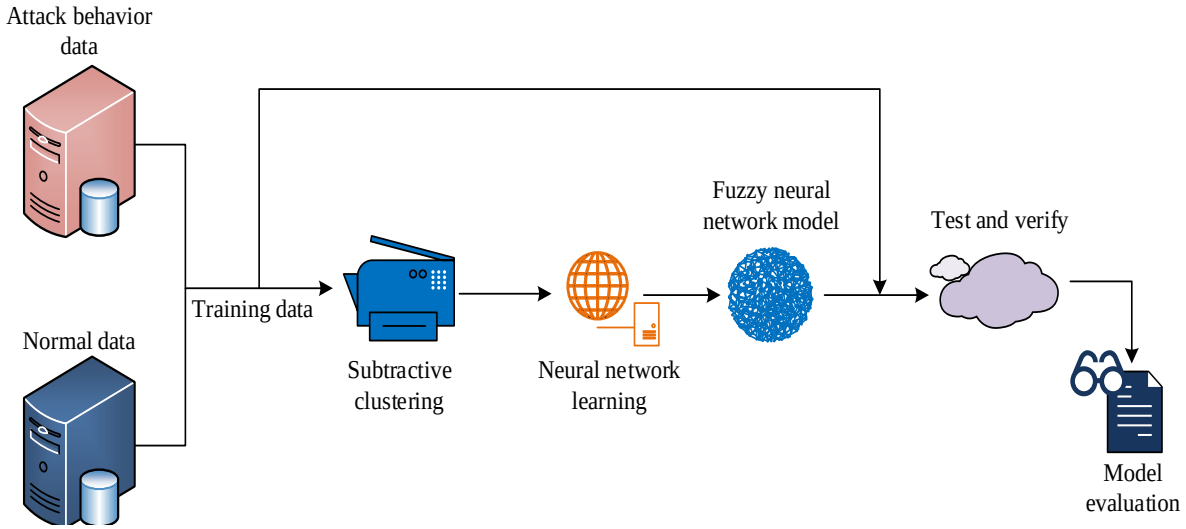

Fig. 6. The process of obtaining feature statistics.

IV. Results and Discussion

Section III (A) of this chapter analyzes the performance of the designed MGD-ANFIS algorithm, and the second section analyzes the actual application effect of the user behavior portrait model designed on the basis of the MGD-ANFIS algorithm.

A. MGD-ANFIS Algorithm Performance Analysis

To verify the designed MGD-ANFIS's performance, this study first generated 1000 pairs of datasets using the Sphere test function, and divided them into training and testing sets in 4:1. The maximum number of iterations was set to 500, and simulation comparisons were made with the MGD-ANFIS algorithm and ANFIS algorithm, respectively. The optimization results are shown in Fig. 7.

From Fig. 7(a), it can be seen that the optimization results of the ANFIS algorithm have a low fit with the test function. In Fig. 7(b), the optimization trend of the MGD-ANFIS algorithm was basically consistent with the results of the test function, with a high degree of fit. The above results indicated that the MGD-ANFIS algorithm had high accuracy and proved its effectiveness. The next step was to calculate the recall, accuracy, and F1 score of the MGD-ANFIS algorithm separately, and compare them with the ANFIS algorithm and FNN algorithm. The results are shown in Fig. 8.

In Fig. 8, the recall, accuracy, and F1 score of the MGD-ANFIS algorithm are 0.23, 0.99, and 0.20, respectively. The recall, accuracy, and F1 score of the ANFIS algorithm are 0.17, 0.96, and 0.13, respectively. The recall, accuracy, and F1 score of the FNN algorithm are 0.15, 0.92, and 0.12, respectively. Analysis shows that the MGD-ANFIS algorithm had a recall rate of 0.07 and 0.09 higher than the other two algorithms, an accuracy rate of 0.03 and 0.07 higher than the other two algorithms, and an F1 score of 0.07 and 0.08 higher than the other two algorithms. The above results demonstrate that the MGD-ANFIS algorithm had good overall performance. Finally, the MGD-ANFIS algorithm was used to perform error analysis on the training and testing sets, and compared with the ANFIS and FNN algorithms. The results are shown in Fig. 9.

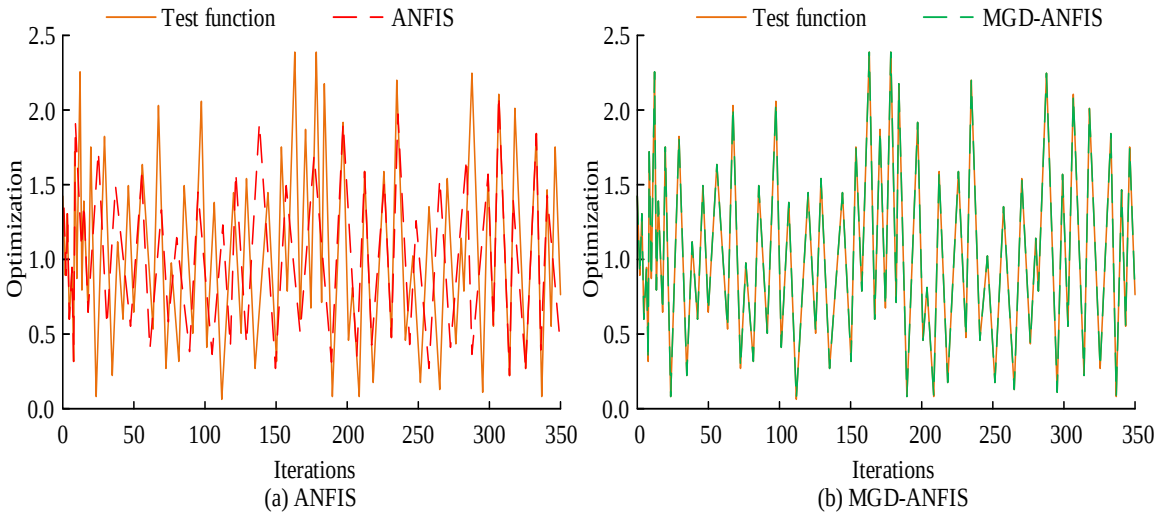

Fig. 7. MGD-ANFIS and ANFIS simulation comparison.

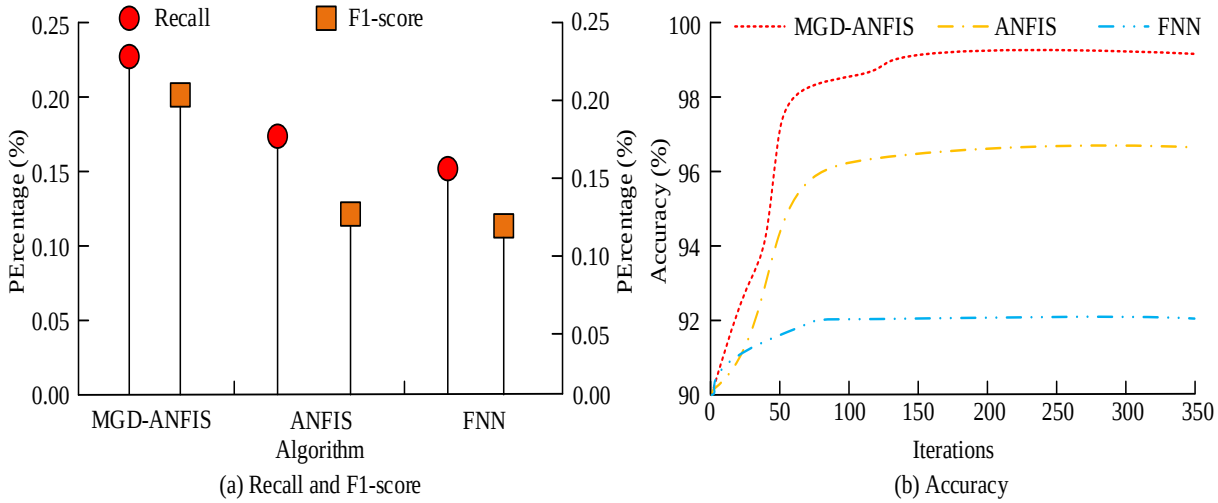

Fig. 8. Recall rate, accuracy rate, and f1 score of different algorithms.

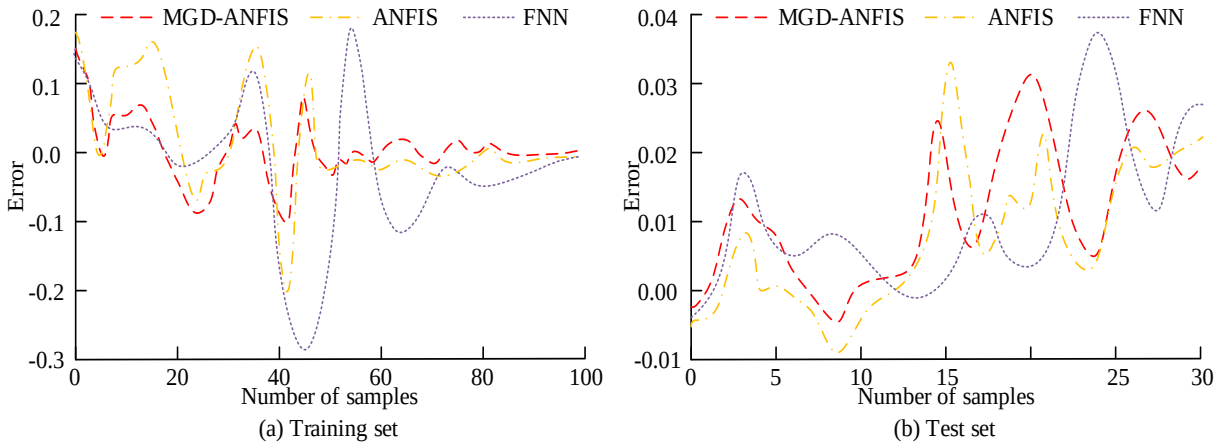

Fig. 9. Error of different algorithms on training and testing sets.

In Fig. 9 (a), in the training set error analysis, the error range of the MGD-ANFIS algorithm is between -0.10 and 0.10, the error range of ANFIS is between -0.20 and 0.18, and the error range of FNN is between -0.30 and 0.20. In Figure 9(b), in test set error analysis, the error range of the MGD-ANFIS algorithm is between -0.005 and 0.03, the error range of ANFIS is between -0.008 and 0.035, and the error range of FNN is between -0.01 and 0.04. Analysis shows that in the error analysis of the training and testing sets, the error range of the MGD-ANFIS algorithm was smaller than the other two algorithms, further indicating its high accuracy and demonstrating its good performance.

B. *Effect Analysis of User Behavior Portrait Model*

To test the performance of the designed user behavior profiling model, the study first conducted simulation experiments using the KDD99 dataset. 5 sets of data were selected with 4 training data and 1 testing data, which contained malicious user attack behavior. In training set 1, the number of normal and abnormal data was equal, and in training set 2 and the test set, the number of normal and abnormal data was also equal. Training set 2 was smaller than the training set 1 and greater than the test set. In training set 3, there was more normal data than abnormal data, while in training set 4, there was less normal data. In the network configuration, the output layer node was set to 1, with an output of 1 indicating abnormal behavior and an output of 0 indicating normal behavior. The allowable error was set to 0.2, and the clustering radius was set to 0.5. It was recommended to train the network for 50 iterations. Each training set was mixed with the test set to detect malicious user attack behavior, and the results are shown below.

TABLE I. MALICIOUS USER ATTACK BEHAVIOR IN TRAINING AND TESTING SETS

Dataset	Test error	Training Error	Training false alarm rate	Training detection rate	Test false alarm rate	Test detection rate
Training 1	0.1297	0.2673	99.2	4.9	96.3	6.3
Training 2	0.1527	0.2628	99.4	4.5	95.9	5.5
Training 3	0.2401	0.2654	98.3	4.7	94.6	6.7
Training 4	0.1399	0.2493	99.3	5.2	95.8	6.9

In Table I, the four training sets' errors are smaller than those of the test set, and the detection rate and false alarm rate are both higher than those of the test set. However, overall, different combinations of training and testing sets had higher detection rates and lower false positives and errors, indicating that the designed model had higher adaptability. The next step was to calculate the decision values for normal user behavior, vulnerability-based attacks (attack behavior 1), and network protocol defect-based attacks (attack behavior 2), as shown in Fig. 10.

In Fig. 10, users with normal behavior have a relatively small fluctuation in the judgment curve, with a maximum judgment value of 1.00, a minimum judgment value of 0.95, and an average judgment value of approximately 0.98. Attack behavior 1 involved exploiting system vulnerabilities by sending requests to the host, with a maximum decision value of 1.00, a minimum decision value of 0.58, and an average decision value of approximately 0.79. Attack behavior 2 utilized flaws in network protocols to attack, which was significantly different from normal user modes. Its maximum decision value was only 0.80, the minimum decision value was 0.10, and the average decision value was 0.45. Overall, the designed user behavior profiling model effectively distinguished between users with normal and abnormal behavior, and further proved its effectiveness. Finally, the detection rate of user behavior profiling models based on different algorithms was tested using ROC curves, and the results are shown in Fig. 11.

In Fig. 11, the designed MGD-ANFIS-based user behavior profiling model has a maximum detection rate of 1.000, a

minimum of 0.950, and an average of 0.975. The maximum of the ANFIS-based user behavior profiling model was 0.98, the minimum was 0.84, and the average was 0.91. The maximum of the user behavior profiling model based on FNN was 0.98, the minimum was 0.66, and the average was 0.82. Analysis shows that the average detection rate of the MGD-ANFIS-based user behavior profiling model was 0.065 and 0.155 higher than the other two models, respectively, proving its high detection accuracy.

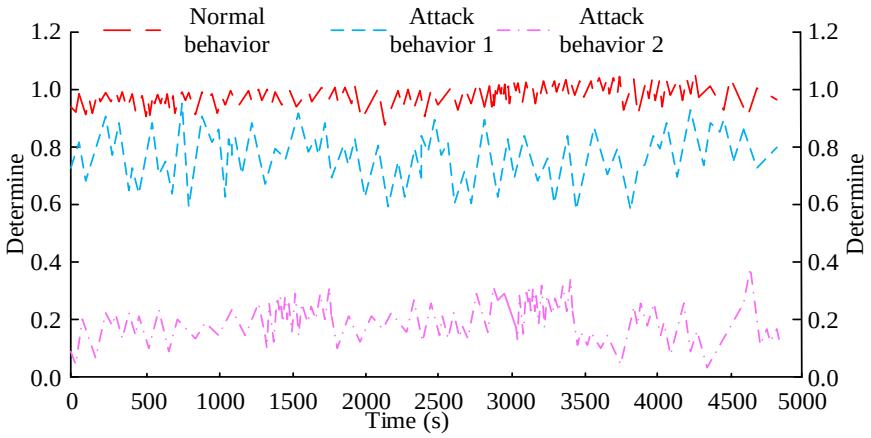

Fig. 10. Judgment values for three behaviors.

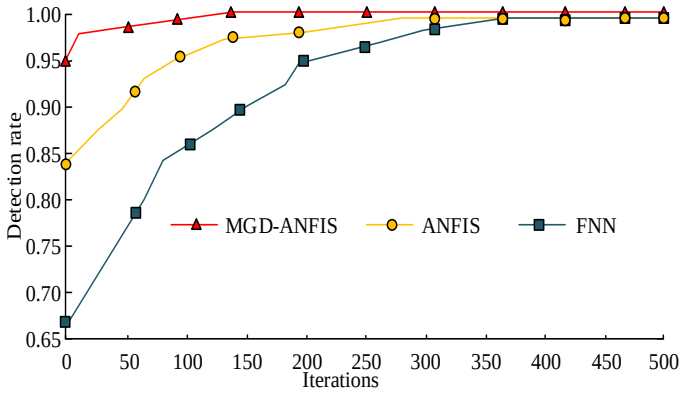

Fig. 11. Detection rate of user behavior profiling models using different algorithms.

V. Conclusion

With the popularization of mobile devices and the advancement of network technology, user behavior data have been generated in the network. How to better distinguish user behavior has become a focus of research by relevant personnel. In order to better detect user behavior, the study first adjusts the network parameters of FNN using the BP algorithm and LSE algorithm, then optimizes and learns the adjusted parameters using GDM, and optimizes the structure of FNN through SCM to shorten training time. MGD-ANFIS algorithm is designed, and finally, MGD-ANFIS is applied to user behavior profiling and a model is constructed. The results showed that in the simulation comparison, the trend of the optimization results of the MGD-ANFIS algorithm and the fitting degree of the test function were higher than those of the ANFIS algorithm, indicating its high accuracy and proving its effectiveness. In the calculation of recall, accuracy, and F1 score, the three indicator values of the MGD-ANFIS algorithm were 0.20, 0.99, and 0.20, respectively. The three indicator values of the ANFIS algorithm were 0.13, 0.96, and 0.13, respectively. The three indicator values of FNN were 0.11, 0.92, and 0.12, respectively. The three indicator values of the MGD-ANFIS algorithm were all higher than other algorithms, proving its good comprehensive performance. In the simulation experiment of the KDD99 dataset, different combinations of training and testing sets had higher detection rates and smaller false positives and errors, indicating that the designed user profile model had high adaptability. In the calculation of decision values, the average decision value for users with normal behavior was about 0.98, the average decision value for attack behavior 1 was 0.79, and the average decision value for attack behavior 2 was 0.45, proving the effectiveness of the designed model. The study only analyzed vulnerability-based attacks and protocol defect-based attacks, which had a certain impact on behavior judgment. Further exploration will be conducted in related aspects in the future. The study will conduct a more in-depth analysis of attack behavior, including the attacker's behavior patterns, attack time, and frequency, in order to better understand the attacker's motivation and strategy. Meanwhile, when constructing user behavior profiles, the quality and completeness of data have a significant impact on the accuracy and reliability of the results. Therefore, in future

research, it is necessary to consider the quality and completeness of data more comprehensively.

VI. Acknowledgment

The research is supported by: Jiangsu Qinglan Project QLGG-2022-03; Doctor Program of Xuzhou College of Industrial Technology XGY2021EA01; Industrial R&D projects of Xuzhou College of Industrial Technology XGY2022CXZ06.

References

[1] Vitiello M, Walk S, Helic D, Chang V, Guetl C. User Behavioral Patterns and Early Dropouts Detection: Improved Users Profiling through Analysis of Successive Offering of MOOC. J. Univers. Comput. Sci., 2018, 24(8): 1131-1150.

[2] Fei J, Wang Z, Liang X, Feng Z, Xue Y. Fractional sliding-mode control for microgyroscope based on multilayer recurrent fuzzy neural network. IEEE transactions on fuzzy systems, 2021, 30(6): 1712-1721.

[3] Garud K S, Jayaraj S, Lee M Y. A review on modeling of solar photovoltaic systems using artificial neural networks, fuzzy logic, genetic algorithm and hybrid models. International Journal of Energy Research, 2021, 45(1): 6-35.

[4] Kumar S, Lohia D, Pratap D, Krishna A, Panda B S. MDER: modified degree with exclusion ratio algorithm for influence maximisation in social networks. Computing, 2022, 104(2): 359-382.

[5] Zhao P, Luo A, Liu Y, Xu J, Li Z, Zhuang F, Zhou X. Where to go next: A spatio-temporal gated network for next poi recommendation. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(5): 2512-2524.

[6] Wu S, Tang Y, Zhu Y, Wang L, Xie, X S, Tan T. ession-based recommendation with graph neural networks//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(1): 346-353.

[7] Kumar D T S. Construction of hybrid deep learning model for predicting children behavior based on their emotional reaction. Journal of Information Technology and Digital World, 2021, 3(1): 29-43.

[8] Chen L, Wu L, Hong R, Zhang K, Wang M. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(1): 27-34.

[9] Adam M, Wessel M, Benlian A. AI-based chatbots in customer service and their effects on user compliance. Electronic Markets, 2021, 31(2): 427-445.

[10] Zhang Z, Li Y, Wang W, Song H, Dong H. Malware detection with dynamic evolving graph convolutional networks. International Journal of Intelligent Systems, 2022, 37(10): 7261-7280.

[11] Zhang H, Li Y, Lv Z, Sangaiah A K, Huang T. A real-time and ubiquitous network attack detection based on deep belief network and support vector machine. IEEE/CAA Journal of Automatica Sinica, 2020, 7(3): 790-799.

[12] Boone T, Ganeshan R, Jain A, Sanders N R. Forecasting sales in the supply chain: Consumer analytics in the big data era. International Journal of Forecasting, 2019, 35(1): 170-180.

[13] Ullah M U, Hassan A, Asif M, Farooq M S, Saleem M. Intelligent Intrusion Detection System for Apache Web Server Empowered with Machine Learning Approaches. International Journal of Computational and Innovative Sciences, 2022, 1(1): 21-27.

[14] Chen M, Cao Y, Wang R, Li Y, Wu D, Liu Z. DeepFocus: Deep encoding brainwaves and emotions with multi-scenario behavior analytics for human attention enhancement. IEEE Network, 2019, 33(6): 70-77.

[15] Cui Z, Xu X, Fei X, Cai X, Cao Y, Zhang W, Chen J. Personalized recommendation system based on collaborative filtering for IoT scenarios. IEEE Transactions on Services Computing, 2020, 13(4): 685-695.

[16] Debnath S. Fuzzy quadripartitioned neutrosophic soft matrix theory and its decision-making approach. Journal of Computational and Cognitive Engineering, 2022, 1(2): 88-93.

[17] Saeed M, Ahmad M R, & Rahman A U. Refined Pythagorean Fuzzy Sets: Properties, Set-Theoretic Operations and Axiomatic Results. Journal of Computational and Cognitive Engineering, 2022, 2(1), 10–16.

[18] Hou S, Chu Y, Fei J. Adaptive type-2 fuzzy neural network inherited terminal sliding mode control for power quality improvement. IEEE transactions on industrial informatics, 2021, 17(11): 7564-7574.

[19] Yao Q. Adaptive fuzzy neural network control for a space manipulator in the presence of output constraints and input nonlinearities. Advances in Space Research, 2021, 67(6): 1830-1843.

[20] Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2019, 2(1): 1-22.