Malicious Traffic Detection Algorithm for the Internet of Things Based on Temporal Spatial Feature Fusion

Linzhong Zhang
School of Electronic Information Engineering, Tianjin Vocational Institute, Tianjin, 300410, China

***Abstract*—With the rapid development of the Internet of Things, the security issues of its network environment have gradually attracted attention. To enable faster and more accurate identification and detection of malicious traffic attacks in the Internet of Things, an optimized malicious traffic detection algorithm based on fusion of temporal and spatial features is proposed. This method improves the feature extraction performance of traffic data and increases the accuracy of traffic detection. The test results showed that the comprehensive performance of the fusion algorithm was superior to the other four algorithms used for comparison. On the KDD99-CUP dataset, the F1 of the feature fusion algorithm reached 93.16%, while the F1 of algorithms 1-4 were 81.36%, 67.89%, 90.56%, and 92.24%, respectively. On the test set, 182 traffic samples were accurately identified, including 139 correctly identified malicious traffic and 43 correctly identified normal traffic, with recognition accuracy of 98.73% and 97.65%, respectively. Experimental results revealed that the use of fused feature extraction in traffic detection systems could improve detection efficiency and accuracy, providing a safer and more reliable guarantee for the interaction process of the Internet of Things network, and safeguarding the rapid development and application of the Internet of Things.**

***Keywords*—Internet of Things; network security; temporal-spatial characteristics; traffic detection; fusion algorithm**

I. INTRODUCTION

With the rapid development of the Internet, the Internet of Things (IoT) has also been widely used in a series of intelligent building systems such as smart cities, smart offices and smart homes [1-2]. However, because some IoT devices are directly exposed to the Internet, they face more security problems than other network interaction methods [3]. IoT based network attacks may affect communication quality, and even cause signal loss, network paralysis, and other phenomena, seriously threatening users' privacy and security [4]. For traditional anomaly traffic detection techniques, due to the concealment and complexity of existing IoT network attacks, and the fact that traffic attacks exhibit different characteristics with different network environments, there are problems with imbalanced traffic datasets and difficult feature extraction [5]. These issues have increased the difficulty of designing malicious traffic identification algorithms. In current traffic detection systems, the main focus is on identifying and analyzing a single type of traffic to achieve abnormal alerts. However, due to the different types and characteristics of attacks, and some attacks being of unknown types, the accuracy of traffic detection systems is not ideal, making it difficult to predict unknown attacks. To solve the above problems, a traffic detection algorithm based on temporal feature fusion is proposed, and malicious traffic analysis and detection in the IoT are completed on this basis.

II. RELATED WORKS

In today's era of rapid development of information technology, network traffic analysis has become an important means to ensure network security, optimize network performance, and improve user experience. Furthermore, with the explosion of Internet users and the popularization of various network applications, the complexity and diversity of network traffic are also increasing. Therefore, in-depth analysis of the characteristics and detection methods of malicious traffic is of great importance for network management and security protection. Yang H et al. proposed a fast strategy hill-climbing learning method to optimize the power allocation for malicious traffic detection in intelligent malicious traffic controllers. Therefore, the malicious traffic detection system could quickly achieve the optimal strategy when the malicious traffic model was unknown. The test results showed that the transmission rate of this malicious traffic detection method increased by 12.28% compared to the original, which made the malicious traffic detection system perform better [6]. Salem A et al. proposed an adaptive adaptation scheme that could make the detection of malicious traffic techniques constructive for legitimate users and destructive for eavesdroppers. Based on the average symbol error probability in different scenarios, this method used a finite rate to represent the overall retention rate. Malicious traffic detection technology could achieve an additional gain of 17dB in transmission signal-to-noise ratio and a gain of 10dB in total secrecy rate [7]. Su N et al. used joint design to transmit and receive beamforming and destructive malicious traffic techniques to weaken the eavesdropping signals of eavesdropping radar in wireless networks. The experimental results showed that this technology performed better than dual function radar network technology in terms of secure transmission [8]. Many radars and networks work in a coordinated manner, but Du Z et al. proposed using a non coordinated approach to study the impact of malicious network traffic on radar target detection. The study solved the maximum likelihood estimation in homogeneous clutter to optimize detection performance. Through verification, it was found that the improved target detection system showed significant improvement in performance in network environments with high levels of malicious traffic [9]. Hosseinali J et al. proposed an improved adaptive algorithm to address the issue of malicious attacks on high-power ranging devices. The algorithm optimized power allocation for each reliable subcarrier, while subcarriers subjected to highly malicious traffic were deactivated. This adaptive technology could reduce the bit error rate to 10-7 under malicious traffic in high-power ranging devices, improving the reliability of network systems [10]. In the scenario of multi-user and multi-transceiver simultaneous network, to eliminate malicious traffic on reconfigurable intelligent surfaces, the Jiang T team proposed an alternating projection algorithm, which took the solution obtained by the algorithm approaching 0 as the initial value for subsequent optimization, and changed the phase of the reconfigurable surface components accordingly. Tests showed that the improved algorithm could detect malicious traffic on reconfigurable surfaces within the experimental range [11].

The malicious traffic detection performance of network systems is largely influenced by algorithms. Some scholars have conducted more in-depth research and experiments on the optimization of deep learning algorithms. Lin J et al. proposed an improved genetic algorithm based on four quadrant photodetectors to improve the accuracy and stability of visible light positioning. This algorithm enabled the detector to locate the measurement point based on the received illuminance value. The average positioning error of the positioning system using optimization algorithms was 4.023cm [12]. Mahmoud B et al. proposed a hybrid model of deep learning algorithm and tabu search to achieve balanced coverage of all targets in sensor networks. The optimized algorithm required the use of multiple sensors for coverage. Several experiments showed that this fusion algorithm outperformed algorithms based on automatic learning [13]. To extend the lifespan of wireless sensor networks, Rajan L et al. proposed a new optimization algorithm based on the grey wolf algorithm using Na deep learning algorithm, which selected the optimal cluster head under constraints such as separation distance and energy consumption. Compared to the classical grey wolf algorithm and particle swarm optimization algorithm, the improved grey wolf algorithm improved overall performance by 28.6% and 31.5%, respectively [14]. Shaikh M et al. proposed using optimized deep learning algorithms to achieve higher accuracy in the calculation of parameters for overhead transmission lines. This optimization algorithm was applied to single-phase and three-phase transmission lines, achieving optimal solutions for the vast majority of benchmark functions. The accuracy and computational efficiency of the optimized deep learning algorithm had been improved [15].

In conclusion, despite the numerous inferences and experiments conducted by scholars on optimizing detection algorithms and enhancing traffic detection performance, the intricate neural network structure inherent to deep learning algorithms results in a slow convergence speed and a notable decline in detection accuracy. To further enhance the performance of malicious traffic detection, an optimized malicious traffic detection algorithm based on the fusion of temporal and spatial features is proposed. This is achieved through the use of a mixed sampling and variational autoencoder data augmentation algorithm, which enables more intelligent and efficient detection results in complex environments.

III. Methods and Materials

A. *Flow Detection Algorithm Integrating Temporal and Spatial Features*

When faced with malicious traffic attacks, intrusion detection systems can use detection algorithms that continuously learn traffic information characteristics to identify them. When attacked, the detection system can determine the traffic attack behavior [16]. The process of traffic detection mainly consists of network status detection and traffic detection. The former is to detect the operating status of network and host and monitor the fluctuation of network system in real time, while the latter is to extract the characteristics of traffic data and complete the detection and analysis. The characteristics of traffic data can be analyzed from both temporal and spatial dimensions. For the temporal dimension, there is correlation between historical traffic data, while for the spatial dimension, there is local spatial correlation between traffic characteristics [17]. By integrating these two different types of features, traffic detection can simultaneously capture feature information from different dimensions, identify different traffic activities, and make the results of detection algorithms more accurate and reliable. The architecture of the data traffic detection system is depicted in Fig. 1.

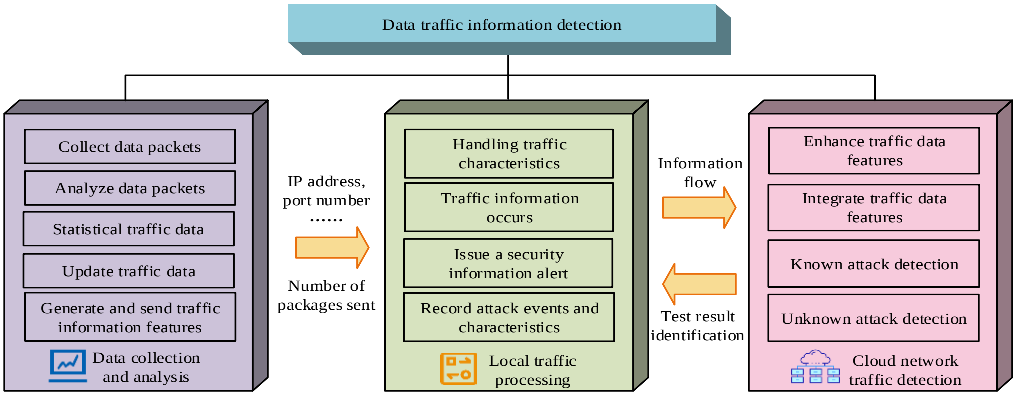

Data traffic information detection
Collect data packets
Analyze data packets
Statistical traffic data
Update traffic data
Generate and send traffic information features
Data collection and analysis
IP address, port number ……
Number of packages sent
Handling traffic characteristics
Traffic information occurs
Issue a security information alert
Record attack events and characteristics
Local traffic processing
Information flow
Test result identification
Enhance traffic data features
Integrate traffic data features
Known attack detection
Unknown attack detection
Cloud network traffic detection

Fig. 1. The architecture of data traffic detection system.

In Fig. 1, the system mainly includes a data collection and analysis module, a local traffic processing module, and a cloud network traffic detection module. A traffic detection method based on temporal and spatial feature fusion has been proposed, and the main detection process is shown in Fig. 2.

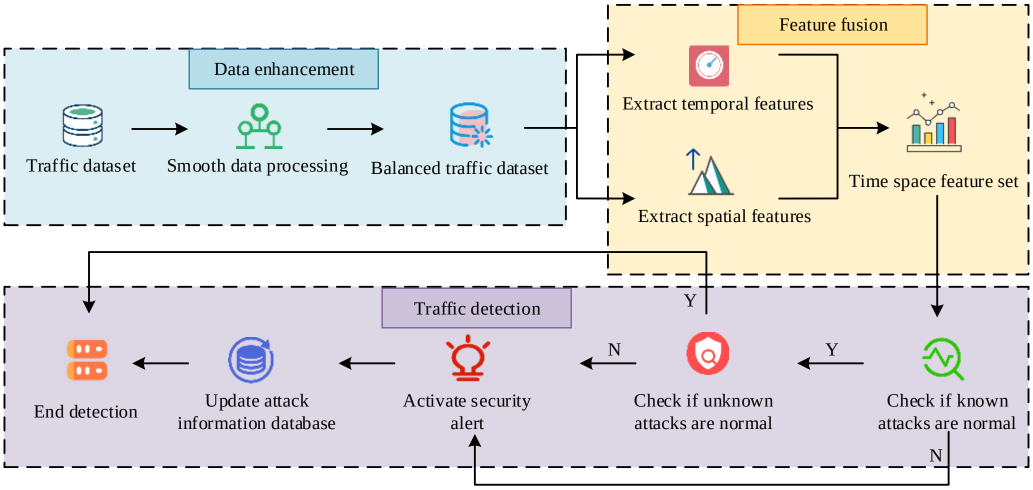


Fig. 2. Main process of detection.

Probability can be mainly divided into parametric estimation and nonparametric estimation, mainly used to estimate the potential probability density function of target information that is invisible [18]. The condition for parameter estimation is that the target information needs to follow a probability distribution and its parameters are unknown, so the corresponding parameters need to be solved through known data. Non parameter estimation requires the target information to have a probability density function, which is solved through observation data. Although it is mainly data-driven and does not require a probability distribution, its computational complexity is relatively high. If the one-dimensional random variable is the probability density function x of $f(x)$ can be represented by Eq. (1).

$$f(x)=\lim_{h\to 0}\frac{F(x+h)-F(x-h)}{2h} \tag{1}$$

In Eq. (1), x represents a one-dimensional random variable, h is the bandwidth width parameter, $F(x)$ is the probability distribution function, and the definition of $F(x)$ can be seen in Eq. (2).

$$F(x)=P(X\le x) \tag{2}$$

Frequency estimation probability can be applied to the dataset. If a one-dimensional random variable has n samples $\{x_1,x_2,x_3,....,x_n\}$, its probability distribution function $\hat{F}(x)$ can also be expressed as Eq. (3).

$$\hat{F}(x)=\frac{k}{n} \tag{3}$$

In Eq. (3), k represents the number of samples smaller than x, and the number of samples is set as m, thus obtaining the estimated equation for the probability density function $\hat{f}(x)$ as shown in Eq. (4).

$$\hat{f}(x)=\frac{m}{2nh} \tag{4}$$

If a uniformly distributed function is defined as $K(x)$, the estimation of the probability density function can also be called a kernel function, which can be represented by Eq. (5).

$$\hat{f}(x)=\frac{1}{nh}\sum_{i=1}^{n}K(\frac{x-x_i}{h}) \tag{5}$$

Multidimensional kernel density estimation can be obtained through univariate kernel density estimation. If a p-dimensional continuous random variable x is set, its multidimensional kernel density estimation equation can be represented by Eq. (6).

$$\hat{f}(x)=\frac{1}{n|H|^{1/2}\ 0}\sum_{i=1}^{n}K(H^{-1/2}(x-xi)) \tag{6}$$

Among them, K is a multidimensional kernel function, and H is a symmetric bandwidth matrix. Analysis Eq. (6) shows that the influencing factors of kernel density estimation are mainly determined by the selection of kernel function K and the size of bandwidth width h. When the cloud receives data information, it needs to first enhance the data before completing data feature extraction and fusion. The process flow of the data collection and analysis module can be seen in Fig. 3.

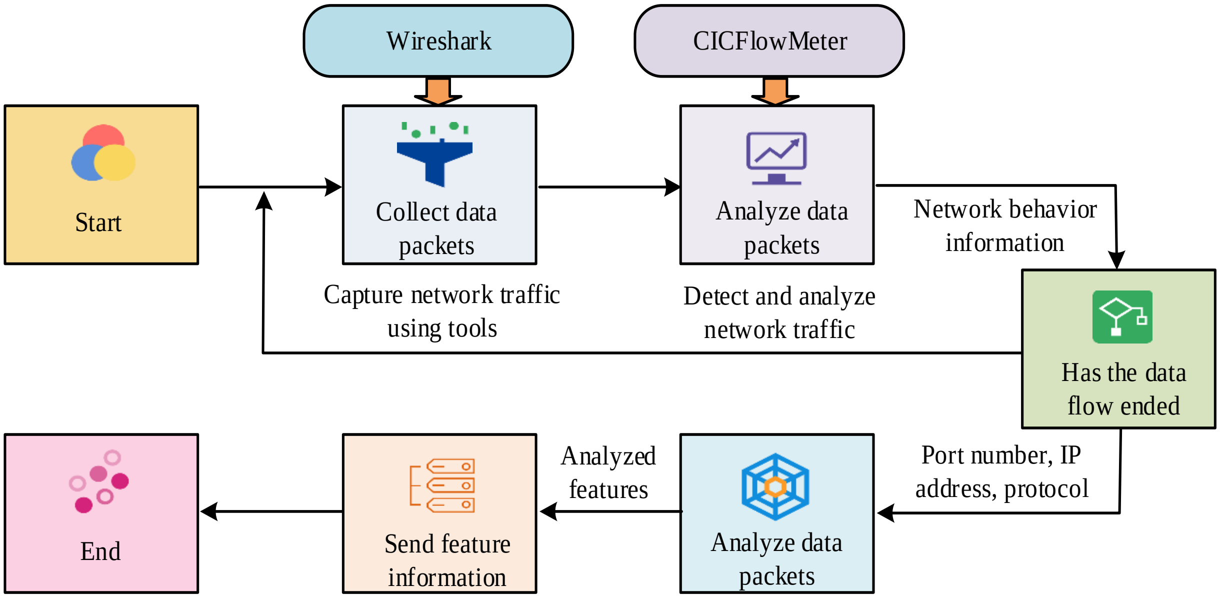


Fig. 3. Data collection and analysis module process.

The extracted and fused feature information will be used for known attack detection. If the detection result is abnormal traffic, further determination of the specific type of abnormal attack is required. If it is detected as normal traffic, further unknown attack detection will be carried out to prevent the IoT from being attacked by unknown traffic. The encoder is mainly composed of an input layer and a fully connected layer. Set the n dimensional feature vector received by the system as x, compress x to obtain a low dimensional latent space z of m dimension, set the neural network parameters of the encoder as f, and the compression function as $h()$, then the encoder can be expressed as Eq. (7).

$$z = h\left(x,\varphi\right) \tag{7}$$

The decoder consists of a fully connected layer and an output layer. Nonlinear computation can be used to reconstruct the low dimensional representation of m dimensional latent space as z into n as the eigenvector x. If the $\hat{x}$ neural network parameters of the decoder are set to θ and the reconstruction function is set to $g()$, the decoder can be expressed as Eq. (8).

$$\hat{x} = g(z,\theta) \tag{8}$$

The autoencoder compresses the input by using the encoder to obtain a low dimensional latent space, which preserves the effective features of the input data and enables the decoder to complete the reconstruction process of the original input data. The optimization objective can be represented by Eq. (9).

$$\varphi,\theta = \arg\min_{\varphi,\theta} \sum_{i=1}^{n} L(x_i,\hat{x}_i) \tag{9}$$

In Eq. (9), x_i is the original feature vector input to the autoencoder, and $\hat{x}_i$ is the reconstructed feature vector output by the autoencoder. The function for measuring vector differences is set as $L()$, which can generally be measured using the mean square loss function. The loss function can be expressed as Eq. (10).

$$s = \sum_{i=1}^{n} L(x_i,\hat{x}_i) \tag{10}$$

The compression network is represented by z, and the reconstruction loss of the input layer u and output layer v can be used as a low dimensional representation of the input features, as expressed in Eq. (11).

$$\begin{cases} u = \left[educlidean\_loss, \cos ine\_loss\right] \\ v = \left[z,u\right] \end{cases} \tag{11}$$

When extracting traffic features, the extraction process can be completed through time and space. Temporal feature extraction mainly targets multiple traffic data, while spatial feature extraction targets individual traffic data, and there are significant differences between these two extraction methods [19]. A feature fusion encoder is proposed by studying the fusion extraction method of temporal and spatial features. It is mainly divided into bidirectional attention temporal encoder and asymmetric multi-scale spatial encoder. The former is mainly responsible for extracting temporal features from spatial features, while the latter is responsible for extracting spatial features from raw traffic data. Through two different extraction methods, different dimensional fusion effects are achieved. The structure of the feature fusion encoder can be seen in Fig. 4.

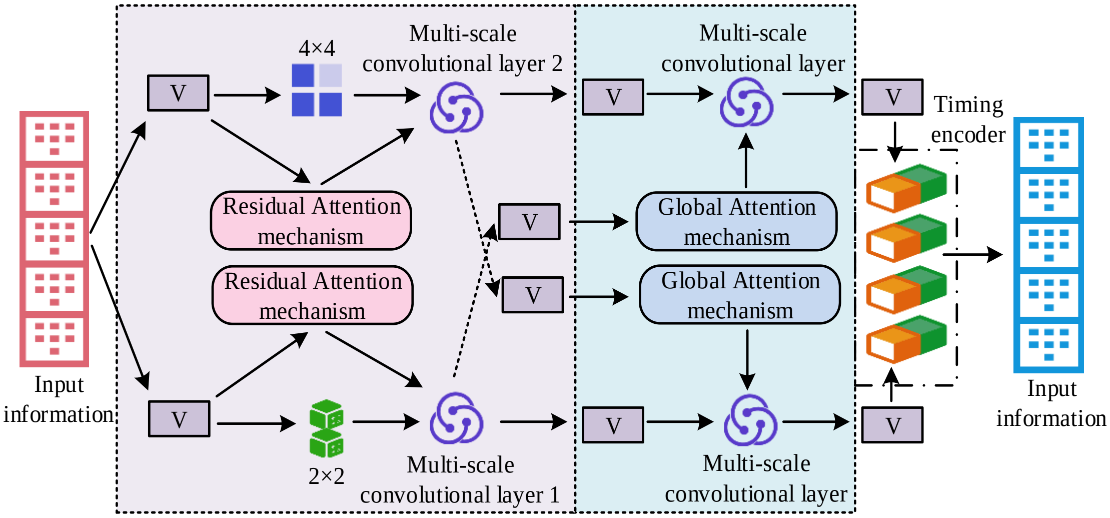

Input information
V
V
4×4
2×2
Multi-scale convolutional layer 2
Multi-scale convolutional layer 1
Residual Attention mechanism
Residual Attention mechanism
V
V
V
V
Multi-scale convolutional layer
Multi-scale convolutional layer
Global Attention mechanism
Global Attention mechanism
V
V
Timing encoder
Input information

Fig. 4. Structure of fusion encoder.

B. Analysis and Detection of Malicious Traffic in the IoT

For malicious traffic anomaly detection, the balance between the number of normal and abnormal data in the dataset cannot be achieved, so its performance cannot be measured solely by accuracy. Accuracy and recall are two important measurement indicators [20]. If the accuracy is set to Acc, its calculation can be represented by Eq. (12).

$$Acc = \frac{TP+TN}{FN+TN+FP+TP} \tag{12}$$

In Eq. (7), TP represents the number of normal traffic samples detected by the detection model as normal, and TN represents the number of abnormal traffic samples detected by the model as abnormal. FN indicates the total number of samples is represented by the number of normal abnormal traffic samples detected by the detection model, and FP represents the number of normal samples detected as abnormal. The sum of these four types of samples is the total number of samples. If the accuracy of the detection is $\Pr e$, the calculation equation is shown in Eq. (13).

$$\Pr e = \frac{TP}{TP+FP} \tag{13}$$

The recall rate represents the proportion of correct predictions in the normal records of the predictive model. If $\operatorname{Re} c$ is used to represent the recall rate, its expression is shown in Eq. (14).

$$\operatorname{Re} c = \frac{TP}{FN+TP} \tag{14}$$

The relationship between accuracy and recall is quite conflicting. If the accuracy is increased by raising the threshold during detection, the recall rate will decrease. Therefore, when evaluating the performance of detection models, it is necessary to comprehensively evaluate the accuracy and recall rates for a more accurate assessment. If F represents the weighted harmonic of precision and recall, its expression can be seen in Eq. (15).

$$F = \frac{(1+\varepsilon)(\Pr e * \operatorname{Re} c)}{(\alpha^2 * \Pr e) + \operatorname{Re} c} \tag{15}$$

In Eq. (15), in general, the value of α is 1. After determining the evaluation criteria for detection performance, the interaction design between the traffic detection system and external modules is first completed. When the traffic detection system is attacked, the corresponding attack behavior can be detected, and the message can be transmitted to other functional modules through alarm information [21]. The architecture of Dbus is client service, which is a simple and fast communication method that supports point-to-point communication and can send messages to specific processes in a directed manner. Research has chosen Dbus as the mechanism for message notification to facilitate information exchange between the detection system and the external environment. The relationship between the system and external modules can be seen in Fig. 5.

After receiving relevant information, the local processing module completes the processing of information data, alarms and related records. When data information is received from the data acquisition module and the data analysis module, appropriate numerical and normalization processing of the feature data is required. The preprocessed information is sent to the cloud processing module, and after analysis and recognition by the cloud processing module, it is sent back to the local processing module [22]. The local processing module completes the parsing and judgment. If it is judged as normal traffic, it waits for the next traffic information from the data collection and analysis module. If it is judged as malicious attack traffic, a security alarm will be triggered and the traffic event will be sent to the corresponding module for recording. The process flow of the local processing module can be seen in Fig. 6.

The cloud network traffic detection module is mainly responsible for analyzing and judging traffic information. The traffic detection system can be divided into three parts: data collection and analysis, local processing, and cloud network traffic detection. The data collection and analysis module is mainly responsible for parsing traffic packets, attacking and updating traffic messages, and extracting and sending traffic characteristics. The collection and parsing of data packets is the process of parsing the data packets captured by the gateway and generating the specified data format. The statistics and updates of traffic messages involve updating traffic information. The process of generating traffic information features mainly involves transforming the parsed traffic into features such as address, number of packets, duration, port number, etc.

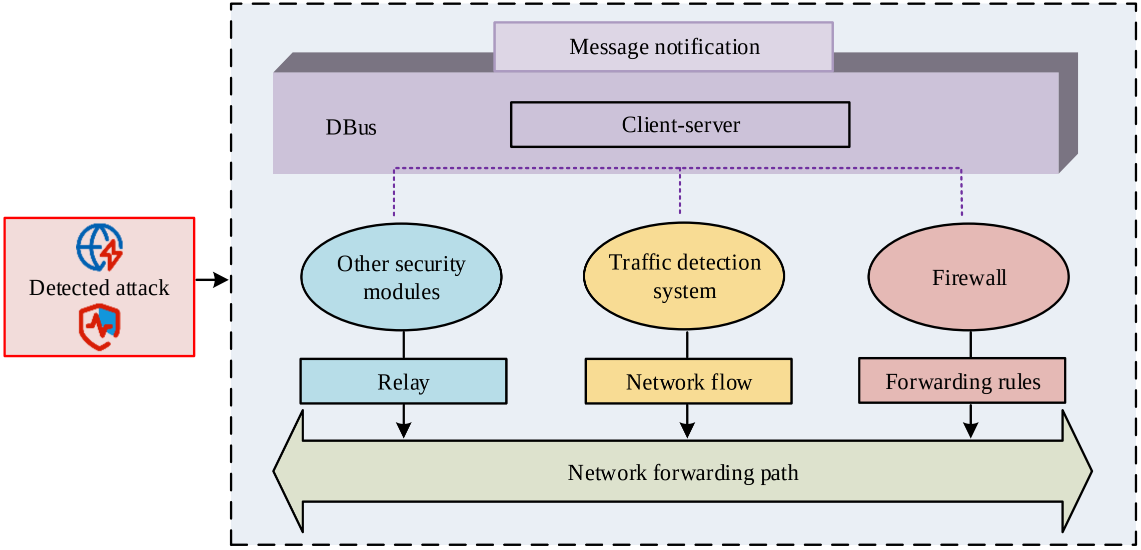


Fig. 5. Relationship between system and external modules.

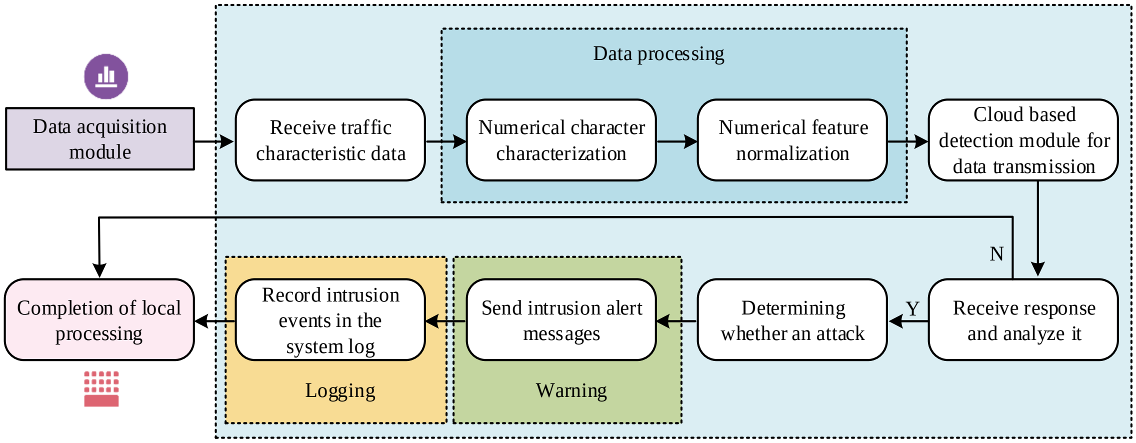

Data processing
Data acquisition module
Receive traffic characteristic data
Numerical character characterization
Numerical feature normalization
Cloud based detection module for data transmission
N
Completion of local processing
Record intrusion events in the system log
Logging
Send intrusion alert messages
Warning
Determining whether an attack
Y
Receive response and analyze it

Fig. 6. Flow chart of data local processing module.

IV. Results

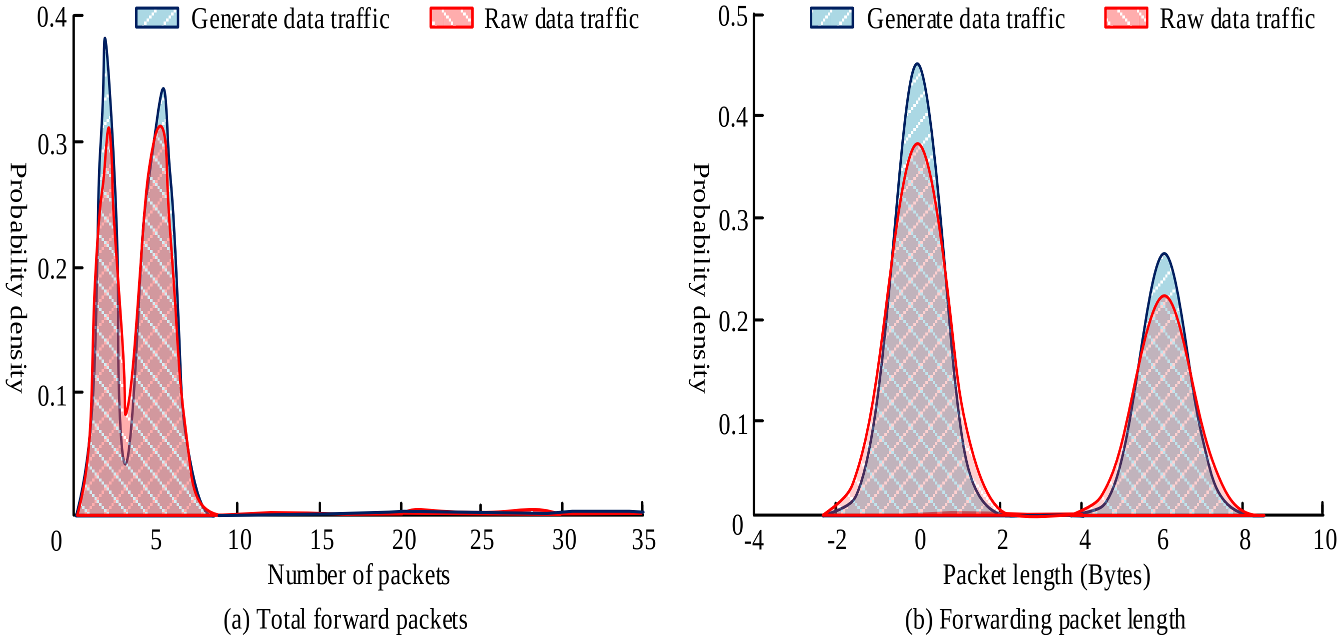
A. *Performance Testing of Traffic Monitoring Algorithms*

To verify the consistency between the original traffic information and the generated traffic information features based on feature fusion detection, the experiment analyzed the fitting situation before and after balancing from two aspects: actual forwarded data packets and packet length. The feature probability density distribution of the dataset was shown in Fig. 7.

In Fig. 7, the horizontal axis represented features, while the vertical axis represented the corresponding probability density. From Fig. 7(a), when the actual number of forwarded packets was less than 10, the probability density values of both the original data traffic and the preprocessed data traffic fluctuated between 0 and 0.4. When the number of forwarded packets was 2 and 5, the corresponding probability density could reach its maximum value. The probability densities of the original data traffic were 0.31 and 0.32, respectively, and the probability densities of the preprocessed generated data traffic were 0.38 and 0.35, respectively. In Fig. 7(b), when the probability density of the original data flow reached its maximum value of 0.45, the probability density of the generated data flow also reached its maximum value of 0.35. By observing the traffic feature distribution of the original data and the generated data, the feature distribution of the dataset was basically consistent before and after balancing. The data preprocessing method used in the study could ensure that the data features remained consistent before and after processing. The experiment referred to the single class support vector machine detection algorithm, autoencoder detection algorithm, and isolated forest detection algorithm as detection algorithms 1-3, respectively. The feature fusion detection algorithm proposed in the study was trained on traffic data on both IoTID20 and XIoTID datasets. The IoTID20 dataset was based on smart home environments and mainly collected data from terminal IoT devices corresponding to smart speakers, smartphones, and smart cameras. In simulated attack scenarios, smart speakers and smart cameras were targeted. This dataset contained 83 rich features, with over 70% of the features scoring over 0.5, which could improve the classification ability of detection algorithms and techniques and reduce training time. The XIoTID dataset contained 19 categories, with the majority accounting for 55.24% and the minority accounting for over 0.01%, respectively. From this, the comparison of training effects of different algorithms can be obtained as shown in Fig. 8.

On the IoTID20 dataset in Fig. 8(a), as the number of iterations increases, the accuracy of the four detection algorithms gradually improved and eventually stabilized. The feature fusion detection algorithm proposed in the study showed an increase in accuracy from the initial 77% to 98.3% after the 40th iteration. At this point, the accuracy of the compared detection algorithms 1, 2, and 3 was 95.7%, 94.2%, and 88.2%, respectively. After increasing the number of iterations to 50, the accuracy of the four algorithms remained basically unchanged. On the XIoTID dataset in Fig. 8(b), the accuracy of the four algorithms reached stability after the 45th iteration. At the 60th iteration, the accuracy of the feature fusion algorithm was the highest, at 84.2%. At this time, the accuracy of detection algorithms 1-3 were 79.5%, 76.8%, and 82.6%, respectively. The aforementioned outcomes may be attributed to the proposed spatial and temporal feature fusion methodology, which enabled the comprehensive integration of the original feature map and the deep feature map, and facilitates a thorough examination of the interrelationship between historical data from both temporal and spatial perspectives. This approach enhanced the informativity of the extracted feature data and markedly improved the detection performance. To conduct a more accurate analysis of the detection performance of feature fusion algorithms, the deep autoencoder algorithm was added to the existing comparison algorithms as comparison Algorithm 4 in the experiment. The accuracy, precision, recall, and F1 of each algorithm were recorded and analyzed. The comparison results could be seen in Fig. 9.

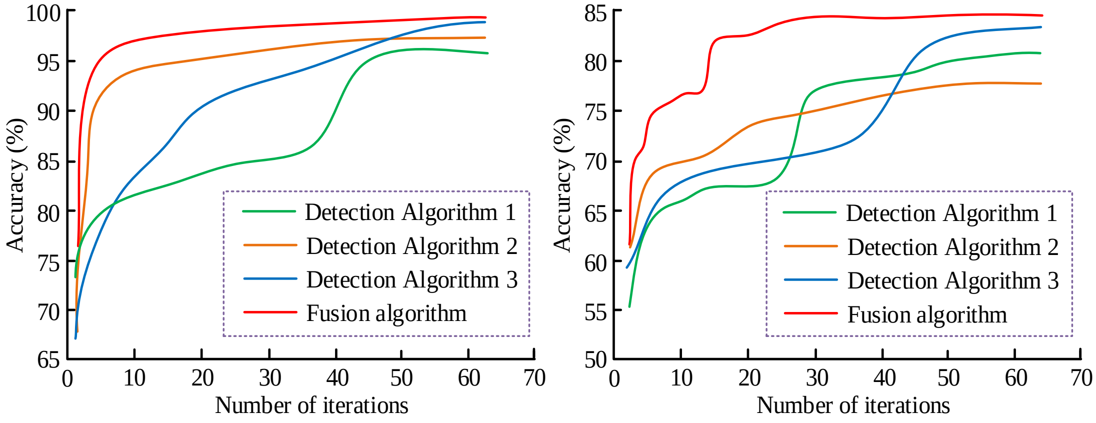
From Fig. 9, the comprehensive performance of the fusion algorithm was superior to the other four algorithms. On the KDD99-CUP dataset in Fig. 9(a), the F1 of the feature fusion algorithm could reach 93.16%, while the F1 of algorithms 1-4 were 81.36%, 67.89%, 90.56%, and 92.24%, respectively. On the MTA-KDD dataset in Fig. 9(b), the accuracy of the fusion algorithm was 92.91%, the recall rate was 88.12%, and the F1 was 91.54%, still higher than other algorithms. In Fig. 9(c), Algorithm 2 had the lowest F1 on the N-BaIoT dataset, at 65.02%, while the feature fusion algorithm had the highest F1, at 98.21%. On the MedBIoT dataset in Fig. 9(d), the recall rate of the feature fusion algorithm was 95.08%, and the F1 was 97.33%, which was the highest among the five algorithms. The above results were due to the tendency of the research method to lead to precise resolution of feature roots in low-dimensional feature spaces, thereby optimizing the fitting effect of the data distribution.



(a) Total forward packets

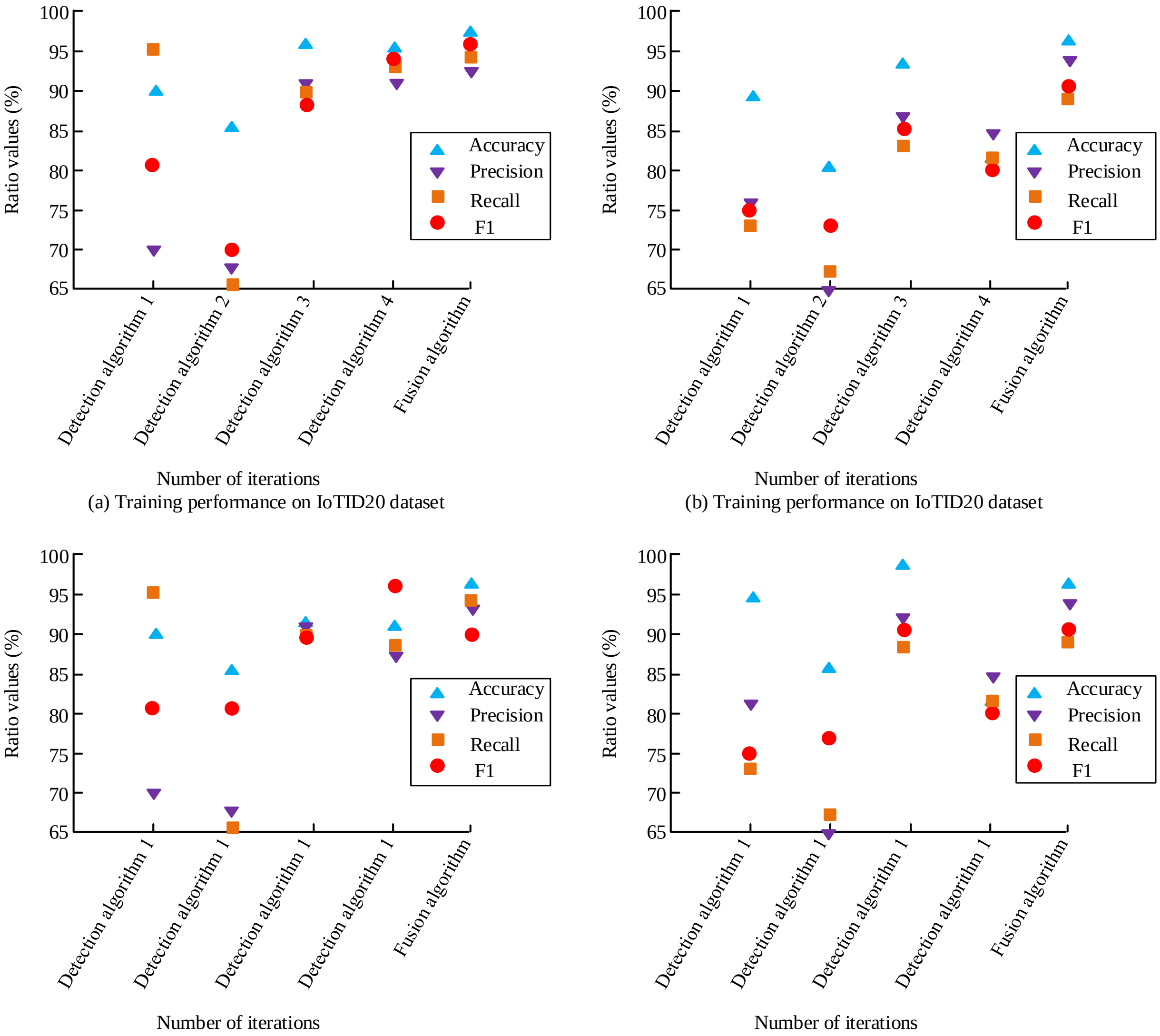
(b) Forwarding packet length

Fig. 7. Distribution of feature probability density in the dataset.



(a) Training performance on IoTID20 dataset (b) Training performance on XIIoTID datasetX

Fig. 8. Comparison of training effects of different algorithms.



(a) Training performance on IoTID20 dataset

(b) Training performance on IoTID20 dataset

(c) Training performance on IoTID20 dataset

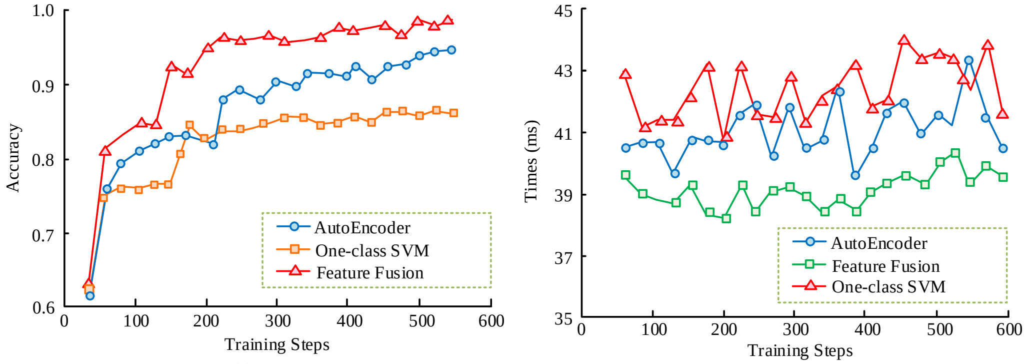
(d) Training performance on IoTID20 dataset

Fig. 9. Performance comparison of various algorithms.

B. IoT Malicious Traffic Detection Experiment

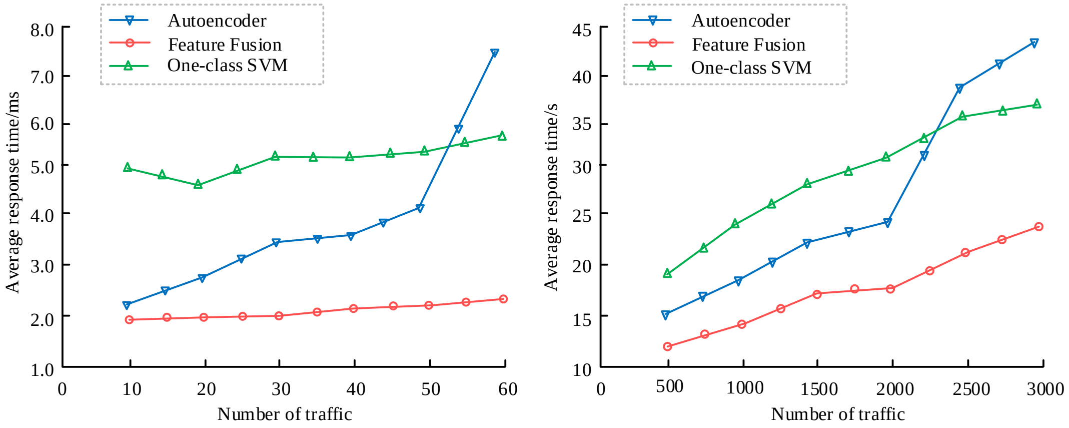
In the experiment on accuracy and model training time, the training steps were set to 600, and the automatic encoding detection model and single classification detection model were selected as controls. The comparison of accuracy and training time of the three models can be seen in Fig. 10.

In Fig. 10(a), when the training steps of the three models were less than 60, the difference in accuracy among the three models was small, and the accuracy of the feature fusion detection model was also below 0.8. As the number of training steps gradually increased, the accuracy of the three models begun to show significant differences. The detection accuracy of the feature fusion detection model proposed in the study was significantly higher than that of the other two models. When the training steps were 500, the accuracy of the feature fusion model reached 0.98, while at this time, the accuracy of the automatic encoding detection model and the single classification detection model were 0.91 and 0.84, respectively. In Fig. 10(b), there was not much difference in training time among the three models. The training time of the feature fusion detection model was slightly shorter than the other two detection models. After calculation, it was known that the average training time of the feature fusion detection model was reduced by 14.3% compared to the automatic encoding detection model and 17.43% compared to the single classification detection model. To verify the detection time of the model in the face of traffic attacks, the control model remained unchanged and the traffic quantity was divided into small-scale and large-scale tests. The comparison of traffic detection time for the three models was shown in Fig. 11.



(a) Accuracy (b) Training time

Fig. 10. Comparison of accuracy and training time.



(a) Small number of traffic (b) Large number of traffic

Fig. 11. Comparison of traffic detection time for various models.

In Fig. 11(a), when the number of attack traffic was small, the average detection time of the feature fusion detection model was relatively stable, and slowly increased with the increase of attack traffic. During the process of increasing the number of attack traffic from 10 to 60, the average detection time fluctuated between 2ms and 2.3ms. The growth rate of autoencoder detection models was relatively large, with the average detection time increasing from the initial 2.1ms to 7.8ms as the number of attack traffic increases. In Fig. 11(b), as the number of attack traffic increased significantly, the average detection time of the three detection models also increased significantly. When the number of attack traffic was 500, the average detection time of the feature fusion detection model was 12ms. When the number of attack traffic increased to 3000, the average detection time also increased to 23ms. At this time, the average detection time of the automatic encoding detection model and the single classification detection model were 43ms and 37ms, respectively. The malicious traffic detection time of the feature fusion detection model was shorter and the detection efficiency was higher than the other two types of detection models. To further verify the detection performance of the feature fusion detection model, 400 samples were selected for the experiment. 40% of the samples were used as the test model, and 60% of the samples were used as the training model. The analysis and recognition results of the fusion detection model on traffic types were shown in Fig. 12.

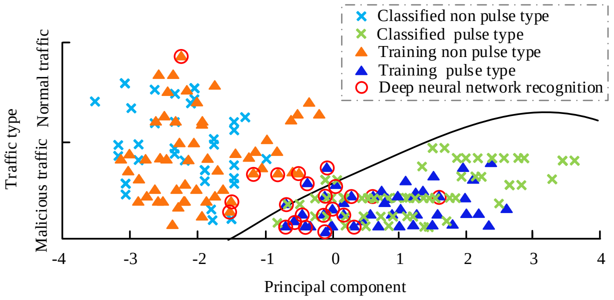


Fig. 12. Analysis and identification results of traffic types by fusion detection model.

In Fig. 12, the probability of correctly identifying the traffic samples used as training models was relatively high. Calculations showed that the accuracy of identifying malicious traffic on the training set was 96.89%. On the test set, 182 traffic samples were accurately identified, including 139 correctly identified malicious traffic and 43 correctly identified normal traffic, with recognition accuracy of 98.73% and 97.65%, respectively. Experimental results showed that both the training set for identifying attack traffic types and the test set for analyzing and judging traffic sample types had excellent performance, with high recognition rates and accuracy for traffic. It also proved that the detection algorithm that integrated temporal and spatial features could efficiently and reliably extract and judge traffic data features, with strong stability in detection and recognition.

To scientifically validate the performance of research methods, the latest malicious traffic detection algorithms were introduced for comparative experiments, namely convolutional neural networks and bidirectional gated space recurrent units (CNN-BiGSRU) based on convolutional neural networks and bidirectional gated space recurrent units, improved k-nearest neighbor based on cost sensitivity (IKN-CS) based on cost sensitivity, stacking and multi feature fusion (SMFF) based on multiple feature fusion, and a hybrid model based on deep learning algorithms and tabu search (DLTS). From this, the performance comparison of different malicious traffic detection algorithms can be obtained, as shown in Table I.

According to Table I, the research method had the best F1 score, accuracy, precision, and recall results, which were 99.2%, 99.5%, 99.6%, and 99.7%, respectively. In addition, the latest mainstream malicious traffic detection algorithms performed well, with all indicators exceeding 95%. The performance of the SMFF method was the worst, which may be due to the fact that the dataset used for testing contained normal and malicious traffic that were not of the same magnitude. However, this method was more suitable for detecting normal and malicious traffic of the same order of magnitude, but in practical application scenarios, normal and abnormal traffic were usually unbalanced. In summary, compared with mainstream methods, the research method still maintained excellent detection performance.

TABLE I. PERFORMANCE COMPARISON OF DIFFERENT MALICIOUS TRAFFIC DETECTION ALGORITHMS

Algorithm	**F1 value/%**	**Accuracy/%**	**Precision/%**	**Recall/%**
Feature fusion algorithm	99.2	99.5	99.6	99.7
DLTS	98.3	98.3	97.2	98.1
CNN-BiGSRU	97.9	98.1	98.4	97.6
IKN-CS	96.6	97.5	96.7	96.4
SMFF	95.8	96.4	96.3	95.7

V. DISCUSSION

To solve the problem of massive IoT data being vulnerable to attacks, an optimized malicious traffic detection algorithm is studied and designed, which integrates temporal and spatial features and enhances algorithm performance through mixed sampling and variational autoencoder data.

Tests on the IoTID20 dataset revealed that the accuracy of all four detection algorithms increased with the number of iterations and eventually stabilized. After the 40th iteration, the accuracy of the research algorithm increased from the initial 77% to 98.3%. The accuracy results of the XIoTID dataset were as follows, and the research method achieved the highest accuracy of 84.2% at the 60th iteration. Tang D et al. proposed a low-frequency DDOS attack detection algorithm based on multifeature fusion and sperm donation neural network, which could accurately detect DDOS attacks similar to normal traffic. The research results showed that fusing various network features into a feature map to represent the state of the network could effectively help improve the performance of the detection algorithm [23]. The above findings were consistent with this study due to the fact that the feature fusion algorithm integrally considered the correlation of the historical data before and after, which enriched the information content of the extracted feature data.

Compared with the current state-of-the-art malicious traffic detection methods, the results showed that the research method had the best F1 score, accuracy, precision, and recall rate, corresponding to 99.2%, 99.5%, 99.6%, and 99.7%, respectively. In addition, the latest mainstream malicious traffic detection algorithms perform well, with all indicators exceeding 95%. Liu Z et al. proposed a Bayesian meta-learning technique for the detection of encrypted malicious traffic to solve the problem of small sample size. The experimental results showed that when the sample size of malicious traffic was reduced to 100, the detection accuracy of the research model was 96.35% [24]. The study demonstrated that the accuracy of 98.3% can be achieved even with a limited sample size through the incorporation of a data augmentation processing method based on mixed sampling and variational autoencoder. This approach could effectively enhance the accuracy of research methods.

In summary, the research method can effectively improve the security level of IoT access devices, and ensure the security of information data processing and protection.

VI. Conclusion

To improve the detection capability of traffic detection systems for malicious traffic and achieve real-time security checks on IoT devices to achieve system security protection, an optimized malicious traffic detection algorithm was proposed. This research analyzed and identified traffic data by integrating temporal and spatial features, and used hybrid sampling and variational autoencoders to improve algorithm performance. As a result, in both the KDD99-CUP dataset and the XIoTID dataset, the performance of the proposed feature fusion algorithm was the highest, with an F-value of 93.16% in the former dataset and the highest accuracy of 84.2% in the latter dataset. Compared with the latest algorithms, the research method had the best F1, accuracy, precision, and recall results, which were 99.2%, 99.5%, 99.6%, and 99.7%, respectively. It also performed well against the latest mainstream malicious traffic detection algorithms, with all metrics exceeding 95%. Experimental results showed that the detection accuracy and efficiency of the IoT malicious traffic detection model based on feature fusion are high. However, the study only analyzed and identified normal and abnormal traffic, without conducting more in-depth identification and classification of abnormal traffic. Future research will further analyze the identified abnormal traffic to distinguish the types of unknown traffic attacks and achieve more favorable warning effects.

References

[1] Bandewad G, Datta K P, Gawali B W, & Pawar, S. N. Review on Discrimination of Hazardous Gases by Smart Sensing Technology. Artificial Intelligence and Applications. 2023, 1(2): 86-97

[2] Ahmad Muhammad Thantawi, Sri Astuti Indriyati. Conceptual Design Impacts in New Normal Era: The Use of Artificial Intelligence (AI) And Internet of Things (IOT) (Case Studies: Class Room and Restaurant). Acta Informatica Malaysia. 2022; 6(2): 39-42.

[3] Egrioglu E, Grosan C, Bas E. A new genetic algorithm method based on statistical-based replacement for the training of multiplicative neuron model artificial neural networks. Journal of supercomputing, 2023, 79(7): 7286-7304

[4] Senthilkumar M, Murugan BS. Enhancing the Security of An Organization from Shadow Iot Devices Using Blow-Fish Encryption Standard. Acta Informatica Malaysia. 2022; 6(1): 22-24.

[5] Doerr B. The Runtime of the Compact Genetic Algorithm on Jump Functions. Algorithmica, 2021, 83(10): 3059-3107

[6] Yang H, Xiong Z, Zhao J, Niyato D, Wu Q, Poor H, Tornatore M. Intelligent Reflecting Surface Assisted Anti-Jamming Communications: A Fast Reinforcement Learning Approach. IEEE transactions on wireless communications, 2021, 20(3): 1963-1974

[7] Salem A, Masouros C, Wong K. On the Secrecy Performance of Interference Exploitation With PSK: A Non-Gaussian Signaling Analysis. IEEE transactions on wireless communications, 2021, 20(11): 2014-2018

[8] Su N, Liu F, Wei Z, Liu Y, Masouros C. Secure Dual-Functional Radar-Communication Transmission: Exploiting Interference for Resilience Against Target Eavesdropping. IEEE transactions on wireless communications, 2022, 21(9): 7238-7252

[9] Du Z, Zhang F, Zhang Z, Yu W. Radar Detector in Uncoordinated Communication Interference Plus Partially Homogeneous Clutter. IEEE Communications Letters, 2021, 25(6): 1999-2003

[10] Hosseinali J, David M. Multicarrier Spectral Shaping for Non-White Interference Channels: Application to Aeronautical Communications in the L-Band. IEEE Transactions on Vehicular Technology, 2021, 70(10): 10686-10694

[11] Jiang T, Yu W. Interference Nulling Using Reconfigurable Intelligent Surface. IEEE Journal on Selected Areas in Communications, 2022, 40(5): 1392-1406

[12] Lin J, Lin D, Lu X, Chen J, Li C, Lin P, Huang C, Zheng Y. Using four-quadrant photodetector and improved genetic algorithm for visible-light positioning system. Optical Engineering, 2022, 61(4): 44107-44119

[13] Mahmoudi B, Motameni H, Mohamadi H. A new hybrid algorithm integrating genetic algorithm with Tabu search to solve imbalanced k-coverage problem in directional sensor networks. IET communications, 2023, 17(11): 1243-1254

[14] Shaikh M, Hua C, Jatoi M A, Ansari M, Qader A. Application of grey wolf optimisation algorithm in parameter calculation of overhead transmission line system. IET Science, Measurement & Technology, 2021, 15(2): 218-231

[15] Nagarajan L, Thangavelu S. Hybrid grey wolf sunflower optimisation algorithm for energy-efficient cluster head selection in wireless sensor networks for lifetime enhancement. IET communications, 2021, 15(3): 384-396

[16] Chen X, Rechavi O. Plant and animal small RNA communications between cells and organisms. Nature Reviews Molecular Cell Biology, 2021, 23(3): 185-203

[17] Luo J, Chen Z, Castellano D, Bao Bi, Han W, Li J, Kim G, An D, Lu W, Wu C. Lipids regulate peripheral serotonin release via gut CD1d. Immunity, 2023, 56(7): 1533-1547

[18] Nguyen D N, Dam H. Machine learning-aided Genetic algorithm in investigating the structure–property relationship of SmFe12-based structures. Journal of Applied Physics, 2023, 133(6), 63901-639010

[19] Greifenstein M, Dreizler A. MARSFT: Efficient fitting of CARS spectra using a library-based genetic algorithm. Journal of Raman Spectroscopy, 2021, 52(3), 655-663

[20] Yang R, Ma Y, Zhao M, Wei H, Qian L, Zhangyuan C, Aimin W, Szeyun S, Shinji Y, Zhigang Z. Flat visible spectrum by a genetic algorithm optimized photonic crystal fiber in the GHz comb spacing. Optics Letters, 2023, 48(11): 2829-2832

[21] Ji W, Ma K, Zhong L,Ying M, Guolin L. A Genetic Algorithm-Optimized Extreme Learning Machine Model for Process Ethylene Analysis Robustness Enhancement. Spectroscopy, 2021, 23(1): 44-48

[22] Rugo A, Ardagna C, Ioini N. A Security Review in the UAVNet Era: Threats, Countermeasures, and Gap Analysis. ACM Computing Surveys (CSUR), 2022, 55(1): 21-55

[23] Tang D, Tang L, Shi W, Zhan S, Yang Q. MF-CNN: a New Approach for LDoS Attack Detection Based on Multi-feature Fusion and CNN. Mobile Networks and Applications, 2020. 26(7):1705-1722.

[24] Liu Z, Lv Z, Zhao L, Li M, Liu X. A malicious traffic detection method based on Bayesian meta-learning for few samples. International Journal of Embedded Systems, 2023, 16(3):235-244.