Campus Network Intrusion Detection Based on Gated Recurrent Neural Network and Domain Generation Algorithm

Qi Rong[1*], Guang Zhao[2]
Party Committee Organization Department, Jilin Institute of Architecture and Technology, Changchun, China[1]
School of Statistics, Shandong University of Finance and Economics, Jinan, China[2]

***Abstract*—Network attacks are diversified, rare and Universal generalization. This has made the exploration and construction of network information flow packet threat detection systems, which becomes a hot research topic in preventing network attacks. So this study establishes a network data threat detection model based on traditional network threat detection systems and deep learning neural networks. And convolutional neural network and data enhancement technology are used to optimize the model and improve rare data recognizing accuracy. The experiment confirms that this detection model has a recognition probability of approximately 11% and 42% for two rare attacks when N=1, respectively. When N=2, their probabilities are 52% and 78%, respectively. When N=3, their recognition probabilities are approximately 85% and 92%, respectively. When N=4, their recognition probabilities are about 58% and 68%, respectively, with N=3 having the best recognition effect. In addition, the recognition efficiency of this model for malicious domain name attacks and normal data remains around 90%, which has significant advantages compared to traditional detection systems. The proposed network data flow threat detection model that integrates Gated Recurrent Neural Network and Domain Generation Algorithm has certain practicality and feasibility.**

***Keywords*—Gated recurrent; domain generation algorithm; campus network; threat detection; neural network**

I. Introduction

Technological progress has led to an increasing demand for information technology and a high informatization degree. Therefore, the internet has become an important way for people to improve their efficiency, quality of life, and increase personal income. Internet popularization has made people's lives more and more transparent. People rely heavily on the computer internet in many fields such as daily life, economic management, and financial investment [1]. The global Internet economy has even accounted for 10% of the global GDP, reaching the level of more than ten trillion US dollars. However, while the network has brought us convenience, many malicious attackers will use various means such as Botnet, system vulnerabilities, malicious domain names and trojans to attack, steal, destroy or modify data on the local network without authorization [2]. As of 2022, the losses caused by global cybercrime to people have exceeded $6 trillion, and these cyber threats are increasing exponentially over time [3]. Currently, campus education networks and research departments are the primary targets of attackers, with each institution or campus network facing an average of thousands of malicious network attacks per week, an increase of at least 50% compared to 2021 [4]. The existing attack methods for network data have the characteristics of diversification, uniqueness, and fast update. Traditional intrusion detection systems are often unable to effectively identify new and unknown attacks, because they rely on known rules and features [5]. For example, for new attacks such as zero-day vulnerabilities and advanced persistent threats, traditional intrusion detection may not be able to detect and alert in time. And traditional intrusion detection may fail to detect attacks based on covert communication, such as steganography or encrypted communication, and even generate a large number of false positives, wasting time and resources. Therefore, traditional intrusion detection systems have great limitations when facing new attacks, high false positive rate, complex environment and lack of context information. In this context, research attempts to integrate deep learning neural networks and data augmentation techniques into the traditional Network Intrusion Detection System (NIDS) to make it more intelligent and self-learning, to enhance the recognition probability of new rare attack methods.

This study conducted technical exploration and analysis from four aspects. Firstly, the relevant research on the current network DS threat detection system was discussed and summarized. Secondly, the comprehensive applications of Gated Recurrent Neural Network (GRNN), Convolutional Neural Network (CNN), and Domain Generation Algorithm (DGA) were analyzed, including the construction of a network data threat detection system. Then, experimental verification and data comparison analysis were conducted on the data stream (DS) threat detection model of network. Finally, there is a comprehensive overview of entire article and a reflection and summary of its shortcomings.

II. Related Works

While internet popularization has made people's lives more convenient, network data threats and intrusions have also become increasingly common. Building a DS detection model has become a hot research and exploration field for some experts at home and abroad. Its characteristics include fast and accurate learning, recognition, and fine segmentation of campus network and other DS quantities. Zhou et al. proposed a hierarchical adversarial attack generation method based on the graph neural network (NN) for the intelligent intrusion detection (ID) problem of Internet of Things (IoT), which

improved the system's recognition accuracy against network attacks [6]. The vulnerabilities in IoT are prone to attacks and other issues. In this regard, Nimbalkar and Kshirsagar proposed an attack data detection and comparison system based on the ID system of Feature selection and information gain method, which improved the identification efficiency of denial of service and other attacks [7]. After the automobile network is connected, it is vulnerable to network attacks and accidents. In response, Moubayed et al. proposed a multi-layer hybrid ID system based on feature based ID systems and anomaly based ID systems, thereby improving the recognition efficiency against known and unknown attacks [8]. Guo et al. proposed a spam detection method based on pre trained bidirectional encoder representation and machine learning (ML) algorithm to address the issue of spam detection. Two datasets were used for performance testing in the experiment, confirming that this method effectively improves the detection efficiency of spam [9]. Hidayat et al. proposed a new network ID technology for data detection in network attacks, based on the ML model and integrating multiple technologies, which improved the detection efficiency of network attacks [10].

In addition, Binbusayyis and Vaiyapuri constructed a joint optimization ID framework based on classifiers such as convolutional encoders to address network security issues. This effectively improves the detection ability for unknown attacks [11]. Wang et al. proposed a new ID model for network intrusion based on ML and Decision boundary, integrating popular evaluation methods and ID system models. This increases the recognition probability against boundary attacks [12]. Krishnaveni and others proposed a new attack classification method based on the univariate integration Feature selection technology and classification technology to solve the problem that cloud computing servers are vulnerable to attacks. This improves the detection efficiency of the ID model for attack data [13]. Azizan et al. proposed a new ID method for identifying attack data in large amounts of data, based on ML and incorporating algorithms such as decision jungle. This improves the identification efficiency of attack data in big data [14]. Aimed at the identification problem under multiple attacks on the network, Almomani combines Particle swarm optimization algorithm based on ID system and proposes a new ID model. This effectively improves the recognition efficiency for multiple attacks [15]. Rashid et al. proposed a new ID model based on ML, which integrates tree based stack integration technology to address the classification problem of anomalies and normals. This interference improved the recognition efficiency of network abnormal traffic [16].

From the research from various countries, ID systems have low efficiency in identifying multiple types of network attacks. Most studies only focus on improving ID system's efficiency in identifying attack types. They overlooked the intelligence of ID system and improved learning efficiency for unknown attacks and recognition efficiency for rare attacks. Therefore, the deep detection network model developed using GRNN and DGA has a certain degree of innovation.

III. Design and Implementation of NIDS

Unlike traditional threat detection systems, the detection model using GRNN model and DGA fusion has a certain innovation. Therefore, to ensure data detection model's detection accuracy can continuously be deeply refined, the model design and implementation are particularly important. Therefore, this section mainly analyzes the model implementation principle and system construction.

A. Network Threat Detection Model and NN Technology

To conduct real-time monitoring and analysis of DS generated locally on campus network and DS passing through campus local network, NIDS is needed to timely identify network attacks and adopt corresponding strategies [17]. Fig. 1 shows the detection process of this system against DS threats.

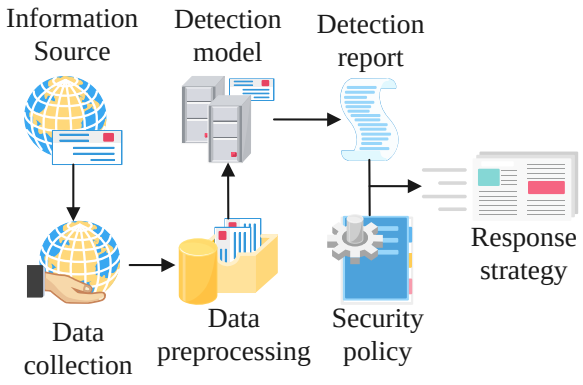

Fig. 1. Flowchart of data flow threat detection.

According to Fig. 1, the entire NIDS detection process mainly includes three steps: information flow data capture, network packet structure content analysis, and abnormal activity strategy response. Among them, data capture mainly utilizes packet capture tools for data capture. Data analysis mainly uses models to make detailed judgments and intelligent decisions. Policy response is mainly aimed at the network abnormal activities by saving the characteristic code, sending information or screen display abnormalities to remind and cooperate with the Network engineer to carry out human intervention to eliminate hidden dangers. Fig. 2 shows the detection logic model of NIDS for network DS.

Through Fig. 2, NIDS accumulates the original feature library based on the initial data and supplements and corrects the feature library through continuous learning and training. Then it makes judgments and decisions on new real-time data on the foundation of vast feature library. Among them, learning historical training experience data is particularly important for NIDS, so deep learning Recurrent Neural Network (RNN) is needed to make NIDS more intelligent [18]. RNN model mainly collects and analyzes the texture characteristics of data through a multi-layer NN composed of simulated neurons. So it can excavate the original feature expressions behind diverse data and make intelligent judgments on richer new data. Fig. 3 is the schematic diagram of RNN.

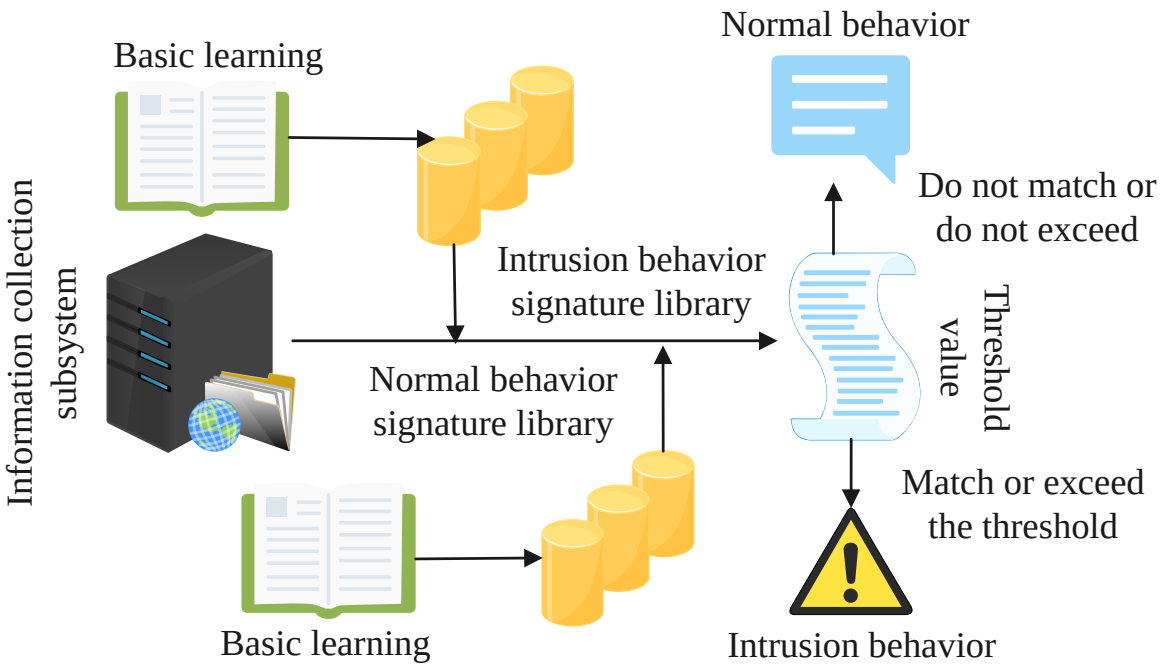

Fig. 2. Data flow detection model.

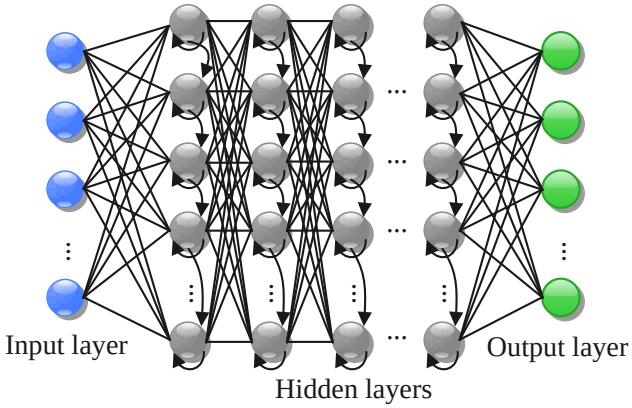

Fig. 3. Recurrent neural network models for deep learning.

From Fig. 3, RNN is a memory feedback bidirectional Transitive model composed of one input layer, multiple hidden layers and one output layer. Each layer of neurons can receive signals from the same or previous layer of neurons and perform nonlinear transformations to accumulate and output to the next layer until the output layer is reached. Due to the tendency of RNN to form gradient decay or gradient explosion when the time difference is large or small, it is difficult for RNN to obtain long-term or short-term dependencies in time series in practical applications. To this end, Gated Recurrent Unit (GRU) should be introduced into RNN to improve its memory unit and better capture long-term or short-term dependencies in time series. Eq. (1) represents its connection relationship [19].

$$\begin{cases} f_t = S(D_{xf}x_t + D_{yf}y_{t-1} + D_{zf}z_{t-1}) \\ i_t = S(D_{xi}x_t + D_{yi}y_{t-1} + D_{zi}z_{t-1}) \\ z_t = f_t Hz_{t-1} + i_t H\tanh(D_{xz}x_t + D_{yz}y_{t-1}) \\ O_t = S(D_{xO}x_t + D_{yO}y_{t-1} + D_{zO}z_{t-1}) \\ y_t = O_t H\tanh(z_t) \end{cases} \quad (1)$$

In Eq. (1), f refers to the forgetting gate. S refers to a growth curve function (sigmoid function). D refers to a pending parameter. x refers to the input value. y refers to the output value. z refers to cell state value. t and $t-1$ represent the current and the previous time, respectively. i refers to the input gate. H refers to the image matrix transformation Hadamard matrix. O refers to the output gate. tanh() refers to a hyperbolic tangent function. In addition, the lack of feedback from neurons themselves is a one-way transmission lacking error adjustment mechanisms. Therefore, to make the gradient descent optimization algorithm and the error Backpropagation work and improve the accuracy of DNN, it is necessary to quantify error information between the estimated and the actual values with cost function. And because NN input values are generally nonlinear discrete values, it is necessary to introduce a linear regression Softmax function to better predict the discrete results. Only by accurately classifying the results according to different weights can more accurate cost function values be obtained. Eq. (2) is the mathematical expression of Softmax value.

$$\begin{cases} \alpha(x)_j = e^{x_j} \Big/ \sum_{k=1}^{k} e^{x_k}, j = 1,..,k \\ f_\beta(x^i) = \begin{bmatrix} e^{\beta_0^T x^i} \\ e^{\beta_1^T x^i} \\ ... \\ e^{\beta_{k-1}^T x^i} \end{bmatrix} \Big/ \sum_{j=0}^{k-1} e^{\beta_j^T x^i} \end{cases} \quad (2)$$

In Eq. (2), x refers to the vector value. $\alpha()$ refers to a dimensional vector. e refers to the natural base. k refers to the maximum dimension value. j refers to the current dimension. β refers to an undetermined parameter. $f()$ refers to the probability function. T refers to the temperature coefficient used to adjust curve smoothness. Eq. (3) is the cross entropy of cost function.

$$H(p,q) = -\sum_{i=1}^{N} p(x_i) \log q(x_i) \quad (3)$$

In Eq. (3), H is the cross entropy of the cost function. p refers to the true distribution probability. q refers to the fitted distribution probability. x refers to the sample space. N is represented as a variable coefficient. Cross entropy mainly expresses the information quantity fully used to eliminate uncertainty. Then, the gradient descent algorithm was used to search for various parameters that optimize cost function. Finally, Backpropagation is used to transmit the optimized error information to model's initial neurons. So it can correct whole model's error to improve its accuracy. The model's excessive reliance on its own training data to fit iterative loop transmission can lead to a decrease in prediction accuracy. At this point, it is necessary to use regularization methods that modify the penalty term of cost function and regularization discarding methods that discard neurons to reduce model complexity and prevent overfitting in Equation (4).

$$\begin{cases} L1 = H(p,q) + \alpha \sum_{i=1} |\beta_i| \\ L2 = H(p,q) + \alpha \sum_{i=1} \beta_i^2 \end{cases} \quad (4)$$

In Eq. (4), $L1$ is expressed as regularization that minimizes weight's absolute value. $L2$ is expressed as regularization that minimizes the square of weights. α is expressed as a regularization coefficient. β is expressed as a weight coefficient. $L1$ is suitable for avoiding overfitting by reducing weight density to simplify data while accurately modeling. $L2$ is suitable for avoiding overfitting in computer image recognition by attenuating weights. So combining the two can achieve good regularization results. When RNN is applied to NIDS to detect real data, it can effectively improve the accuracy of analyzing common attack data.

B. Design and Implementation of Domain Name Classification Network Model

The model needs to introduce DGA and CNN to address rare attacks such as malicious domain names [20, 21]. Compared to traditional network models, CNN can further simulate the multi-layer NN architecture of brain to analyze complex information. It can improve data model's ability to analyze and judge complex feature relationships between network packets. Fig. 4 show its network structure.

According to Fig. 4, CNN contains a 5-layer structure. Among them, input layer can standardize multidimensional data to improve model's learning efficiency and result expression. Convolutional layer mainly uses excitation functions to assist in feature recognition and extraction of standardized input data. Pooling layer uses pooling functions to select the characteristics of the extracted feature map and screen key information, so as to reduce parameters number and realize feature data invariance to reduce subsequent calculation. Fully connected layer uses classification algorithms to perform nonlinear combination of transformed extracted features to achieve the function of a "classifier". The output layer uses logic or normalization functions to process data and output a numerical matrix. Eq. (5) and (6) represent the calculation of fully connected layers.

$$y_i = relu(w_i * y_{i-1} + b_i) \quad (5)$$

In Eq. (5), y_i is expressed as the output of i -th layer. w_i is expressed as a weight coefficient. b_i is expressed as an offset parameter. $relu$ is expressed as activation function.

$$F(y) = i \big| x; \theta(w,b)) = e^{\theta(w,b)x} \Big/ \sum_{j=1}^{k} e_j^{\theta(w,b)x} \quad (6)$$

In Eq. (6), $F()$ is the classification function. y refers to the predicted project category value. x refers to the sample value. $\theta(w,b)$ refers to the classification parameter. e is represented as a natural base. k refers to the label type of classified data. The preprocessed data enters the output layer for normalization processing and outputs the result, which is mathematically expressed as Eq. (7).

$$T_{i,j} = (T_{i,j} - \min(T_{i,j})) / (\max(T_{i,j}) - \min(T_{i,j})) \quad (7)$$

In Eq. (7), $T_{i,j}$ is expressed as the characteristic values of a certain row and column. This model introduces DGA on the foundation of CNN, and collects samples from all network DSs and classifies them into small DSs according to certain standards for fine processing. Thus, an N-gram combined Character Based Deep Network (NCBDN) on the foundation of DGA was proposed. Fig. 5 shows the model process.

From Fig. 5, the model mainly consists of several parts, including data collection, data feature extraction and classification, abnormal data classification, and response strategy. The data detection model divides the overall network DS into smaller DS according to the network protocol, such as transmission control protocol, User Datagram Protocol and Internet control message protocol [22]. After further dividing each DS into normal and abnormal data, the experiment further classifies the abnormal DS into different attack types to form policy responses. Fig. 6 shows the data detection module and abnormal data learning module.

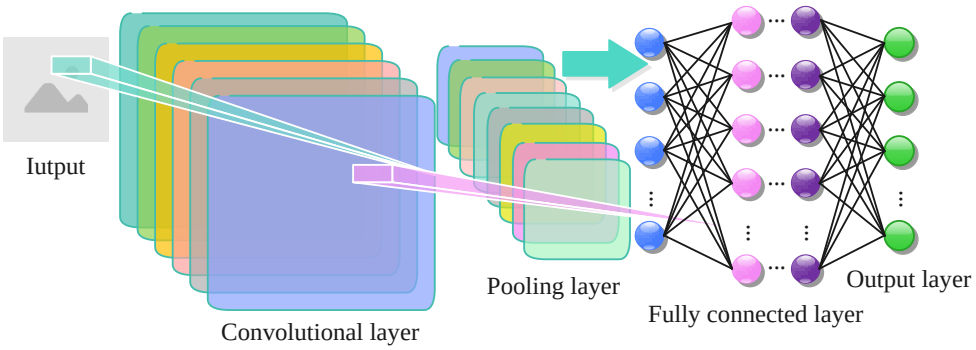

Fig. 4. CNN structure diagram.

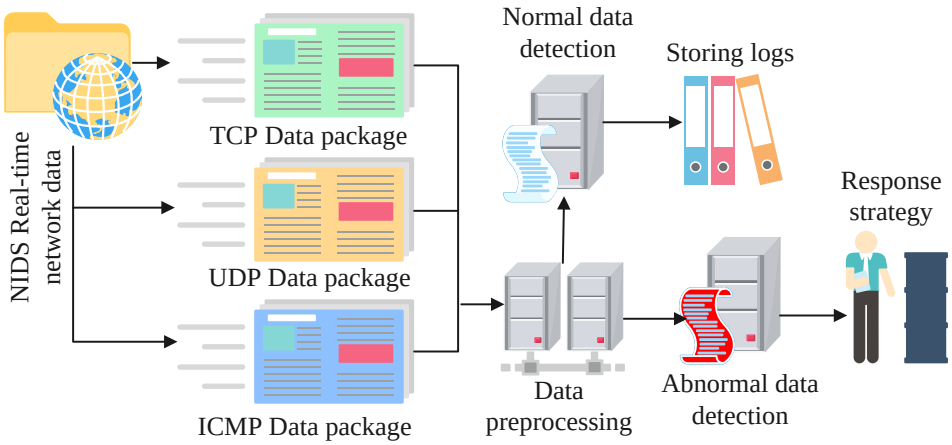

Fig. 5. Flowchart of NCBDN data probe model.

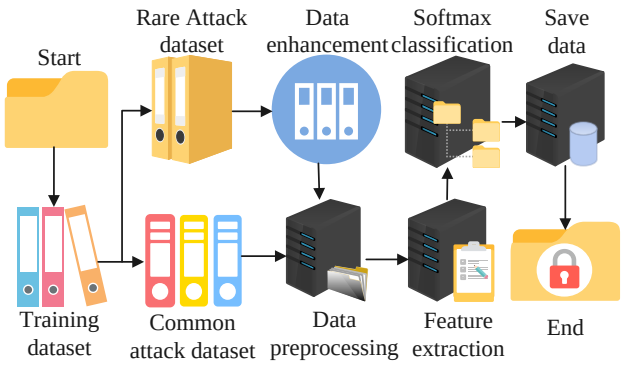

Fig. 6. Flowchart of data detection module and flow chart of abnormal data training module.

From Fig. 6, the model categorizes attack types into ordinary attacks and rare attacks. In order to form the initial training experience of the model, it is necessary to use the training set for initial training to form a basic feature library. To address malicious domain name attacks, it is necessary to first calculate the correlation coefficients between malicious domain names and normal domain names based on the characteristics of the DGA algorithm. Eq. (8) is its mathematical expression.

$$p = \left(\sum_{i=1}^{N}(z_i - \bar{z})(e_i - \bar{e})\right) \Big/ \left(\sqrt{\sum_{i=1}^{N}(z_i - \bar{z})^2}\sqrt{\sum_{i=1}^{N}(e_i - \bar{e})^2}\right) \quad (8)$$

In Eq. (8), p refers to the correlation coefficient. N refers to the total sample points. z_i and e_i respectively represent the spatial distribution values of individual normal and malicious domain names. $\bar{z}$ and $\bar{e}$ both represent the average domain name distribution value. According to the calculated correlation coefficient, it is necessary to perform data transformation on the incoming raw training data. This facilitates a more comprehensive analysis and classification of model in the next step. The preprocessing of data transformation first requires feature extraction of data gain. Eq.(9) is the definition of entropy.

$$Info(S) = -\sum^{\infty} Q_i \log(Q_i) \quad (9)$$

In Eq. (9), *Info* is expressed as an information return function. S is expressed as a variable. Q is expressed as the probability of each variable value. So Equation (10) is the information required for identifying variables.

$$Info(L, S) = \sum_{i=1}^{m} |S_i| / |S| * Info(S_i) \quad (10)$$

In Eq. (10), L is expressed as a non-classified label value, and m is expressed as sample number. Combining Eq. (9) and (10) can ultimately obtain the required data gain in Eq. (11).

$$Gain(L,S) = Info(S) - Info(L,S) \tag{11}$$

In Eq. (11), *Gain* is expressed as a gain function. After the preprocessed DS enters detection module, if the feature code matches the regular data feature code, it will be classified and archived. If the signature matches the abnormal feature, a policy response will be triggered. In addition, the judgment criteria for classifiers include multiple aspects. Eq. (12) refers to accuracy and false positive rate.

$$\begin{cases} Accuracy = (TP+TN)/(TP+TN+FP+FN) \\ FalsePositiveRate = FP/(FP+TN) \end{cases} \tag{12}$$

In Eq. (12), *TP* is true class, *TN* is true negative class, *FP* is false positive class, and *FN* is false negative class. The mathematical expressions for precision ($\Pr ecision$) and recall ($\mathrm{Re}call$) in Eq. (13) are as follows.

$$\begin{cases} \Pr ecision = TP/(TP+FP) \\ \mathrm{Re}call = TP/(TP+FN) \end{cases} \tag{13}$$

Eq. (14) is the mathematical expression for micro precision (*microP*) and micro recall (*microR*).

$$\begin{cases} microP = \overline{TP}/(\overline{TP}+\overline{FP}) \\ microR = \overline{TP}/(\overline{TP}+\overline{FN}) \end{cases} \tag{14}$$

The micro F1 equation in Eq. (15) can be obtained from Equation (14).

$$microF1 = (2\times microP\times microR)/(microP+microR) \tag{15}$$

In Equation (15), *microF*1 refers to average micro harmonic value, and Equation (16) refers to macro F1.

$$macroF1 = ((2/n^2)\sum_{1}^{n} P_i R_i) \Big/ ((1/n)\sum_{1}^{n} P_i + (1/n)\sum_{1}^{n} R_i) \tag{16}$$

In Eq. (16), *macroF*1 is average macro harmonic value, P is precision, and R is recall. The intelligent judgment accuracy can be comprehensively evaluated through classifier's judgment criteria. This facilitates timely adjustment and optimization of system parameters to ensure stable model operation within the optimal range for a long time.

IV. Model Validation and Data Analysis

Based on the above algorithm analysis, NCBDN has higher precision in detecting, analyzing and classifying rare attacks, compared with the traditional detection model including malicious domain names based on DGA algorithm. To verify model performance advantage in detecting and identifying malicious domain names generated by DGA, NCBDN was used in this experiment to compare data detection classification of 16 domain names generated by DGA. Fig. 7 shows the comparison of four testing standards for binary (B) character detection.

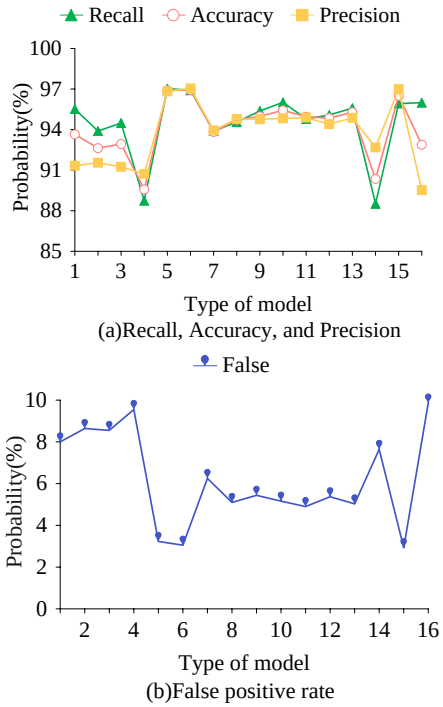

Fig. 7. Standard probability plot for bigram character detection.

From Fig. 7, NCBDN based on binary syntax has an average accuracy of about 94%, an average detection rate of about 94%, an average precision of about 93%, and an average error rate of about 6% for the recognition of 16 malicious domain names generated by DGA algorithm. Fig. 8 shows the comparison of trigram (T) character detection indicators.

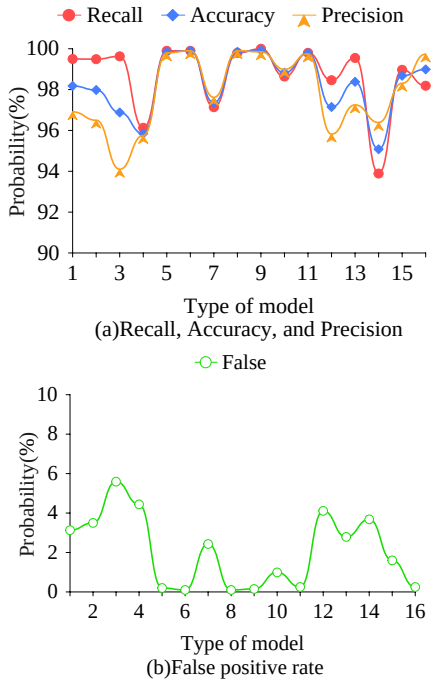

Fig. 8. Standard probability plot for trigram character detection.

From Fig. 8, NCBDN recognition rate with ternary syntax for malicious domain names is further improved by about 98%, detection rate by about 98%, precision by about 98%, and error rate by about 2%. So when N=3, NCBDN has the highest efficiency in identifying malicious domain names. To further validate model advantages in identifying malicious domain names, NCBDN binary and ternary models were compared with six models for malicious domain name recognition. They include Logistic Regression (LR), Naive Bayesian Model (NB), K-Nearest Neighbor (KN), and Support Vector Machine (SV) in Fig. 9.

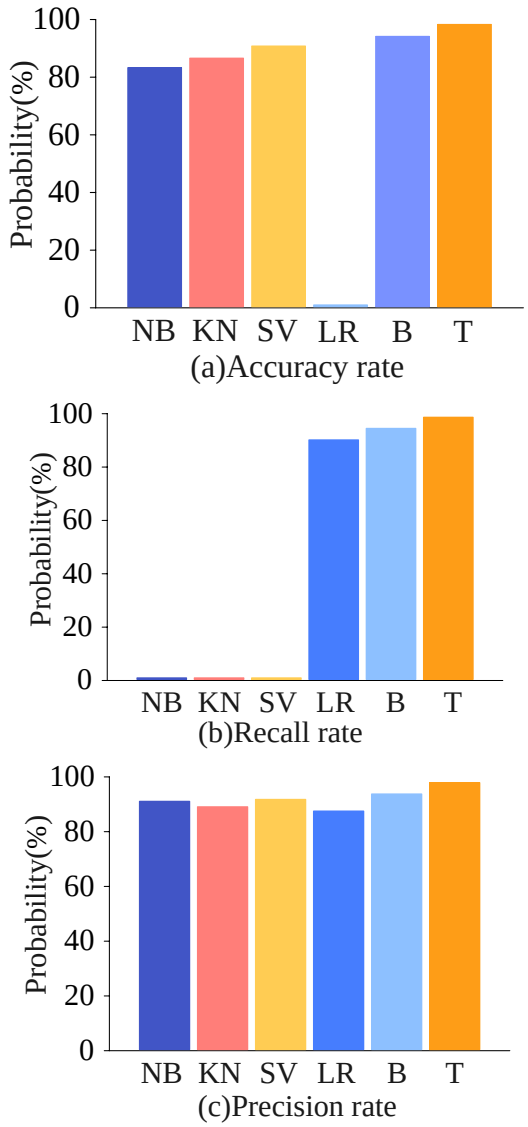

Fig. 9. Comparison of malicious domain name recognition probabilities of 17 models.

In Fig. 9, only NCBDN model under ternary syntax has an average DR of over 98%. This model can more effectively identify malicious domain names generated by DGA. This model adopts feature extraction methods to enhance the recognition probability of this model against unknown attacks. So seven methods were used, including Empirical Mode Decomposition (EMD), Hierarchical Spatial temporal features based ID System (HAST-IDS), Text-CNN and RF based ID System (TR-IDS), Analog Network ID System (A-NIDS), Recursive Feature Addition(RFA), SVM and Intuitionistic Fuzzy Set for Anomaly Detection (IFSE-AD). The training datasets are Information Security Center of Excellence 2012 (ISCX2012) and Canadian Institute for Cybersecurity ID System 2017 (CICDS2017).

The ISCX2012 dataset was collected by the Network Security Laboratory of Dalhousie University in Canada to provide a real network traffic dataset for research in network intrusion detection and traffic analysis. The ISCX2012 dataset consists of two sub-datasets. The Botnet-based sub-dataset contains normal traffic from the real network and malicious traffic from Zeus and ZeroAccess, among which normal traffic accounts for about 85% of the whole dataset and malicious traffic accounts for about 15%. The DDOS-based sub-dataset contains normal traffic from the real network and malicious traffic from different types of DDoS attacks, where normal traffic accounts for about 80% of the whole dataset and malicious traffic accounts for about 20%.

The CICIDS2017 dataset was collected in collaboration with the Institute for National Security and Counterterrorism (INSA) laboratory and the Cybersecurity Laboratory at Concordia University. This paper aims to provide a diverse network intrusion detection dataset for evaluating and comparing different intrusion detection systems. The CICIDS2017 dataset consists of several sub-datasets, the most commonly used of which are: The Botnet-based sub-dataset contained normal traffic from the real network and malicious traffic from malicious botnets such as Mirai, Gafgyt and TBot, among which normal traffic accounted for about 60% of the whole dataset and malicious traffic accounted for about 40%. The DDOS-based sub-dataset contains normal traffic from the real network and malicious traffic from different types of DDoS attacks, where normal traffic accounts for about 80% of the whole dataset and malicious traffic accounts for about 20%.

It is important to note that the components and proportions of the dataset may vary from version to version and use.

Table I shows the data results and the recognition efficiency of the model for unknown attacks.

In Table I, N/A indicates that the return value is invalid. From Table I, as sample number increases, model recognition accuracy increases. The recognition accuracy of EMD is 90% when data volume reaches around 10000. HAST-IDS has a recognition accuracy of over 90% with a data volume of around 900000. The accuracy of TR-IDS exceeds 90% when data volume reaches 30000. The accuracy of A-NIDS is close to 90% when data volume reaches 80000. The recognition rate of RFA is about 70% when data volume is 30. The recognition probability of IFSE-AD is over 95% when data volume reaches 20000. The accuracy of NCBDN can reach over 95% when data volume is 20. So this model has certain advantages in learning efficiency. The model has a high recognition probability for ordinary attacks, but the recognition probability for rare attacks is unknown. Fig. 10 shows the identification probability for verifying against rare attacks.

TABLE I. LEARNING AND RECOGNITION EFFICIENCY OF DIFFERENT PROBE MODELS

Method	Data Set	Sample Size	A (%)	R (%)
EMD	ISCX2012	9548	91.20	91.78
HAST-IDS	ISCX2012	819167	89.46	85.69
HAST-IDS	ISCX2012	908734	98.92	95.10
TR-IDS	ISCX2012	31407	91.45	91.76
A-NIDS	ISCX2012	80645	88.10	N/A
RFA	ISCX2012	24	76.50	70.40
RFA	ISCX2012	488	91.54	88.47
SVM	CICIDS2017	N/A	N/A	93.89
IFSE-AD	CICIDS2017	2422	92.55	N/A
IFSE-AD	CICIDS2017	23194	96.79	N/A
NCBDN	ISCX2012	8	96.26	97.99
NCBDN	ISCX2012	16	98.73	98.91
NCBDN	CICIDS2017	8	93.58	99.24
NCBDN	CICIDS2017	16	96.91	99.55

In Fig. 10, five types of data are introduced, namely Denial of Service (Dos), User to Root (U2R), Remote to Local (R2L), Probe, and Normal. U2R and R2L are rare attacks. As meta grammar increases from primeval number to 4, their recognition probability against rare attacks increases first and then decreases. When N=3, its recognition probability is the highest, about 85% for U2R and 92% for R2L. To further confirm model recognition performance when N=3, the detection accuracy was compared with Hierarchical ID model (HIDM), Managed ID model (MIDM). Fully connected detection model (FCDM), and CNN model (CNNM) in Fig. 11, respectively.

In Fig. 11, NCBDN has significant advantages over traditional models in terms of learning efficiency and detection efficiency of rare attacks. NCBDN has a high learning efficiency for unknown attacks and a high probability of identifying rare attack data.

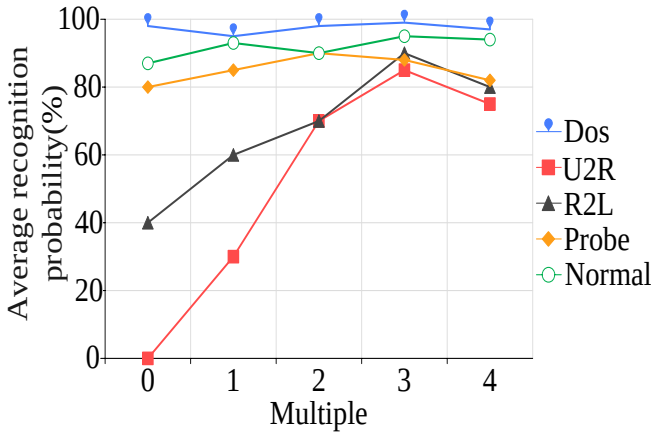

Fig. 10. Comparison plot of data augmentation probabilities.

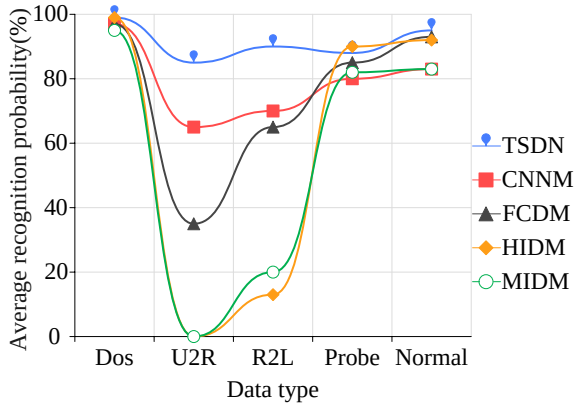

Fig. 11. Comparison of common detection models.

V. DISCUSSION

This study aims to explore the network intrusion detection method based on deep learning technology. With the rapid development of computer networks, cyberspace carries more and more high-value information, and the problem of network security is becoming more and more serious. The traditional intrusion detection system has the problems of slow detection speed and high false alarm rate, so it is necessary to use deep learning technology to improve the detection effect.

Several deep learning-based network intrusion detection methods have been proposed in the research. Firstly, by combining recurrent neural network with Gated Recurrent Unit (GRU) and Multilayer Perceptron (MLP), a network intrusion detection method based on GRU was proposed. Experimental results show that the proposed method has higher detection rate and lower false alarm rate. Secondly, a domain name detection method based on semantic expression is proposed. The detection of malicious domain names is realized by using deep convolutional neural networks. Experimental results show that the method has a good detection effect on domain names generated by different types of algorithms.

In addition, a network intrusion detection method for small sample based on meta-learning framework is proposed. This method realizes the detection of network intrusion behaviors in small sample scenarios through differential expression. Experimental results show that the proposed method has higher detection rate in small sample scenarios.

Finally, a low-rate denial of service attack detection method based on hybrid deep neural network was proposed. This method realizes the detection of low-rate DOS attacks by one-dimensional convolutional neural network and gated recurrent unit. Experimental results show that the proposed method has good detection effect in both large sample and small sample scenes.

In summary, the research has proposed a variety of effective network intrusion detection methods by applying deep learning techniques. Experimental results show that these methods can achieve good detection results in different scenes. Future work can further study the application of deep learning technology in the field of network security, solve the security problem of deep neural network itself, and further improve the

intrusion detection method according to the characteristics of the development of emerging networks.

VI. Conclusion

In response to issues such as low learning and recognition efficiency in malicious domain name attacks, this study first based on traditional NIDS models and introduced a recurrent neural deep learning network model containing GRU to make it more intelligent. To enable it to identify unknown rare attacks more quickly and accurately, this research further deepened and constructed NCBDN based on DGA algorithm. The experiment trained and learned the model using DGA algorithm and a dataset containing attack DS such as Dos, U2R, R2L, and Probe, as well as normal DS. The optimal DGA, Dos, 99%, U2R, 85%, R2L, Probe, 81%, and Normal recognition probabilities for NCBDN for malicious domain names and five types of data were 98%, 97%, and 97%, respectively. The traditional ID system has a low comprehensive average recognition rate for DGA and two rare attacks, with HIDM being 0% and 16%, and MIDM being 0% and 20%, respectively. The average recognition rate of general NN is relatively high, with FCDM being 35% and 65%, and RNNM being 68% and 71%, respectively. The highest recognition rates for NCBDN are 98% and 92%, respectively. NCBDN model is obtained by improving N-gram of the traditional network based on DGA. This model not only ensures high recognition probability for normal data and ordinary attack data. And it further improves learning efficiency for unknown attacks and the recognition probability for rare attack data. However, NCBDN has low efficiency in collecting threat data in large network traffic. Therefore, model detection efficiency for threats in big data needs to be further improved.

References

[1] K. Tsiknas, D. Taketzis, K. Demertzis, and C. Skianis, “Cyber threats to industrial IoT: A survey on attacks and countermeasures”, IoT, vol. 2, pp. 163-186, January 2021.

[2] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, “A survey on space-air-ground-sea integrated network security in 6G”, IEEE Commun. Surv. Tutorials, vol. 24, pp. 53-87, November 2021.

[3] M. H. U. Sharif and M. A. Mohammed, “A literature review of financial losses statistics for cyber security and future trend”, World J. Adv. Res. Rev., vol. 15, pp. 138-156, August 2022.

[4] W. Duo, M. C. Zhou, and A. Abusorrah, “A survey of cyber attacks on cyber physical systems: Recent advances and challenges”, IEEE/CAA J. Automatica Sin., vol. 9, pp. 784-800, April 2022.

[5] P. Kumar, G. P. Gupta, and R. Tripathi, “Design of anomaly-based intrusion detection system using fog computing for IoT network”, Automatic Contr. Comput. Sci., vol. 55, pp. 137-147, May 2021.

[6] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, I. Kevin, and, K. Wang, “Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system”, IEEE Internet Things J., vol. 9, pp. 9310-9319, November 2021.

[7] P. Nimbalkar and D. Kshirsagar, “Feature selection for intrusion detection system in Internet-of-Things (IoT)”, ICT Express, vol. 7, pp. 177-181, June 2021.

[8] A. Moubayed, L. Yang, and A. Shami, “MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles”, IEEE Internet Things J., vol. 9, pp. 616-632, May 2021.

[9] Y. Guo, Z. Mustafaoglu, and D. Koundal, “Spam detection using bidirectional transformers and machine learning classifier algorithms”, J. Comput. Cogn. Eng., vol. 2, pp. 5-9, April 2023.

[10] I. Hidayat, M. Z. Ali, and A. Arshad, “Machine learning-based intrusion detection system: An experimental comparison”, J. Comput. Cogn. Eng., vol. 2, pp. 88-97, July 2022.

[11] A. Binbusayyis and T. Vaiyapuri, “Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM”, Appl. Intell., vol. 51, pp. 7094-7108, February 2021.

[12] N. Wang, Y. Chen, Y. Xiao, Y. Hu, W. Lou, and Y. T. Hou, “Manda: On adversarial example detection for network intrusion detection system”, IEEE Trans. Dependable Secur. Comput., vol. 20, pp. 1139-1153, February 2022.

[13] S. Krishnaveni, S. Sivamohan, S. S. Sridhar, and S. Prabakaran, “Efficient feature selection and classification through ensemble method for network intrusion detection on cloud computing”, Cluster Comput., vol. 24, pp. 1761-1779, January 2021.

[14] A. H. Azizan, S. A. Mostafa, A. Mustapha, C. F. M. Foozy, M. H. A. Wahab, M. A. Mohammed, and B. A. Khalaf, “A machine learning approach for improving the performance of network intrusion detection systems”, Ann. Emerg. Technol. Comput. (AETiC), vol. 5, pp. 201-208, March 2021.

[15] O. Almomani, “A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system”, Comput., Mater. Contin., vol. 68, pp. 409-429, March 2021.

[16] M. Rashid, J. Kamruzzaman, T. Imam, S. Wibowo, and S. Gordon, “A tree-based stacking ensemble technique with feature selection for network intrusion detection”, Appl. Intell., vol. 52, pp. 9768-9781, January 2022.

[17] M. U. Ilyas and S. A. Alharbi, “Machine learning approaches to network intrusion detection for contemporary internet traffic”, Comput., vol. 104, pp. 1061-1076, January 2022.

[18] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, “Explaining deep neural networks and beyond: A review of methods and applications”, Proc. IEEE, vol. 109, pp. 247-278, March 2021.

[19] W. Zhang, H. Li, L. Tang, X. Gu, L. Wang, and L. Wang, “Displacement prediction of Jiuxianping landslide using gated recurrent unit (GRU) networks”, Acta Geotech., vol. 17, pp. 1367-1382, April 2022.

[20] M. Tripathi, “Analysis of convolutional neural network based image classification techniques”, J. Innov. Image Process. (JIIP), vol. 3, pp. 100-117, June 2021.

[21] V. Ravi, M. Alazab, S. Srinivasan, A. Arunachalam, and K. P. Soman, “Adversarial defense: DGA-based botnets and DNS homographs detection through integrated deep learning”, IEEE Trans. Eng. Manage., vol. 70, pp. 249-266, March 2021.

[22] M. Aboubakar, M. Kellil, and P. Roux, “A review of IoT network management: Current status and perspectives”, J. King Saud University-Computer Inform. Sci., vol. 34, pp. 4163-4176, July 2022.