Fusion of Improved LSTM with Graph Attention Networks for Malicious IoT Traffic Detection

Yonghong Li*
School of Information Engineering, Henan Institute of International Business and Economics, Zhengzhou 450002, China
*Corresponding author

Keywords: Internet of things, malicious traffic identification, improved LSTM, graph attention network, time series modeling

Received: January 6, 2025

In the rapid development of the Internet of Things, malicious traffic is highly mixed with normal business traffic, and exhibits strong long-term and short-term temporal dependencies and suddenness. Traditional detection methods have shortcomings in identifying covert attacks and reducing false positives. To this end, this study proposes an Internet of Things malicious traffic detection method that integrates an improved Long Short-Term Memory (LSTM) network with a Multi-Feature Graph Attention Network (MFGAT). In the temporal modeling stage, an attention mechanism and residual connections are introduced to enhance the representation of critical time-slice features, while lightweight gating and parameter compression strategies are employed to reduce model complexity. In the spatial modeling stage, a graph attention mechanism is utilized to weight the relationships among traffic nodes, enabling collaborative enhancement of multi-dimensional traffic features. The experimental results show that the area under the curve of the proposed method reaches 0.96, with an average accuracy of 0.91. The ablation experiment shows that after introducing improvement measures, the F1 value increases from 0.87 to 0.92, the parameter counts decreases from 5.13M to 4.23M, and the single sample inference delay is shortened from 7.64ms to 5.61ms. In complex scenarios, this method maintains an F1 score above 0.80 under highly imbalanced class distributions. conditions, with an average detection delay of only about 30ms under sudden attacks, and maintains long-term stability through rolling updates in concept drift scenarios. The proposed method not only outperforms existing methods in detection accuracy and robustness, but also has the advantages of lightweight and real-time performance, providing a feasible solution for efficient identification of malicious traffic in the Internet of Things environment.

Povzetek: Študija predstavlja učinkovit in lahek model za zaznavanje zlonamernega prometa v internetu stvari, ki izboljša natančnost, hitrost in zanesljivost odkrivanja tudi v zahtevnih razmerah.

1 Introduction

With the swift advancement of the Internet of Things (IoT), massive terminal devices are widely deployed in smart cities, industrial Internet, intelligent transportation, medical health and other scenarios, promoting the digital transformation of society [1]. However, the openness and heterogeneity brought about by large-scale access also make the IoT environment a high-risk area for network attacks. The IoT network traffic is not only large in quantity and complex in type, but also often highly mixed with normal business and malicious traffic [2]. Typical malicious attacks such as distributed DoS, botnet propagation, penetration scanning, and data theft often hide in seemingly normal sensor data reporting, firmware updates, and video transmission, with strong concealment and diversity [3]. At the same time, IoT traffic exhibits obvious long-term and short-term temporal dependencies and burst features, with both regular periodic communication and the possibility of concentrated attacks erupting within an extremely brief timeframe. These features not only require higher accuracy and generalization ability of the detection model, but also make it particularly difficult to achieve real-time detection under limited computing and storage resources [4-5]. Although some progress has been made in identifying malicious traffic, there are still overall shortcomings. On the one hand, existing methods have limited modeling capabilities for long short-term dependencies and sudden features, making it difficult to maintain stability in complex scenarios. On the other hand, many studies have overlooked the interaction between traffic, resulting in incomplete characterization of attack propagation patterns [6]. In view of this, a method for identifying malicious traffic in the IoT based on an improved Long Short-Term Memory (LSTM) network is proposed. This method introduces attention mechanisms, residual connections, and lightweight gating in structure to enhance the attention of key features, improve the stability of long sequence modeling, and reduce computational overhead. At the same time, graph attention networks capture traffic topology interaction information, providing more discriminative inputs for temporal modeling. The research

aims to elevate the precision, stability, and instantaneous efficiency of malicious traffic identification in the IoT environment, while also considering the lightweight and deployable nature of the model.

Table 1: Comparison of existing intrusion detection methods

Reference	Method type	Dataset/scenario	Representative performance	Main limitation
Fallah S et al. [8]	LSTM	Smartphone traffic	Acc > 90%, AUC ≈ 99.9%	Limited adaptability to bursty and complex attacks
Ullah F et al. [9]	Transformer + CNN-LSTM	Public network datasets	F1 significantly improved	High structural complexity and computational cost
Han D et al. [10]	Lightweight CNN + Attention	IoT traffic	Acc > 99.9%	Limited modeling of traffic correlations
Ali S et al. [11]	Multitask LSTM	IoT malware traffic	Acc > 95%	Limited generalization to unknown attacks
Anitha T [12]	BiLSTM + CNN	Smart device traffic	Acc ≈ 99.98%	Insufficient modeling of long-term dependencies
Shi G [13]	GAN-enhanced model	Industrial IoT traffic	Improved robustness	Complex training and limited stability
Li Z [14]	DAE + GAN	Public anomaly datasets	Acc > 98%	Performance depends on generation quality, limited generalization

To further clarify the research objectives, this study focuses on the following research questions: (RQ1) In IoT traffic scenarios with highly imbalanced class distributions, can a detection model that integrates a graph attention network with an improved LSTM outperform conventional deep learning methods in terms of detection accuracy and false alarm control? (RQ2) In dynamic network environments with burst attacks and concept drift, can the proposed fusion model maintain stable detection performance with low detection latency? (RQ3) While ensuring detection performance, can the introduction of attention mechanisms, residual connections, and lightweight gating designs effectively reduce model complexity and improve its deployment feasibility in IoT scenarios?

2 Literature review

With the popularization of the IoT and intelligent terminals, network traffic presents features such as large scale, complex features, and diverse attack types. Malicious traffic identification and anomaly detection have become important research directions in the realm of network security. In recent times, a large quantity of studies have introduced deep learning and machine learning methods into traffic detection tasks, improving detection performance by mining temporal features, spatial features, and behavioral patterns [7]. Significant achievements have been made in the detection of malicious software on mobile terminals, traffic recognition in IoT environments, and anomaly detection in industrial IoT, providing new ideas for handling intricate and varied network assaults. Fallah S et al. introduced a traffic sequence modeling method based on LSTM to address the threat of smartphone malware to user privacy and security. This method could distinguish between malicious and normal traffic while identifying unknown malware families, with a detection accuracy of over 90% and an area under the curve of 99.9% [8]. Ullah F et al. introduced a transfer learning approach based on converters to address the problems of complex network traffic features and class imbalance. This method combined oversampling with a hybrid structure of convolutional neural networks and LSTM, effectively improving the detection capability of minority class attacks and achieving high performance on multiple public datasets [9]. Han D et al. introduced a lightweight anomaly traffic detection model to address the limitations of computing and storage in IoT terminals. By combining an improved mobile network model with coordinate attention mechanism, the model achieved detection accuracy of over 99.9% on public datasets and demonstrated good deployment adaptability [10]. Ali S et al. introduced a behavior traffic analysis method based on multi-task deep learning for the detection and classification of malicious software traffic in the IoT. LSTM was used to identify traffic and distinguish malicious software types, achieving a classification accuracy of over 95% on large-scale traffic data from multiple devices [11].

In addition, Anita T et al. introduced a detection approach that combines bidirectional LSTM and convolutional neural network to address the real-time and accuracy issues of malicious traffic detection in smart devices, achieving a detection accuracy of 99.98% and extremely low false alarm rate while reducing prediction time [12]. Shi G et al. introduced a deep detection model grounded in high-order features and generative adversarial strategy to address the problem of relying on manual features and insufficient adaptability in detecting abnormal traffic in industrial IoT. This improved the expression ability of abnormal features and achieved higher robustness and detection performance on real industrial IoT data [13]. Li Z et al. introduced a deep detection method based on pseudo anomaly generation to address the problem of high false alarm rate in semi supervised abnormal traffic detection. By combining feature extraction framework with denoising autoencoder and generative adversarial network, the detection accuracy was improved, with an accuracy of over 98% on public datasets [14]. The detailed summary of the relevant research is presented in Table 1.

Overall, existing research has made positive progress in detection accuracy, feature modeling, and lightweight deployment, but there are still shortcomings. Many methods only focus on a single dimension of time or

space, failing to fully integrate topological dependencies and temporal features. When facing complex scenarios such as class imbalance, sudden attacks, and concept drift, the robustness and long-term stability of the model are still insufficient. Therefore, the study proposes a fusion graph attention mechanism and an improved LSTM recognition method. Compared with existing methods, the novelty of the research consists in the deep fusion of graph attention mechanism and improved LSTM, achieving the organic combination of topological and temporal features, thus more comprehensively characterizing malicious attack behavior.

3 Methods and materials

3.1 Improved LSTM method for malicious traffic recognition

In the context of the IoT, malicious network traffic is highly mixed with normal business and exhibits strong long- and short-term temporal dependencies and suddenness, which limits the effectiveness of traditional temporal models in identifying covert attacks and controlling low false positives [15-16]. To address this issue, a method for identifying malicious network traffic in the IoT based on improved LSTM is proposed, which enhances detection accuracy and robustness by focusing on key temporal segments and modeling cross window dependencies. Before improvement, in order to clarify the attack link and learnable feature space, it is necessary to first construct a malicious network traffic threat model for the IoT, which can guide feature extraction, sample construction, and model input design. The threat model is shown in Figure 1.

As shown in Figure 1, there are both normal traffic in IoT communication, such as sensor data reporting, device status queries, firmware updates and heartbeat detection, video and audio transmission, as well as malicious traffic injected by attackers. After entering the device through the network, this traffic is collected, stored, and sent to the detection module for identification. To align the threat model with the actual data distribution, the CIC-IDS2017 and UNSW-NB15 datasets adopted in this study contain multiple typical attack types, including denial-of-service attacks (DoS/DDoS), probing and scanning, botnet communications, and infiltration attacks. In these datasets, benign traffic accounts for approximately 70%–80% of the total samples, while attack traffic represents about 20%–30%, with a highly imbalanced distribution among different attack categories. For example, DoS-related attacks constitute more than 40% of all attack samples, whereas probing and infiltration attacks appear in much smaller proportions. These statistical characteristics are consistent with the threat scenario illustrated in Figure 1, where benign traffic is mixed with multiple types of malicious traffic, and they provide a data foundation for evaluating the proposed model under class imbalance and burst-attack conditions.

This threat model reveals the main challenges in identifying malicious traffic in the IoT. Firstly, normal and malicious traffic are highly mixed, and attack behavior is often covert. Secondly, traffic has strong temporal dependence, with both periodic patterns and sudden anomalies [17]. Thirdly, identification needs to ensure real-time performance with limited resources. Therefore, research on improving traditional LSTM includes three aspects, namely introducing attention mechanism to highlight key features, increasing residual connections to enhance temporal modeling capability, and lightweight optimization of gate units to improve deployment efficiency. The framework of the improved LTSM is shown in Figure 2.

As shown in Figure 2, the attention layer is set at the input end, which constructs context by combining the hidden state (HS) of the previous moment with the input of the current or adjacent time slice, making the sequence representation before entering the gating unit more focused on key features. Residual connections directly cross from the previous HS to the current output, and perform additive fusion with the results inside the unit to preserve the original information and enhance the stability of long sequence training. Lightweight gating mainly functions on the forget gate (FG) and input gate (IG), effectively reducing computational complexity and storage overhead through parameter sharing and low rank decomposition without significantly weakening the model's expressive power. The core idea of the attention layer is to enable the model to give higher attention to key segments in the traffic sequence before sequence modeling, and its weighting process is shown in equation (1).

$$\alpha_t = \frac{\exp\left(h_{t-1}^{\bullet} W_a x_t\right)}{\sum_k \exp\left(h_{t-1}^{\bullet} W_a x_k\right)}, \quad x_t' = \alpha_t x_t \tag{1}$$

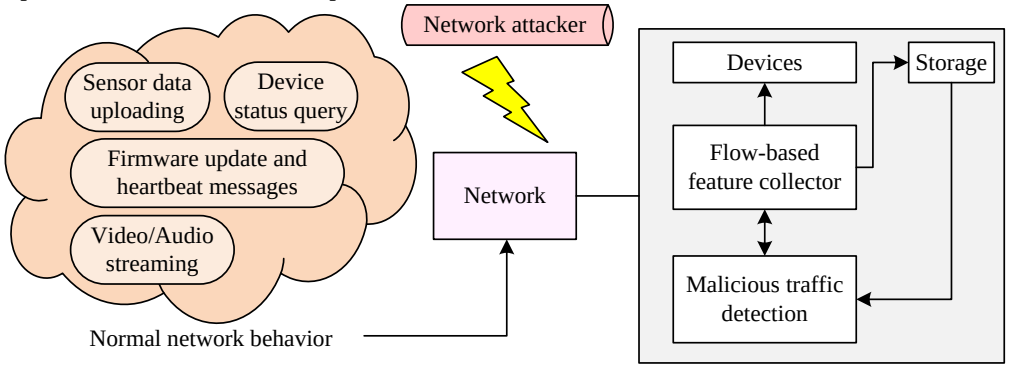


Figure 1: Threat model of malicious network traffic in the IoT

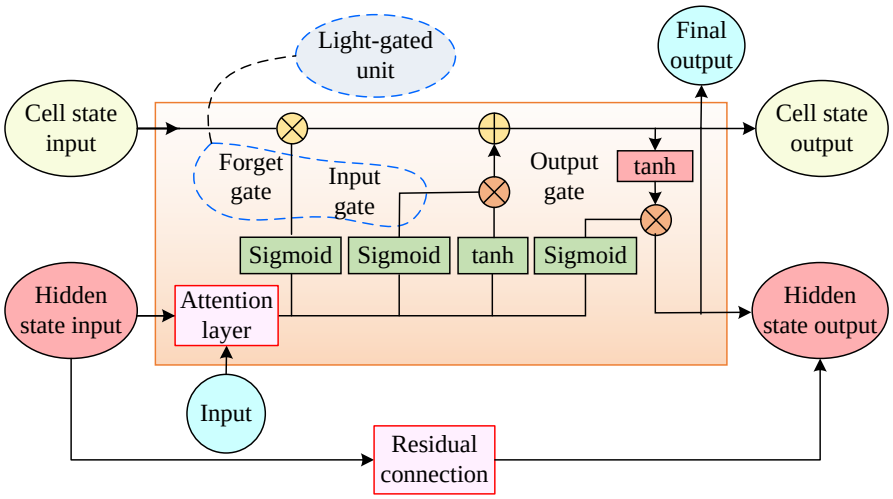


Figure 2: Schematic diagram of the improved LSTM structure

In equation (1), $h_{t-1} \in \mathbb{R}^{d_h}$ represents the HS vector of the previous time step, which contains historical time sequence information, and $x_t \in \mathbb{R}^{d_x}$ represents the input feature vector of the current time step, such as packet size, interval, protocol features, etc. $W_a \in \mathbb{R}^{d_h \times d_x}$ represents the attention parameter matrix, which is used to calculate the correlation between the input and the HS. α_t represents the attention weight, which represents the importance of the input at the current time step; the larger the value, the more critical the feature is [18]. x_t' represents the weighted input vector, which incorporates time sequence context information and is more conducive to identifying hidden malicious traffic mixed in with normal communication. Lightweight optimization shares and compresses some parameters, as shown in equation (2).

$$f_t = \sigma\left(W_f z_t + b_f\right), \quad i_t = \sigma\left(W_i z_t + b_i\right) \quad (2)$$

In equation (2), $f_t \in \mathbb{R}^{d_h}$ represents the FG vector, which controls how much information the memory unit retains in the previous time step. $i_t \in \mathbb{R}^{d_h}$ represents the IG vector, which controls the proportion of the current input updating the memory unit. $z_t \in \mathbb{R}^{d_h + d_x}$ represents the concatenation vector, which consists of the HS in the previous time step and the weighted input. $W_f \in \mathbb{R}^{d_h \times (d_h + d_x)}$ and $W_i \in \mathbb{R}^{d_h \times (d_h + d_x)}$ represent the weight matrices of the FG and the IG , respectively. Due to the use of lightweight optimization, some parameters of these two matrices are shared or compressed [19]. To further reduce the parameter scale, the study adopts the low-rank decomposition form of the weight matrix, thereby significantly reducing the number of parameters and computational overhead while maintaining the gating expressive power. b_f and b_i represent the bias vectors of the FG and the IG, respectively. $\sigma(\cdot)$ is the Sigmoid activation function, which maps the result to the (0,1) interval, which is convenient for gating adjustment. To address the problem of gradient decay that easily occurs in long sequence learning, the study designs a residual connection at the output end, which directly introduces the historical HS into the calculation of the current output, as shown in equation (3).

$$h_t = o_t \odot \tanh\left(c_t\right) + r_t \odot h_{t-1} \quad (3)$$

In equation (3), $h_t \in \mathbb{R}^{d_h}$ represents the HS output at the current time, serving as the final temporal feature representation. $o_t \in \mathbb{R}^{d_h}$ is the output gate vector, controlling which part of the memory cell is output. c_t is the memory cell state at the current time, combining the history and the current input. $\tanh\left(c_t\right)$ is the hyperbolic tangent function, performing nonlinear compression on the memory cell state , with a numerical range of (-1, 1). r_t is the residual control coefficient, used to adjust the influence of the historical HS on the current output. $\odot$ represents element-wise multiplication, ensuring that the gating operation adjusts each dimension independently. Therefore, based on the improved LSTM, the process of identifying malicious traffic is shown in Figure 3.

As shown in Figure 3, the raw network traffic is first collected from devices such as sensors, cameras, and terminals, and redundancy removal, session partitioning, and format standardization are completed through data preprocessing. Subsequently, statistical features such as packet size and time interval, temporal features (long short-term dependencies), and protocol features are extracted and normalized, and fed into an improved LSTM as inputs. The improved LSTM in turn strengthens the attention to key abnormal segments through the attention layer, reduces the computational overhead by using lightweight gating, and stabilizes the long sequence modeling through residual connection, so as to obtain a more discriminative time series representation. Finally, the classifier outputs recognition results based on the Softmax function, dividing the traffic into normal communication or malicious attacks, achieving efficient detection of malicious traffic in complex IoT environments.

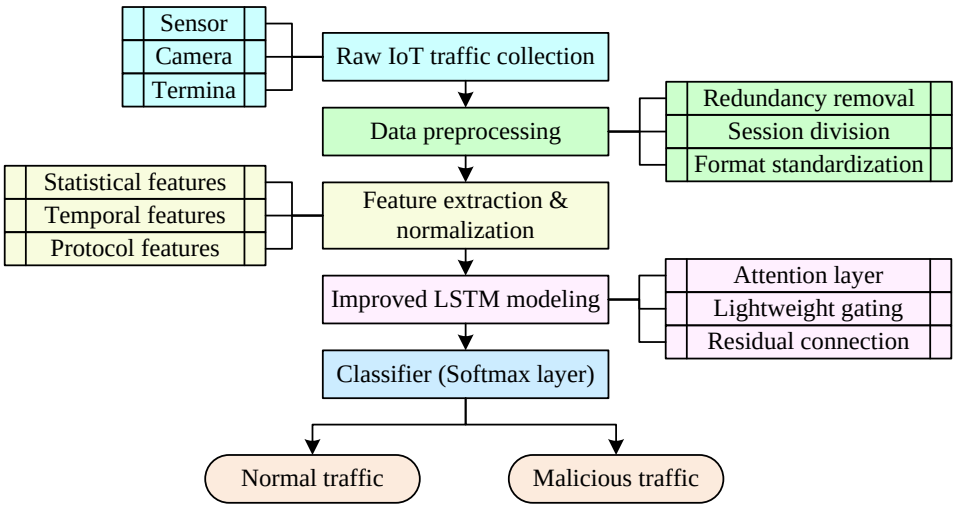


Figure 3: Malicious traffic identification based on improved LSTM

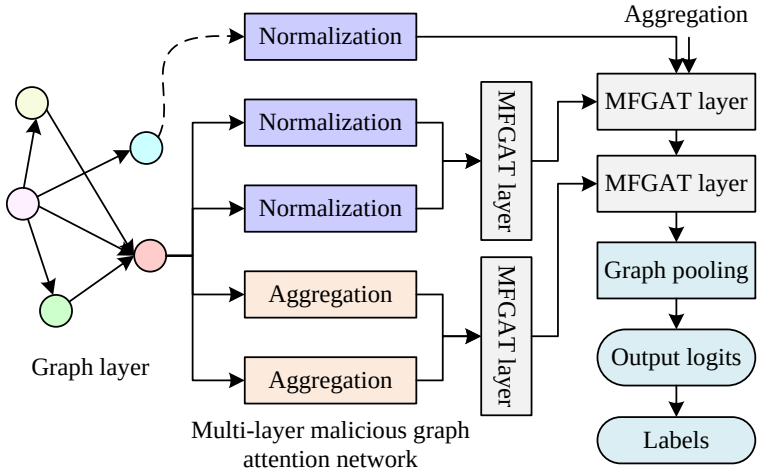


Figure 4: Structural schematic diagram of MFGAT

3.2 Improved Fusion Method of LSTM and MFGAT

The improved LSTM introduces attention mechanisms, residual connections, and lightweight gating, enhancing the temporal modeling capability for malicious traffic in the IoT. However, network traffic is not only manifested as a time series, but also contains complex interactive relationships between devices and sessions. This relationship serves a crucial function in the propagation and evolution of malicious attacks, and relying solely on temporal features often makes it difficult to fully characterize attack behavior [20]. To this end, a Malicious Flow Graph Attention Network (MFGAT) is introduced based on the improved LSTM to capture the dependencies of traffic on topology and interaction patterns.

In the construction of the traffic graph, network traffic sessions (flows/sessions) are treated as graph nodes. Each node corresponds to a traffic session defined by a five-tuple (source IP, destination IP, source port, destination port, and protocol). The node features are composed of statistical and temporal characteristics of the session, including the number of packets, average packet length, inter-arrival time, and protocol type. For edge construction, both temporal proximity and communication correlation are considered to establish connections between nodes. Specifically, if two traffic sessions occur within the same time window ΔT and satisfy at least one of the following conditions: sharing the same source address, sharing the same destination address, or using the same protocol type, an undirected edge is created between the corresponding nodes. This strategy captures the bursty behavior of malicious traffic in short periods as well as its propagation relationships at the communication level. To accommodate the time-varying characteristics of traffic distributions, a sliding time window mechanism is adopted to construct a dynamic graph. For each time window, a corresponding subgraph is generated and fed into the MFGAT for feature enhancement, rather than constructing a single static graph over the entire dataset. The basic structure of MFGAT is shown in Figure 4.

As shown in Figure 4, the overall structure of MFGAT takes layers as input, with nodes representing traffic entities and edges representing their interaction relationships. In each layer, the model calculates the correlation coefficient between nodes and neighbors

through a multi-head attention mechanism, and uses weighted aggregation to update features, thereby highlighting key interaction patterns. The outputs of each layer are normalized and feature aggregated, ensuring the stability of the training process and integrating multi-scale node representations. Subsequently, node level features are compressed into a global graph representation through graph pooling operations, and finally input into a classifier to generate predicted labels. Assuming the input feature of node i is x_i, a low dimensional projection is first obtained through linear transformation, as shown in equation (4).

$$z_i = Wx_i \tag{4}$$

In equation (4), W is the learnable weight matrix. The attention correlation score between node i and neighbor j can be expressed as equation (5).

$$e_{ij} = \text{LeakyReLU}\left(a^T \left[z_i \| z_j \right]\right) \tag{5}$$

In equation (5), $\left[z_i \| z_j \right]$ represents vector concatenation, α is the attention parameter vector. The correlation scores are normalized to obtain attention weights, as shown in equation (6).

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in N} \exp\left(e_{ik}\right)} \tag{6}$$

On the basis of equation (6), the update feature of node i is shown in equation (7).

$$x_i' = \sigma\left(\sum_{j \in N} \alpha_{ij} W x_j\right) \tag{7}$$

In equation (7), σ is the nonlinear activation function. To enhance expression ability, MFGAT adopts a multi-head attention mechanism, which calculates attention weights in parallel in the K sub-spaces and concatenates them, as shown in equation (8).

$$x_i' = \Big\|_{k=1}^{K} \sigma\left(\sum_{j \in N} \alpha_{ij}^{(k)} W^{(k)} x_j\right) \tag{8}$$

In equation (8), $\alpha_{ij}^{(k)}$ represents the weight of the k th attention head and $W^{(k)}$ is the corresponding projection matrix. Based on the above content, MFGAT serves as a pre feature enhancement module for improving LSTM in the malicious network traffic recognition process. It can model the topological interaction relationships between traffic entities and provide more discriminative input representations for subsequent temporal modeling. The data processing between MFGAT and improved LSTM is shown in Figure 5.

As shown in Figure 5, the features output by MFGAT cannot be directly used as the input of the improved LSTM, since the former are mainly node-level representations, whereas the latter requires fixed-dimensional features arranged in a strict temporal order. Therefore, a "pooling–alignment–serialization" transformation is designed. First, in the pooling stage, multiple node features generated by MFGAT within the same time window are aggregated into a single window-level vector using a pooling operation, so that node sets with different sizes across windows are mapped to feature representations with identical length. Second, in the normalization and alignment stage, the aggregated features are normalized and projected to the input dimension required by the improved LSTM, ensuring consistent scale and structure among different time windows. Finally, in the serialization stage, the window-level features are arranged according to the temporal order of windows to form a standard input sequence.

For example, if the first window contains 20 traffic sessions and the second window contains 35 sessions, MFGAT outputs 20 and 35 node features, respectively. After pooling, both are converted into window-level vectors with the same length, which are then concatenated in temporal order to form the input sequence. Through this process, the topological interaction information extracted by MFGAT is effectively mapped into the temporal feature space, enabling a unified representation of structural and temporal information. Therefore, by combining MFGAT with improved LSTM, the final process for identifying malicious network traffic in the IoT is presented in Figure 6.

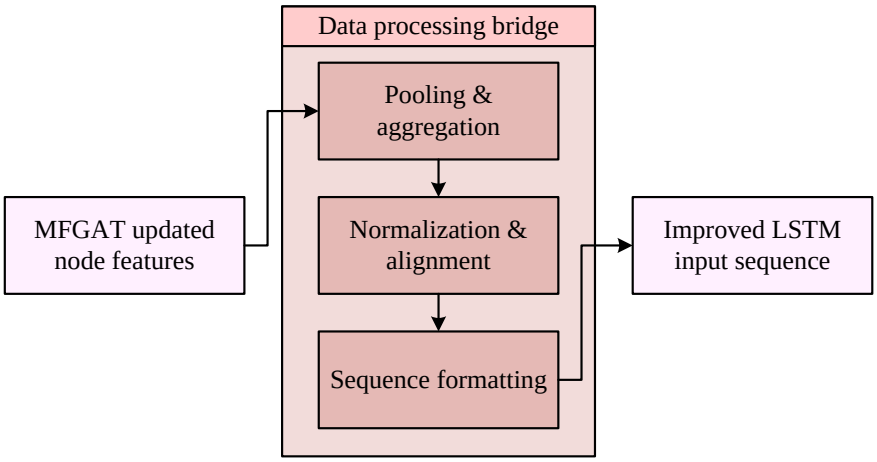


Figure 5: Data processing pipeline between MFGAT output and Improved LSTM input

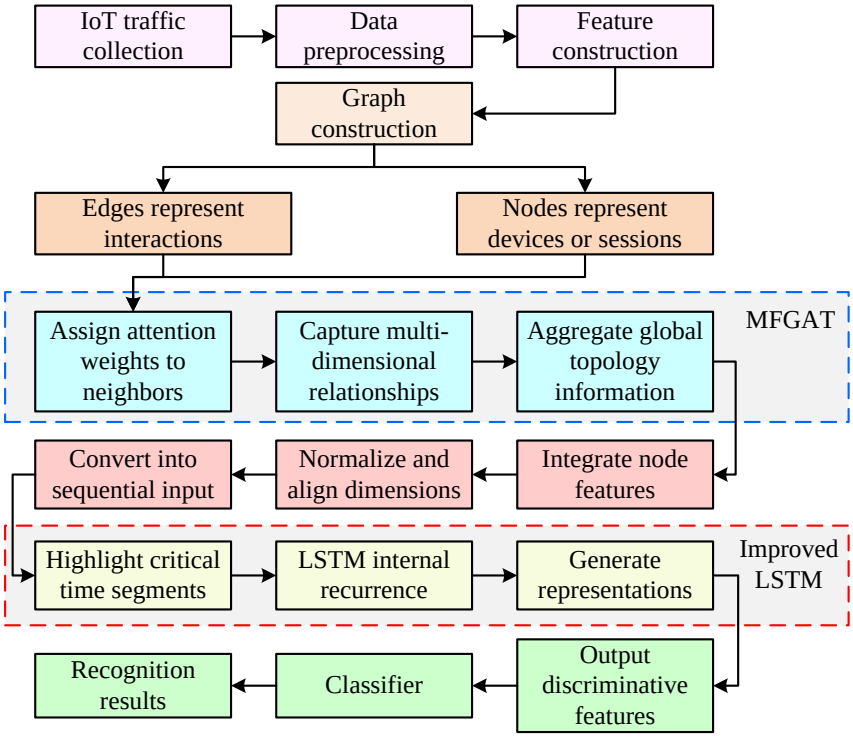


Figure 6: Identification process of malicious network traffic in the IoT

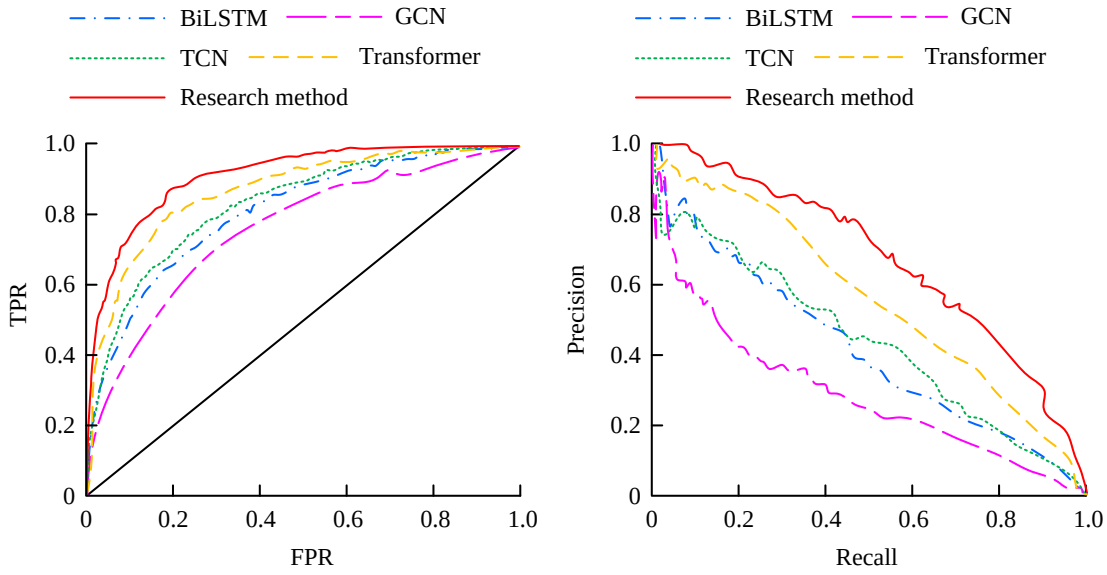
Table 2: Method pseudocode

Input: Traffic session set F, window size T, step size S Feature extractor Phi, pooling method Pool Model parameters for MFGAT, Improved LSTM, and Classifier
Output: Predicted label Y
Split the traffic timeline into windows W1 to Wn using T and S
Initialize an empty sequence List
For each window Wi do
Select sessions Vi that occur in Wi
Extract node features Xi from Vi using Phi
Initialize an empty edge set Ei
For each pair of sessions in Vi do
If they are close in time and share the same source, destination, or protocol then
Add an undirected edge between them into Ei
End if
End for
Construct graph Gi using Vi, Ei, and Xi
Apply MFGAT on Gi to obtain enhanced node features Hi
Aggregate Hi using Pool to obtain a graph-level feature Zi
Normalize Zi to remove scale differences
Align Zi to a unified feature dimension
Append Zi into List in temporal order
End for
Feed List into the Improved LSTM to obtain temporal representation R
Input R into the classifier and output predicted label Y

As shown in Figure 6, unlike relying solely on improved LSTM for malicious traffic identification, the introduction of MFGAT enhancement module adds a modeling step to the traffic graph structure before temporal modeling in the identification process. The research explicitly extracts topological dependency features between traffic entities through neighbor weight allocation, multidimensional relationship capture, and global information aggregation. After feature integration, normalization, and serialization, it is input into an improved LSTM to capture both long and short-term dependencies while preserving key topological information. The final output representation combines both temporal and structural features, which can effectively improve the recognition accuracy and robustness of malicious traffic in complex IoT environments. The pseudo-code of the method is shown in Table 2.

Table 3: Experimental environment and parameter settings

Category	Configuration
Hardware	CPU: Intel i9-12900K; GPU: NVIDIA RTX 3090 (24GB); Memory: 64GB RAM; Storage: 2TB SSD
Operating system	Ubuntu 22.04 LTS
Software	Python 3.10; PyTorch 2.1; PyTorch Geometric 2.4; CUDA 12.1
Optimizer	AdamW (learning rate = 3e-4, weight decay = 1e-4, betas = (0.9, 0.999))
Learning rate schedule	Cosine annealing (minimum lr = 1e-6)
Epochs	80
Batch size	128
Loss function	Weighted BCE
MFGAT parameters	Number of layers L = 2; attention heads h = 4; hidden dimension = 128; Dropout = 0.2
Improved LSTM parameters	Hidden units = 256; number of layers = 2; Dropout = 0.2; residual coefficient λ = 0.2; lightweight gating low-rank rank r = 32; parameter sharing ratio p = 0.5



(a) Comparison of ROC curves

(b) Comparison of PR curves

Figure 7: Overall validity assessment results

4 Results

To confirm the validity and superiority of the proposed improved LSTM method for identifying malicious traffic in the IoT based on the MFGAT enhancement module, the study selected two public datasets, UNSW-NB15 and CIC-IDS2017, as data sources. The former includes normal traffic and nine typical attacks, covering multiple protocols, while the latter covers multiple types of attacks such as DoS attacks, infiltration, and botnets, with a large scale and close to actual scenarios. The preprocessing procedure included session partitioning, redundancy removal, feature extraction, and z-score normalization. After session construction, all sessions were sorted according to their timestamps and then divided into training, validation, and test sets in a chronological order with a ratio of 6:1:3. This ensured that the samples in the test set were strictly later in time than those in the training set, thereby avoiding temporal leakage caused by the same attack events appearing across different subsets, and guaranteeing the fairness and reproducibility of the experimental results. The experimental environment and parameter design are presented in Table 3.

On the basis of Table 3, the study first conducted an overall effectiveness evaluation. Bidirectional Long Short-Term Memory (BiLSTM), Temporal Convolutional Network (TCN), Graph Convolutional Network (GCN), and Transformer models were selected as baseline methods for comparison. All baseline models adopted the same data preprocessing pipeline as the proposed method and used an identical class-weighting strategy to alleviate class imbalance. During training, all baseline models and the proposed method shared the same training/validation/test split and were tuned based on the validation set. Specifically, the same hyperparameter search ranges were defined for all models, and the same tuning strategy was applied to select the optimal configurations, thereby avoiding performance bias caused by different parameter settings and ensuring the comparability and fairness of the experimental results. The Receiver Operating Characteristic (ROC) curve and Precision Recall (PR) curve of the subjects are shown in Figure 7.

Table 4: Comparative analysis of comprehensive performance

Method	Precision (%)	Recall (%)	F1 score (%)	FPR (%)
Transformer	89.35 ± 0.74	86.28 ± 0.69	87.76 ± 0.71	5.92 ± 0.55
TCN	85.83 ± 0.81	82.46 ± 0.77	83.61 ± 0.79	7.68 ± 0.63
BiLSTM	81.74 ± 0.93	78.52 ± 0.86	79.98 ± 0.89	10.94 ± 0.71
GCN	76.58 ± 1.04	72.36 ± 0.97	74.41 ± 1.01	13.82 ± 0.84
GCN-LSTM	88.21 ± 0.72	85.04 ± 0.68	86.59 ± 0.70	6.48 ± 0.57
GAT-LSTM	90.12 ± 0.66	87.35 ± 0.62	88.71 ± 0.64	5.21 ± 0.49
Research method	93.47 ± 0.63	90.86 ± 0.58	91.92 ± 0.61	3.87 ± 0.42

Table 5: Outcomes of the ablation experiment for improved LSTM

Model Variant	Base LSTM	√	√	√	√	√	√	√	√
	Attention	×	√	×	×	√	√	×	√
	Residual	×	×	√	×	√	×	√	√
	Lightweight Gates	×	×	×	√	×	√	√	√
Index	F1	0.870 ± 0.012	0.898 ± 0.011	0.884 ± 0.013	0.878 ± 0.014	0.909 ± 0.010	0.901 ± 0.012	0.892 ± 0.013	0.920 ± 0.009
	PR-AUC	0.880 ± 0.010	0.907 ± 0.009	0.897 ± 0.011	0.889 ± 0.012	0.919 ± 0.009	0.911 ± 0.010	0.903 ± 0.011	0.930 ± 0.008
	p	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05	<0.05
	Parameters (M)	5.13	5.58	5.46	3.72	6.04	4.11	3.94	4.23
	Inference Latency (ms/sample)	7.64	8.27	7.85	5.19	8.83	5.77	5.48	5.61

From Figure 7 (a), the proposed method performed better than the comparison model at various thresholds, with its curve closest to the upper left corner and an Area Under the Curve (AUC) of 0.96. In contrast, the AUC of Transformer and TCN were 0.92 and 0.88, respectively. Although they performed well in the medium to high true positive rate range, their performance slightly decreased under low false positive rate (FPR) requirements. The AUC of BiLSTM and GCN were 0.85 and 0.81, respectively, and the curves were closer to the diagonal, indicating insufficient overall discriminative ability. As shown in Figure 7 (b), the proposed method was also superior to the comparative method, with an average accuracy of 0.91, and could maintain high accuracy throughout the entire recall range. This meant that in actual IoT environments, the methods proposed in the research could reduce false positives while ensuring the capture of the vast majority of malicious traffic. In contrast, the average accuracy of Transformer and TCN were 0.87 and 0.83, respectively, with a faster decrease in accuracy in high recall areas. The average accuracy of BiLSTM and GCN was only 0.79 and 0.74, indicating insufficient ability to identify minority attacks. Therefore, the proposed method had robustness and superiority in both imbalanced and high recall scenarios. To further quantitatively evaluate the classification performance of the model and verify the stability of the results, Table 4 reports the mean and standard deviation of Precision, Recall, F1 score, and FPR obtained from five independent runs with different random seeds.

As shown in Table 4, the proposed method achieved the best performance in terms of Precision, Recall, and F1 score, reaching 93.47%, 90.86%, and 91.92%, respectively, with all standard deviations below 0.7, indicating good stability under different random initializations. Meanwhile, the proposed method yielded the lowest FPR (3.87%), demonstrating its effectiveness in suppressing the misclassification of benign traffic as malicious traffic. From the perspective of baseline models, both GCN-LSTM and GAT-LSTM outperformed the standalone GCN and BiLSTM across all metrics, indicating that joint modeling of topological structure and temporal features contributed to improved detection performance. In particular, GAT-LSTM achieved better results than GCN-LSTM due to the introduction of the attention mechanism. However, these two methods did not explicitly consider the alignment of traffic relationship features with temporal inputs or the constraints of lightweight deployment, making it difficult to simultaneously balance detection accuracy and false alarm control in complex scenarios. In contrast, by integrating MFGAT with the improved LSTM in a cooperative manner, the proposed method further enhanced detection accuracy and robustness while maintaining a low FPR.

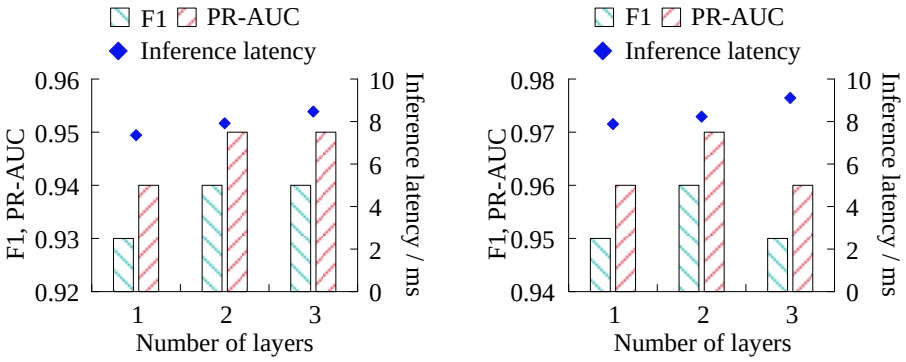
On this basis, an ablation study was conducted to evaluate the contribution of each component in the improved LSTM, and the results are reported in Table 5. To reduce the randomness caused by a single training run, multiple repeated experiments with different random seeds were performed under a fixed data split. The F1 and PR-AUC metrics were reported in the form of mean ± standard deviation, and statistical significance tests were conducted to assess the performance differences between each variant and the base LSTM.

As shown in Table 5, the base LSTM achieved an F1 score of 0.870 ± 0.012 and a PR-AUC of 0.880 ± 0.010. After introducing the attention mechanism, the performance increased to 0.898 ± 0.011 in terms of F1 and 0.907 ± 0.009 in terms of PR-AUC, and the improvement over the base model was statistically significant ($p < 0.05$), indicating that the attention mechanism effectively enhanced the modeling of key anomalous temporal segments. When the residual connection was introduced, the performance reached 0.884 ± 0.013 (F1) and 0.897 ±

0.011 (PR-AUC), and the improvement was also statistically significant ($p < 0.05$), demonstrating that the residual structure helped improve the stability of long-sequence modeling. When only the lightweight gating mechanism was applied, the parameter size was reduced from 5.13M to 3.72M and the inference latency decreased from 7.64 ms to 5.19 ms, while the performance improvement was not significant (F1 = 0.878 ± 0.014, $p < 0.05$), indicating that this mechanism mainly contributed to reducing model complexity and inference overhead rather than directly improving detection accuracy. In the multi-module combinations, the joint use of attention and residual connections improved the performance to 0.909 ± 0.010 in F1 and 0.919 ± 0.009 in PR-AUC ($p < 0.05$). After further incorporating lightweight gating, both the parameter scale and inference latency were reduced while maintaining relatively high detection performance (e.g., F1 = 0.903 ± 0.012, $p < 0.05$). When all three mechanisms were integrated, the model achieved the best overall performance, with F1 and PR-AUC reaching 0.920 ± 0.009 and 0.930 ± 0.008, respectively. Under a parameter size of only 4.23M and an inference latency of 5.61 ms, a favorable balance between performance and efficiency was achieved, and the improvement over the base LSTM was statistically significant ($p < 0.05$). Furthermore, the structural hyperparameter sensitivity of MFGAT was studied and analyzed, and the results are shown in Figure 8.

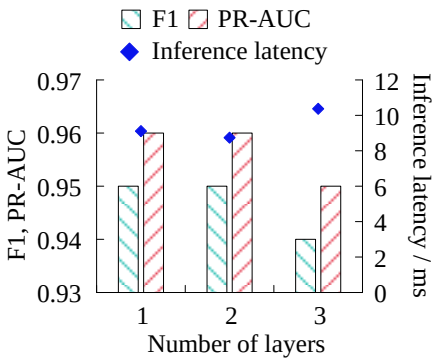
As shown in Figure 8 (a), when the number of attention heads was 2, the F1 and PR-AUC of the model at different layers were basically maintained at 0.93-0.94 and 0.94-0.95, with limited performance improvement. However, the inference latency was relatively low, making it suitable for lightweight scenarios. In Figure 8 (b), when the number of attention heads was 4, the model performance improved, reaching the highest F1 and PR-AUC at L=2, with values of 0.96 and 0.97, respectively. At the same time, the delay was controlled at 8.23ms, maintaining the best balance between performance and efficiency. As shown in Figure 8 (c), when the number of attention heads increased to 8, although F1 and PR-AUC remained at 0.94-0.95 and 0.95-0.96, the latency increases, reaching a maximum of over 10ms, showing a trend of overfitting and efficiency decline. Overall, when the number of attention heads in MFGAT was 4 and the number of layers was 2, the configuration was optimal, ensuring both accuracy and inference efficiency.

Furthermore, the study conducted robustness experiments on complex scenarios, with a total of three scenarios designed: S1 class unbalanced scenario, S2 sudden scenario, and S3 concept drift scenario. Among them, S1 had much more normal traffic than malicious traffic, and the model was prone to miss a few types of attacks. S2 experienced centralized attack traffic in a short period of time, requiring the model to respond quickly. S3's traffic distribution changed over time, and the model needed to maintain stable detection capabilities across time periods. The experimental results of scenario S1 and scenario S2 are shown in Figure 9.



(a) The number of attention heads = 2

(b) The number of attention heads = 4



(c) The number of attention heads = 8

Figure 8: Structural hyperparameter sensitivity analysis of MFGAT

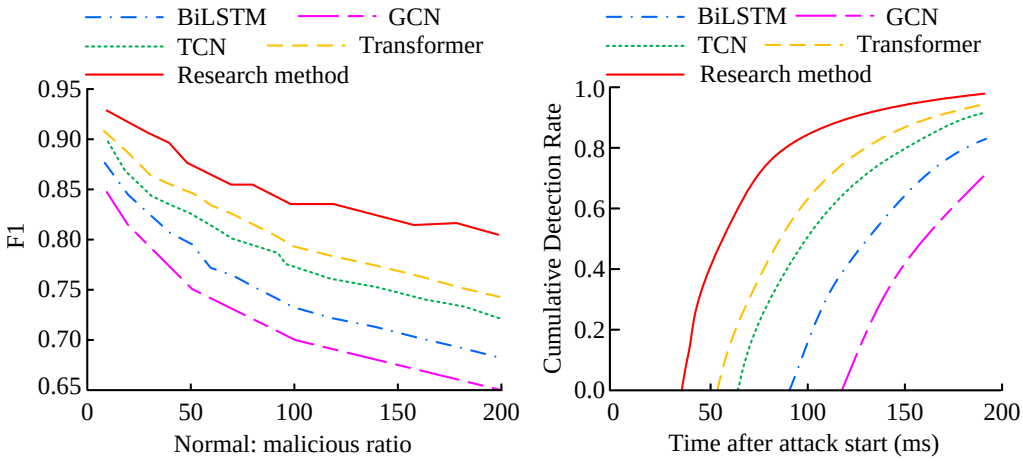


Figure 9: Experimental results of Scene S1 and Scene S2

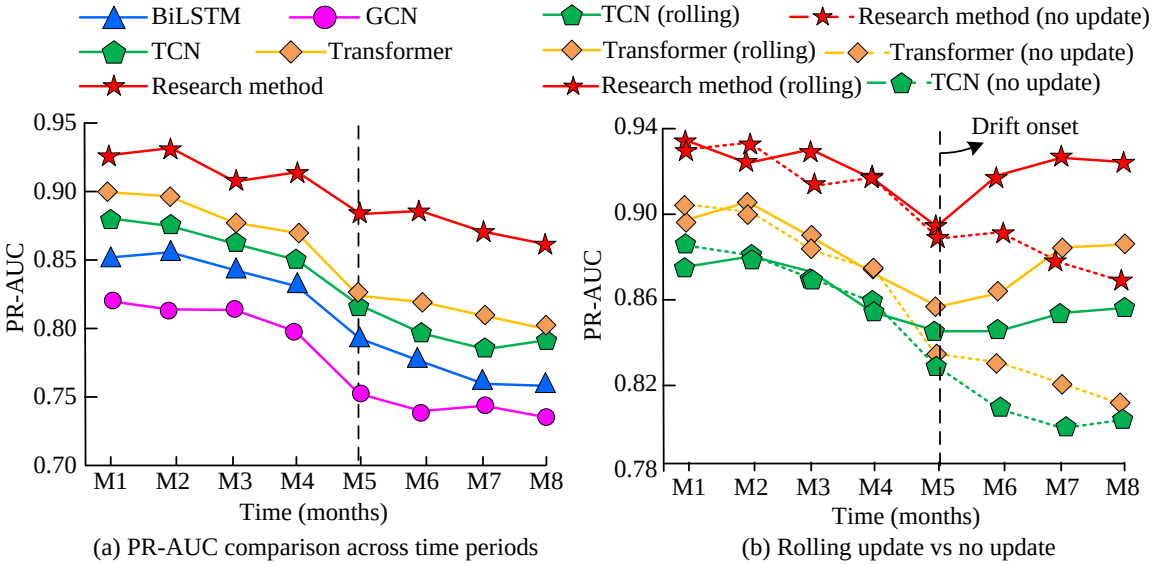


Figure 10: Experimental results of Scene S3

From Figure 9 (a), as the imbalance between normal traffic and malicious traffic intensified, the performance of all methods decreased. However, the method proposed in the study always maintained the highest level, still above 0.80 at 200:1, while Transformer, TCN, and BiLSTM decreased to 0.74, 0.72, and 0.68, respectively, and GCN was the lowest, only 0.65. This indicated that the research method still had stronger robustness under extreme imbalances. As shown in Figure 9 (b), the proposed method could complete the first detection within about 30ms from the beginning of the attack, and the cumulative detection rate quickly increased and approached 100%, which was superior to other comparative methods. Transformer and TCN started detecting at around 50ms and 60ms respectively, with slightly slower overall speed, while BiLSTM and GCN had longer detection delays, with response only after 90ms and 120ms respectively, resulting in higher false alarm rates. The result of scenario S3 is shown in Figure 10.

In the concept drift experiment, M1–M8 denote eight consecutive data stages segmented along the time axis to simulate the distribution evolution of IoT traffic over time. The data in each stage were derived from real traffic collected in different time periods of the original dataset, rather than synthetically generated. As the stages progress, the proportions of different attack types and certain statistical features (such as packet length and inter-arrival time distributions) gradually changed, leading to a shift in the input feature distribution. This setting was intended to reflect concept drift caused by variations in workload and adjustments in attack strategies in real IoT environments. As shown in Figure 10 (a), during the M1-M4 stage, the PR-AUC of each method remained at a high level, with the proposed method stabilizing at 0.92-0.93. However, after the concept drift occurred in M5, the performance of all methods decreased, with GCN showing the most significant decline and ultimately dropping to around 0.75. BiLSTM and TCN also gradually decreased by 0.80.

Transformer was slightly better, but it only stayed around 0.82 in M8. In contrast, the research method showed the smallest decrease, still maintaining in the range of 0.87-0.88, demonstrating stronger resistance to drift. As shown in Figure 10 (b), after using rolling updates, the research method quickly recovered to 0.91-0.92 in M6-M8, which was higher than the non updated curve. Transformer and TCN also had a certain degree of recovery, but the magnitude was smaller than the research method. The results indicated that rolling updates could effectively alleviate the performance degradation caused by concept drift, and the proposed method had the strongest recovery ability under the update mechanism, maintaining long-term stability.

Furthermore, an analysis was conducted from the perspectives of computational complexity and storage overhead. The main cost of the Transformer originated from self-attention operations, whose computational and memory requirements increased rapidly with the sequence length. TCN relied on one-dimensional convolutions for parallel temporal modeling, and its parameter size and computational cost might still be high when deep architectures or multi-channel configurations were adopted. BiLSTM performed bidirectional gating computations, and thus its computational and storage overhead was usually higher than that of unidirectional models. The complexity of GCN was closely related to the number of edges in the graph, and dense graph structures introduced additional computational and memory burdens due to neighborhood aggregation. In contrast, the proposed method reduced redundant parameters in the temporal modeling stage through parameter sharing and low-rank constraints, while maintaining unidirectional modeling to decreased inference latency. On the graph side, local subgraphs were constructed using a sliding window mechanism and attention-based aggregation was employed to highlight key interactions, thereby incorporating topological enhancement while controlling the graph scale. To verify this analysis, the parameter size, FLOPs, peak memory usage, and inference latency of different models were statistically measured, and the results are shown in Table 6.

As shown in Table 6, the proposed method achieved lower parameter size and FLOPs than Transformer, TCN, and BiLSTM, and also required less peak memory. Meanwhile, its inference latency was significantly smaller than that of Transformer and BiLSTM, and it attained higher throughput while preserving topological enhancement (MFGAT). These experimental results were consistent with the theoretical analysis, jointly supporting the conclusion that the proposed method was lightweight and suitable for real-time deployment from both computational cost and deployment efficiency perspectives.

To evaluate the generalization ability of the proposed method under distribution shift, the ToN-IoT dataset, which was more representative of IoT scenarios, was further introduced, and a cross-dataset independent testing experiment was designed. Specifically, a mixed training setting using UNSW-NB15 and CIC-IDS2017 was adopted, and independent testing was performed on the ToN-IoT dataset. The results are reported in Table 7.

As shown in Table 7, under the cross-dataset scenario where the model was trained on the combined UNSW-NB15 and CIC-IDS2017 datasets and tested independently on ToN-IoT, the overall performance of all methods decreased compared with in-domain testing, reflecting the impact of distribution shift on detection models. Nevertheless, the proposed method still achieved the best results in terms of Precision, Recall, and F1 score, and also yielded the lowest FPR. This indicated that the proposed method exhibited better stability and generalization capability under changing feature distributions. In contrast, methods relying solely on temporal or structural modeling suffered more significant performance degradation, whereas models that jointly model topological relationships and temporal features showed relatively more stable performance, further validating the effectiveness of the proposed fusion framework in cross-dataset scenarios.

Table 6: Complexity and real-time comparison with baselines

Model	Params (M)	FLOPs (G/sample)	Peak Memory (MB)	Latency (ms/sample)	Throughput (samples/s)
Transformer	12.8	4.62	512	18.7	53
TCN	6.4	2.35	298	9.6	104
BiLSTM	8.9	2.98	356	13.2	76
GCN	7.5	2.71	402	11.8	85
Research method	4.6	1.21	221	5.4	185

Tbale 7: Cross-dataset generalization results

Method	Precision (%)	Recall (%)	F1-score (%)	FPR (%)
Transformer	82.43 ± 0.91	78.62 ± 0.88	80.47 ± 0.89	7.86 ± 0.74
TCN	79.38 ± 0.97	75.16 ± 0.93	77.12 ± 0.95	9.34 ± 0.81
BiLSTM	74.86 ± 1.06	70.52 ± 1.02	72.61 ± 1.04	11.92 ± 0.93
GCN	70.27 ± 1.13	66.41 ± 1.09	68.23 ± 1.11	13.76 ± 0.98
GCN-LSTM	81.17 ± 0.89	77.48 ± 0.86	79.26 ± 0.87	8.21 ± 0.77
GAT-LSTM	83.56 ± 0.83	79.92 ± 0.79	81.67 ± 0.81	7.24 ± 0.69
Research method	86.74 ± 0.78	82.63 ± 0.74	84.59 ± 0.76	5.68 ± 0.63

5 Discussion

In recent years, several studies have combined graph neural networks with temporal models for malicious traffic detection; however, their modeling targets and fusion strategies differ. Compared with the pure LSTM-based temporal modeling approach in [8], this work explicitly captured the interaction relationships among traffic entities through MFGAT, enabling the model to exploit not only the temporal evolution of individual flows but also the topological correlations among multiple flows. Compared with the method in [9], which integrated Transformer, CNN, and LSTM to address class imbalance and transfer learning, the performance improvement mainly relied on convolutional feature extraction and representation transfer, while structural dependencies among traffic flows were still modeled implicitly. In contrast, this work constructed a traffic interaction graph and introduced feature aggregation, alignment, and serialization mechanisms to achieve a unified mapping from topological features to temporal inputs. Compared with the lightweight or multitask learning-based traffic analysis methods in [10] and [11], which primarily focused on single-flow or independent-sample modeling and provided limited characterization of attack propagation relationships, this work incorporated graph attention-based interaction modeling while maintaining a lightweight design, thereby enabling more effective utilization of topological information.

Overall, unlike existing GAT-LSTM or graph-enhanced temporal models that mainly emphasized the structural stacking of "graph feature extraction + temporal modeling", this work highlighted an interpretable mapping from topological information to temporal modeling and a lightweight design oriented toward real-time deployment in IoT environments. Therefore, the novelty of this work did not lie in a simple combination of GNN and LSTM, but in constructing a fusion framework with consistent semantic mapping tailored to the strong temporal dependence, complex interaction patterns, and resource constraints of IoT malicious traffic, which clearly distinguished it from existing approaches.

Although the proposed method demonstrated good stability in complex attack scenarios, its performance still depended on the quality of traffic relationship graph construction. When the network scale increased significantly or traffic relationships change rapidly, the graph construction and feature alignment processed may introduce additional computational overhead, which could affect real-time performance. Moreover, under extremely imbalanced class distributions or adversarial evasion attack scenarios, the robustness of the model still needs further validation.

6 Conclusion

A recognition method based on the fusion of improved LSTM and MFGAT was proposed to address the high complexity, strong temporal dependence, and resource constraints faced by malicious traffic identification in the IoT environment. The method introduced attention mechanism, residual connection, and lightweight gating into the LSTM structure, which enhanced the modeling ability of key temporal segments while effectively reducing computational complexity. Additionally, MFGAT was used to further capture the topological relationships between traffic entities, achieving the organic combination of structural features and temporal features. The experimental outcomes demonstrated that the proposed method achieved AUC of 0.96 and 0.97 on the UNSW-NB15 and CIC-IDS2017 datasets, respectively, with an average accuracy exceeding 0.91, which was superior to the comparison methods of Transformer, TCN, BiLSTM, and GCN. In complex scenarios such as class imbalance, sudden attacks, and concept drift, this method exhibited stronger robustness and real-time performance. For example, maintaining a detection performance of 0.80 or above under an unbalanced condition of 200:1, with a detection delay of only about 30ms in sudden attack scenarios, and achieving fast recovery through rolling updates in a concept drift environment. Overall, the fusion method proposed in the study not only improved the accuracy and stability of malicious traffic detection, but also balanced lightweight and real-time performance, and had good engineering deployment value.

Although CIC-IDS2017 and UNSW-NB15 have been widely used in intrusion detection research, the traffic contained in these datasets mainly originates from traditional network environments and differs from modern IoT scenarios in terms of lightweight protocols and heterogeneous device communications, which may introduce domain shift and affect the generalization performance of the model in real applications. In addition, these datasets are mostly constructed in experimental or semi-simulated environments, with relatively fixed attack categories, making it difficult to fully cover the increasingly stealthy and low-rate attack behaviors observed in IoT systems, and their data distribution also differs from that of real network traffic. Nevertheless, these datasets provide standardized features and reliable labels, which are helpful for validating the effectiveness of the proposed method under unified evaluation conditions. Future work will further evaluate the proposed approach using IoT-oriented datasets and real traffic traces to improve its practical applicability.

In practical IoT deployments, gateway devices usually have constrained computational and memory resources (e.g., RAM < 1 GB). Although the proposed model incorporates lightweight optimization strategies such as parameter compression and low-rank decomposition to reduce storage and computational overhead, its real-time feasibility on resource-limited gateways still requires further evaluation. In practice, model pruning, quantization, or edge-cloud collaborative deployment can be adopted to balance detection performance and resource consumption.

Moreover, intrusion detection models based on GNN and LSTM may exhibit potential vulnerabilities to evasion attacks, where adversaries deliberately manipulate traffic features or temporal patterns to mimic benign behavior and thereby reduce detection probability. This work does

not yet provide a systematic analysis of such adversarial scenarios. Future research will incorporate adversarial training and robust feature learning mechanisms to enhance the resilience and security of the proposed model against evasion attacks.

References

[1] K. Geetha, and S. H. Brahmananda. Network traffic analysis through deep learning for detection of an army of bots in health IoT network. International Journal of Pervasive Computing and Communications, 19(5):653-665, 2023. https://doi.org/10.1108/IJPCC-10-2021-0259

[2] Zhihui Li, Congyuan Xu, Kun Deng, and Chunyuan Liu. A subspace-based few-shot intrusion detection system for the Internet of Things. Frontiers of Information Technology & Electronic Engineering, 26(6):862-876, 2025. https://doi.org/10.1631/FITEE.2400556

[3] Jean Pierre Ntayagabiri, Youssef Bentaleb, Jeremie Ndikumagenge, and Hind El Makhtoum. OMIC: A bagging-based ensemble learning framework for large-scale IoT intrusion detection. Journal of Future Artificial Intelligence and Technologies, 1(4):401-416, 2025. https://doi.org/10.62411/faith.3048-3719-63

[4] Xiaoheng Deng, Jincai Zhu, Xinjun Pei, Lan Zhang, Zhen Ling, and Kaiping Xue. Flow topology-based graph convolutional network for intrusion detection in label-limited IoT networks. IEEE Transactions on Network and Service Management, 20(1):684-696, 2022. https://doi.org/10.1109/TNSM.2022.3213807

[5] Mohammed Rizvi. Enhancing cybersecurity: The power of artificial intelligence in threat detection and prevention. International Journal of Advanced Engineering Research and Science, 10(5):055-060, 2023. https://doi.org/10.22161/ijaers.105.8

[6] Muhammad Imran, Sangeen Khan, Helmut Hlavacs, Fakhri Alam Khan, and Sajid Anwar. Intrusion detection in networks using cuckoo search optimization. Soft Computing, 26(20):10651-10663, 2022. https://doi.org/10.1007/s00500-022-06798-2

[7] Zeyi Li, Pan Wang, and Zixuan Wang. Flowgananomaly: Flow-based anomaly network intrusion detection with adversarial learning. Chinese Journal of Electronics, 33(1):58-71, 2024. https://doi.org/10.23919/cje.2022.00.173

[8] Somayyeh Fallah, and Amir Jalaly Bidgoly. Android malware detection using network traffic based on sequential deep learning models. Software: Practice and Experience, 52(9):1987-2004, 2022. https://doi.org/10.1002/spe.3112

[9] Farhan Ullah, Shamsher Ullah, Gautam Srivastava, and Jerry Chun-Wei Lin. IDS-INT: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic. Digital Communications and Networks, 10(1):190-204, 2024. https://doi.org/10.1016/j. Dcan.2023.03.008

[10] Dezhi Han, HongXu Zhou, Tien-Hsiung Weng, Zhongdai Wu, Bing Han, Kuan-Ching Li, and Al-Sakib Khan Pathan. LMCA: A lightweight anomaly network traffic detection model integrating adjusted mobilenet and coordinate attention mechanism for IoT. Telecommunication Systems, 84(4):549-564, 2023. https://doi.org/10.1007/s11235-023-01059-5

[11] Sajid Ali, Omar Abusabha; Farman Ali, Muhammad Imran, and Tamer Abuhmed. Effective multitask deep learning for iot malware detection and identification using behavioral traffic analysis. IEEE Transactions on Network and Service Management, 20(2):1199-1209, 2022. https://doi.org/10.1109/TNSM.2022.3200741

[12] T. Anitha, S. Aanjankumar, S. Poonkuntran, and Anand Nayyar. A novel methodology for malicious traffic detection in smart devices using BI-LSTM–CNN-dependent deep learning methodology. Neural Computing and Applications, 35(27):20319-20338, 2023. https://doi.org/10.1007/s00521-023-08818-0

[13] Guolong Shi, Xinyi Shen, Fuke Xiao, and Yigang He. DANTD: A deep abnormal network traffic detection model for security of industrial internet of things using high-order features. IEEE Internet of Things Journal, 10(24):21143-21153, 2023. https://doi.org/10.1109/JIOT.2023.3253777

[14] Zecheng Li, Shengyuan Chen, Hongshu Dai, Dunyuan Xu, Cheng-Kang Chu, and Bin Xiao. Abnormal traffic detection: Traffic feature extraction and DAE-GAN with efficient data augmentation. IEEE Transactions on Reliability, 72(2):498-510, 2022. https://doi.org/10.1109/TR.2022.3204349

[15] Damiano Torre, Anitha Chennamaneni, JaeYun Jo, Gitika Vyas, and Brandon Sabrsula. Toward enhancing privacy preservation of a federated learning CNN intrusion detection system in iot: method and empirical study. ACM Transactions on Software Engineering and Methodology, 34(2):1-48, 2025. https://doi.org/10.1145/3695998

[16] Adel Binbusayyis, Haya Alaskar, Thavavel Vaiyapuri, and M. Dinesh. An investigation and comparison of machine learning approaches for intrusion detection in IoMT network. The Journal of Supercomputing, 78(15):17403-17422, 2022. https://doi.org/10.1007/s11227-022-04568-3.

[17] Nesibe Yalçın, Semih Çakır, and Sibel Ünaldı. Attack detection using artificial intelligence methods for SCADA security. IEEE Internet of Things Journal, 11(24):39550-39559, 2024. https://doi.org/10.1109/JIOT.2024.3447876

[18] Zhong Cao, Zhicai Zhao, Wenli Shang, Shan Ai, and Shen Shen. Using the ToN-IoT dataset to develop a new intrusion detection system for industrial IoT devices. Multimedia Tools and Applications, 2025, 84(16): 16425-16453. https://doi.org/10.1007/s11042-024-19695-7.

[19] Mohanad Sarhan, Siamak Layeghy, and Marius Portmann. Towards a standard feature set for network intrusion detection system datasets. Mobile Networks and Applications, 27(1):357-370, 2022. https://doi.org/10.1007/s11036-021-01843-0

[20] Amit Kumar Mishra, and Shweta Paliwal. Mitigating cyber threats through integration of feature selection

and stacking ensemble learning: the LGBM and random forest intrusion detection perspective. Cluster Computing, 26(4):2339-2350, 2023. https://doi.org/10.1007/s10586-022-03735-8