GAT-CNN Method for Encrypted Malicious Traffic Detection and Classification

Yanan Liu , Suhao Wang * , Zheng Zhang , Tianhao Hou , Pengfei Wang , Shuo Qiu , Lejun Ma

Posted Date: 15 October 2025

doi: 10.20944/preprints202510.1194.v1

Keywords: malicious traffic; encrypted traffic; graph convolutional networks; multi-head attention mechanism; deep learning

 doi:10.20944/preprints202510.1194.v1

Article

GAT-CNN Method for Encrypted Malicious Traffic Detection and Classification

Yanan Liu [1], Suhao Wang [1,*], Zheng Zhang [1,2], Tianhao Hou [1], Pengfei Wang [1], Shuo Qiu [1] and Lejun Ma [1]

[1] School of Network Security, Jinling Institute of Technology, Nanjing 211169, China
[2] School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China
* Correspondence: 2307050012@stu.jit.edu.cn; Tel.:+86-180-0018-0091

Abstract

This paper proposes a Graph Attention based Convolutional Neural Network (GAT-CNN) for encrypted malicious traffic classification and detection, which improves the performance of network encrypted malicious traffic detection, recognition, and classification. By establishing a connectivity diagram between nodes, the spatiotemporal features in network traffic records are strengthened, providing a feature basis for intrusion detection. Using graph convolutional neural networks to dynamically aggregate and extract node behaviors in the graph, generate feature representations with category discrimination ability, and allocate different weights through multi head attention mechanism to achieve accurate classification of network intrusion behavior. To verify the effectiveness of the method, experiments were conducted on the ISCX-VPN encrypted traffic dataset, with a detection accuracy of 98.79% and a recall rate of 99.24%, demonstrating that the proposed method exhibits good performance in detecting malicious traffic.

Keywords: malicious traffic; encrypted traffic; graph convolutional networks; multi-head attention mechanism; deep learning

1. Introduction

The development of the Internet, increased user awareness of security and privacy, and rapid advances in network traffic encryption technologies have made encrypted traffic a critical component of network data flows. Data indicates that over 95% of web pages loaded on the Chrome platform and more than 98% of applications on Android use HTTPs [1]. While SSL/TLS, as the core encryption protocol of HTTPs, ensures data security and privacy, it also introduces challenges for detecting malicious traffic [2]. Attackers employ various techniques to manipulate network traffic, such as leveraging SSL/TLS to encrypt malicious payloads, using traffic obfuscation to alter flow characteristics, and utilizing botnets for DDoS attacks [3]. Furthermore, they bypass interception mechanisms by forging SSL/TLS certificates and identities, among other methods [4]. Malware exploits encryption to circumvent firewalls and launch cyber attacks, and the rapid growth of malicious traffic has become a significant threat in cyberspace. As traffic is encrypted and its features are processed [5], traditional detection techniques based on content inspection and signature matching are no longer adequate in terms of both detection speed and accuracy [6].In response to the problem of malicious network traffic, researchers have begun exploring the application of machine learning for malicious traffic detection [7]. Traditional machine learning models such as Support Vector Machines (SVM) [8], Random Forests [9], and Genetic Algorithms [10] have been proposed and applied to traffic analysis. However, these methods suffer from limitations in generalizability, demonstrating high efficiency only on specific datasets. With the rapid development of neural network technologies, research attention has shifted towards the temporal characteristics of network flows and the logical and spatial relationships within traffic data. Long Short-Term Memory (LSTM) networks [11] and Convolutional Neural Networks (CNN) [12] are now widely used to model the

 doi:10.20944/preprints202510.1194.v1

spatiotemporal sequences of network packets. Although these models excel at extracting local features, they exhibit shortcomings in comprehensively analyzing traffic issues. Within this context, Graph Neural Networks (GNNs) have gradually gained attention due to their unique advantage in representing the spatial relationships within network traffic [13]. Integrating the strengths of different models to create GNN variants [14] has emerged as a promising direction for further research in malicious traffic detection. In summary, most current neural network-based anomaly traffic detection methods focus primarily on traffic feature extraction, often overlooking the topological relationships between network flows. To address the limitations of existing approaches, this paper proposes a GAT-CNN method for encrypted malicious traffic classification and detection. The graph structure, comprising nodes and edges, is more suitable for representing the interactions between communicating hosts and the data flows between them. Existing GNN and Graph Convolutional Network (GCN) methods typically employ Multi-Layer Perceptrons (MLP) to classify the features extracted by the hidden layers of multiple GCN layers, without incorporating additional mechanisms to assist in weight allocation. To enhance the suitability of Graph Convolutional Networks for malicious traffic classification, this paper introduces a multi-head attention mechanism for dynamic weight allocation. Comparative experiments with existing malicious traffic detection models demonstrate that the proposed method achieves superior performance across key metrics, including accuracy and recall, thus fully validating that GAT-CNN provides an effective solution for malicious traffic detection.

To sum up, the contributions of this paper are as follows:

1.Graph structure modeling and spatiotemporal feature enhancement: A Convolutional Neural Network (GAT-CNN) method based on graph attention mechanism is proposed for the classification and detection of encrypted malicious traffic. By constructing a connection graph between nodes, the spatiotemporal features in network traffic records are strengthened, providing a feature basis for intrusion detection.

2.Integration of graph convolution and attention mechanism: Using graph convolutional neural networks to dynamically aggregate and extract node behaviors in the graph, generating feature representations with category discrimination ability, and assigning differentiated weights to different features through multi head attention mechanism to achieve accurate classification of network intrusion behavior.

3.Model validation and performance improvement: Experimental validation was conducted on the ISCX-VPN encrypted traffic dataset, and the proposed method achieved a detection accuracy of 98.79% and a recall rate of 99.24%. Compared with traditional methods, it showed significant advantages in key indicators such as accuracy, recall rate, and F_1 score, verifying its efficiency and feasibility in detecting encrypted malicious traffic.

The main paragraph structure of this article is as follows. Chapter 2 reviews the research results on malicious traffic detection in recent years. Chapter 3 analyzes the related work of encrypted traffic detection and graph based neural networks, and elaborates on the mathematical derivation and implementation details of the proposed GAT-CNN framework, covering three core components: connected graph construction, spatiotemporal feature fusion, and multi head attention weight allocation. Chapter 4 clarifies the experimental environment and evaluation criteria, and introduces the dataset number used in this article. Evaluated the accuracy and precision of the model, analyzed the experimental results, and compared its performance with eight mainstream models. Chapter 5 summarizes this article, emphasizing the advantages of GAT-CNN, identifying current limitations, and outlining future research directions.

2. Research Status

Addressing the challenge of detecting and identifying encrypted malicious network traffic, academia has conducted extensive research, exploring the application of machine learning in malicious traffic detection. Traditional machine learning models such as Support Vector Machines, Random Forests, and Genetic Algorithms have been proposed and applied to traffic analysis. As early

 doi:10.20944/preprints202510.1194.v1

as 2008, Deng He et al. proposed an SVM-based method for P2P traffic classification [15], studying the classification of four types of P2P network traffic in instant messaging (MSN). However, the experimental dataset was solely based on data collected in a specific environment, unable to accurately reflect the types of P2P traffic behaviors that might exist in real network environments. In 2019, Wang Lin et al. proposed a Random Forest algorithm improved by a Genetic Algorithm [16], combining fingerprint recognition with machine learning methods to extract time-related flow features. However, even under optimal conditions, this detection and identification method could only achieve 92% accuracy for one type of traffic, indicating overall low performance.Traditional machine learning algorithms are based on the assumption of training with an infinite number of samples, which data obtained in the intrusion detection field often cannot satisfy. Facing small sample data like new network attacks and zero-day vulnerabilities poses an insurmountable challenge for traditional classification algorithms. Moreover, in such application scenarios, machine learning is mostly used as an auxiliary classifier; its usage is limited, and it can only handle one or a specific type of dataset.With the rapid development of neural network technologies, researchers began focusing on the temporal characteristics of network traffic, and LSTM was introduced into this field to model the temporal sequence of network packets. In 2022, Shi Lin et al. used LSTM for APT attack detection on Linux systems [17], constructing a sandbox based on kernel instrumentation to analyze malicious files and capture malicious behaviors in APT attacks, using sandbox data combined with network attack datasets to build APT attack sets based on the temporal characteristics of attacks. However, the model performed poorly in its initial state and required adjustments for detecting different attacks to finally achieve 92% detection accuracy. Some scholars emphasized the logical and spatial relationships between flows and proposed a CNN for traffic feature extraction. In 2019, Cheng Hua et al. proposed a CNN-based method for identifying encrypted C2C communication traffic [18], identifying server-independent features of malicious software C2C communication and using a multi-window convolutional neural network to extract traffic features for encrypted communication traffic data identification and classification. However, it only distinguished encrypted C2C traffic from web traffic to a certain extent, with an optimal recognition efficiency of 91.07%. To fully consider both temporal and spatial characteristics of traffic, in 2021, Xu Hongping et al. proposed a convolutional recurrent neural network (CRNN) for network traffic anomaly detection technology [19], fully extracting features of network traffic data in both spatial and temporal domains. However, when faced with small sample situations, the fitting capability of the CRNN structure was insufficient, requiring the design of more complex structures.Although CNN models can achieve good detection results in extracting spatial local features of traffic, they require data input of consistent length and will truncate data automatically, unable to fully capture all data content. In this context, GNNs have gradually gained attention due to their unique advantages in representing the spatial relationships of network traffic. In 2024, Luo Guoyu et al. proposed a graph neural network model with random features [20], constructing a graph structure of network communication databases and introducing random features to enrich node features and enhance the expressive ability of the graph neural network. Such a model can achieve optimal detection performance for binary classification tasks, but for multi-classification tasks, it can only achieve average detection performance. With the application of various neural networks in malicious traffic identification, the combination of multiple neural networks, integrating the advantages of different models, has led to variant Graph Neural Networks, providing more powerful analysis tools for malicious traffic detection. In 2022, Zheng et al. proposed a feature extractor based on GCN and a decision tree classifier [21]. However, the model itself combined a neural network GCN with non-neural network decision trees, only showing optimal effects for specific attack types, and the model output had poor interpretability. In 2024, Chen et al. proposed malicious traffic detection integrating graph convolutional networks and a multi-head self-attention mechanism (GCN-MHSA) [22]. However, flow-level feature processing significantly expands quintuple information, resulting in poor overall detection and identification efficiency. The authors used a single-layer attention mechanism, which could not effectively integrate more fine-grained traffic features and identify

 doi:10.20944/preprints202510.1194.v1

correlations between features.Given the current research status, this paper proposes an encrypted malicious traffic classification and detection model based on graph convolutional networks. It uses graph convolutional networks to extract spatial features of traffic, converts them into graph structures for feature processing, introduces a multi-head attention mechanism to assist in processing temporal features, and captures node relationships in complex network traffic. The ISCXVPN2016 dataset is used for model validation to evaluate the model's performance in classifying malicious traffic and compare it with other traffic classification models.

3. GAT-CNN Method

This method uses graph neural networks to extract features from encrypted malicious traffic and constructs a relational graph structure dataset. It utilizes graph convolutional neural networks to extract spatiotemporal features of network traffic, capturing the interaction process between network flows. It combines an attention mechanism to screen key information, improving classification and detection accuracy. The detailed framework of the method is shown in Figure 1, which will be elaborated from the following three aspects.

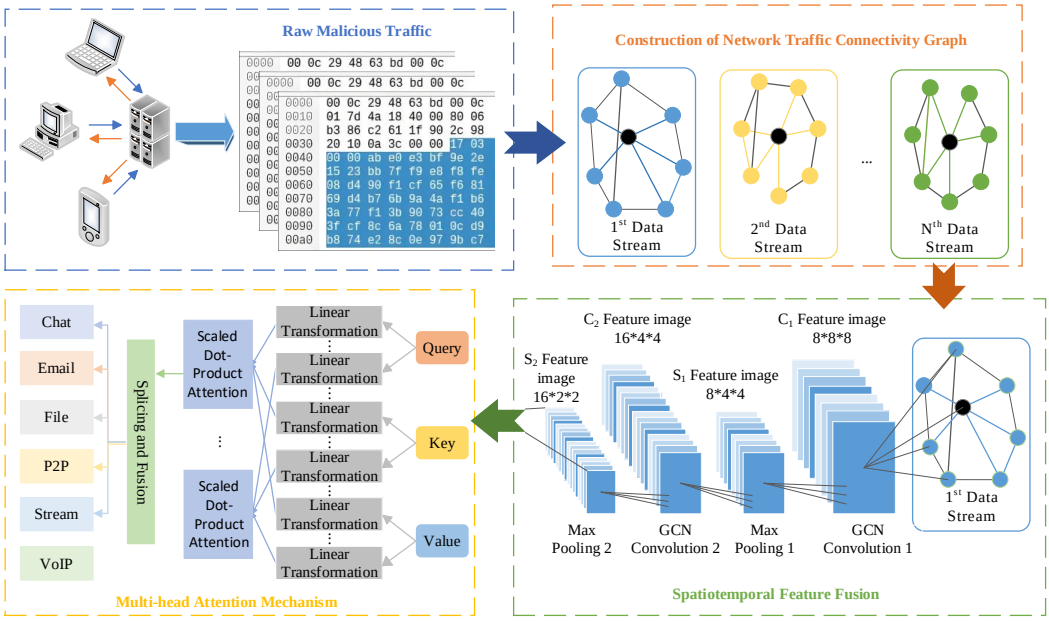


Figure 1. Framework of the GAT-CNN Method.

3.1. Connectivity Relationship Graph Construction

In malicious traffic detection, a fixed number of packets based on the quintuple are selected for grouping. Duplicate identifiers are removed for anonymization to prevent negative impacts on classification results. After completing key feature extraction, network traffic is modeled as a graph structure, where nodes represent IP addresses and host information, and edges represent network flows. The graph structure construction is illustrated in Figure 2.

 doi:10.20944/preprints202510.1194.v1

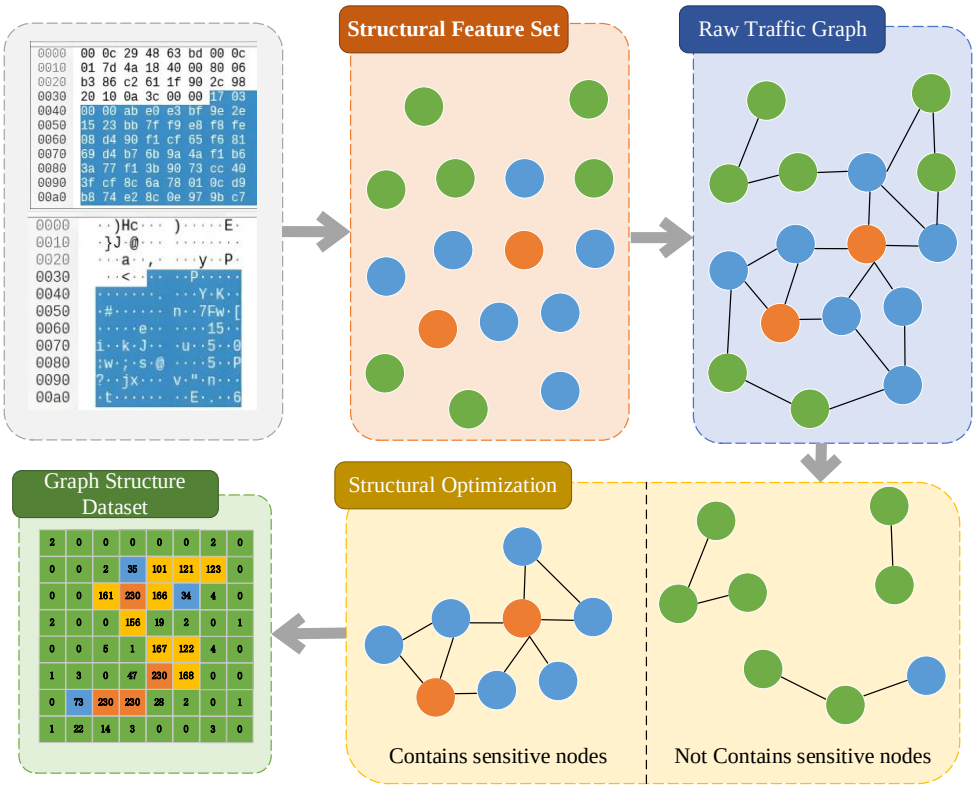


Figure 2. Schematic Diagram of Graph Structure Construction.

By constructing the graph structure, complex relationships and patterns in network traffic can be captured. Ultimately, a graph dataset containing node features, edge information, and labels is generated, providing structured data support for malicious traffic detection. To optimize the graph structure and reduce computational complexity, nodes that do not contain sensitive information are deleted during the feature extraction stage, while ensuring the overall graph structure remains unchanged. The specific implementation algorithm is shown in Algorithm 1:

Algorithm 1: Graph Structure Simplification Algorithm

Input:Graph G (Nodes represent network traffic nodes, edges represent associations between nodes)

Output: The processed and simplified graph G'

```
1. for node in G.nodes():    // Iterate through graph nodes
2.    if is_sensitive(node):    //Check if the node contains sensitive information
3.       add_node_to_graph(G_prime, node)    //If yes
4.    else: neighbors = get_neighbors(G, node)    //If no
5.       for neighbor in neighbors:    //Check neighbor nodes
6.       if not all_non_sensitive:    //At least one
7.          add_node_to_graph(G_prime, node)
8.          remove_edge(G, node, neighbor)    //Remove the edge between nodes
9. return G'    //Return the simplified graph
```

3.2. Spatiotemporal Feature Fusion

To better extract the feature relationships of malicious traffic in the graph dataset, this method introduces two graph convolutional layers. Through feature transformation and refinement, a richer and more precise feature representation is constructed for each node in the graph. Graph pooling operations are used to reduce the dimensionality of the graph, further decrease its scale, perform merging operations for nodes of the same type, and extract global features of the graph. Assume the

 doi:10.20944/preprints202510.1194.v1

input malicious traffic graph node feature matrix is$X^{(0)} \in R^{N*F_0}$, where N represents the number of nodes andF0 is the initial feature dimension. The corresponding adjacency matrix is represented as $A \in R^{N*N}$. The output of the first graph convolutional layer is shown in formula (1):

$$H^{(1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X^{(0)}W^{(0)}\right) \quad (1)$$

where $\tilde{A} = A + I_N$is the adjacency matrix with self-connections added, ensuring that each node can accurately receive its own feature information during the entire traffic detection process. $\tilde{D}$is the degree matrix, $W^{(0)} \in R^{F_0*F_1}$ is a trainable weight matrix for feature information transformation, and$\sigma$is the activation function, increasing the overall nonlinear expressive capability of the model by adjusting it. The output result of the first layer undergoes a max pooling operation to reduce the overall dimension and initially extract important features. The calculation method for the first layer max pooling operation is shown in formula (2):

$$H_{pooled}^{(1)} = MaxPool(H^{(1)}) \quad (2)$$

The max graph pooling operation selects the maximum value in each feature channel to reduce the feature dimension, retain key malicious traffic features, reduce computational complexity, and minimize the risk of model overfitting. The result after the first max graph pooling processing is put into the second graph convolutional layer for feature refinement. $H^{(2)} \in R^{N_1*F_2}$is the output node feature matrix, and F_2 is the new feature dimension. The calculation method for the second-layer feature refinement is shown in formula (3):

$$H^{(2)} = \sigma(\tilde{D}^{(1)-\frac{1}{2}}\tilde{A}^{(1)}\tilde{D}^{(1)-\frac{1}{2}}H_{pooled}^{(1)}W^{(1)}) \quad (3)$$

To reduce the dimensionality of the graph and extract global features, a graph pooling operation is further used. Nodes of the same type are merged to obtain a more concise graph representation while retaining key global feature vectors. The second graph pooling operation method is shown in formula (4):

$$Z = H_{pooled}^{(2)} = MaxPool(H^{(2)}) \quad (1)$$

After two rounds of graph convolution and max graph pooling operations, the output feature matrix Z containing rich and precise features is obtained for input into the classifier for the final network traffic classification processing. The overall process of graph convolution and pooling is shown in Figure 3.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 October 2025 doi:10.20944/preprints202510.1194.v1

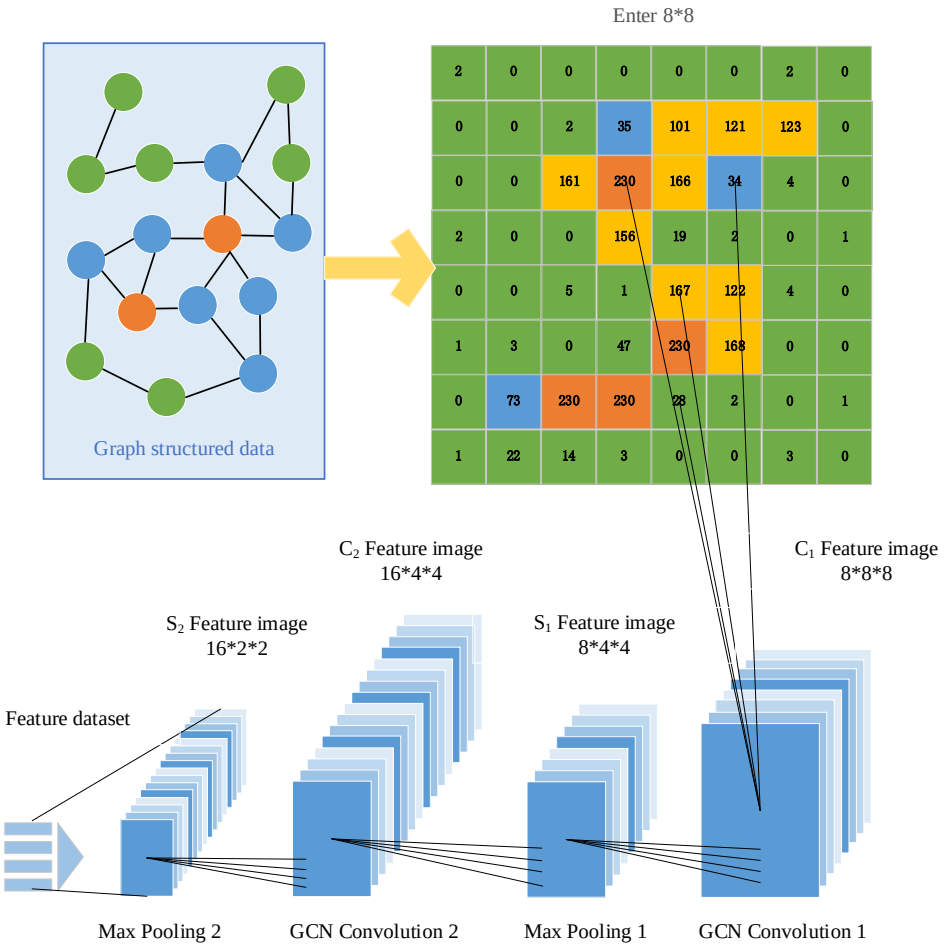


Figure 3. Schematic diagram of the graph convolution and pooling process.

Throughout the training process, this method uses the cross-entropy loss function for calculation, facilitating the binary classification of traffic to determine whether the current traffic type is malicious or normal network traffic. The calculation method for the binary cross-entropy loss function is shown in formula (5):

$$L_{\text{binary}} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \, log \, p_i + (1 - y_i) \, log(1 - p_i)] \tag{5}$$

where N represents the number of samples in this traffic detection training batch, $y_i \in (0,1)$ is the true label of sample i (1 for malicious traffic, 0 for normal traffic), and $p_i$represents the probability that the traffic detection model predicts sample i as malicious traffic. When classifying malicious traffic into specific traffic types, this method chooses to use the multi-class cross-entropy loss function for calculation to determine the probability of specific traffic categories. The calculation method is shown in formula (6):

$$L_{multi} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \, log \, p_{i,c} \tag{6}$$

The final logic of feature extraction is: the constructed connectivity relationship graph dataset is input into the neural network model for training, and the ReLU activation function is used to activate the graph convolutional layers. Imbalanced data sampling is performed on the training set. Training and test data loaders are created. The data is input into the first graph convolutional neural network layer, and the feature values are obtained as training output. After each convolutional layer,

 doi:10.20944/preprints202510.1194.v1

TopKPooling is used for max pooling operation of features for the model, reducing the computational complexity of GCN model training and preventing model overfitting. Global max pooling and global average pooling features are obtained. Finally, the results form a feature dataset, which is ultimately input into the attention mechanism module for weight allocation and classification processing.

3.3. Multi-Head Attention Mechanism Weight Allocation

This method focuses on the analysis of encrypted malicious traffic within network flows. The application of encryption protocols and feature transformation techniques obfuscates the inherent characteristics of the traffic, making distinctive features between network flows less discernible. This obfuscation complicates feature extraction and results in a low correlation between consecutive features. Therefore, effective analysis necessitates a holistic approach that considers the complete context of the network flow. Under these conditions, the complexity or potential incompleteness of the features can significantly compromise classification accuracy. To address this challenge, a multi-head attention mechanism is integrated into the model. This mechanism dynamically allocates weights to different traffic features. Each attention head operates independently to identify the most salient components—those assigned higher weights—within the feature set. This process effectively highlights the most critical nodes and edges in the graph structure, thereby sharpening the model's focus on pivotal features and enhancing overall classification performance.

The input data is linearly transformed to generate corresponding Query (Q) and Key (K) values. The similarity between them is compared to generate corresponding attention scores. Finally, a weighted sum is performed with the assigned Value (V) to obtain the final result. The input data is split into multiple heads. The calculation method for the weighted sum of attention scores for each head is shown in formula (7):

$$O^i = Attention(QW_{Qi}, KW_{Ki}, VW_{Vi}) \tag{7}$$

The calculation process of the model introducing the attention mechanism is shown in Figure 4 and is divided into three stages.

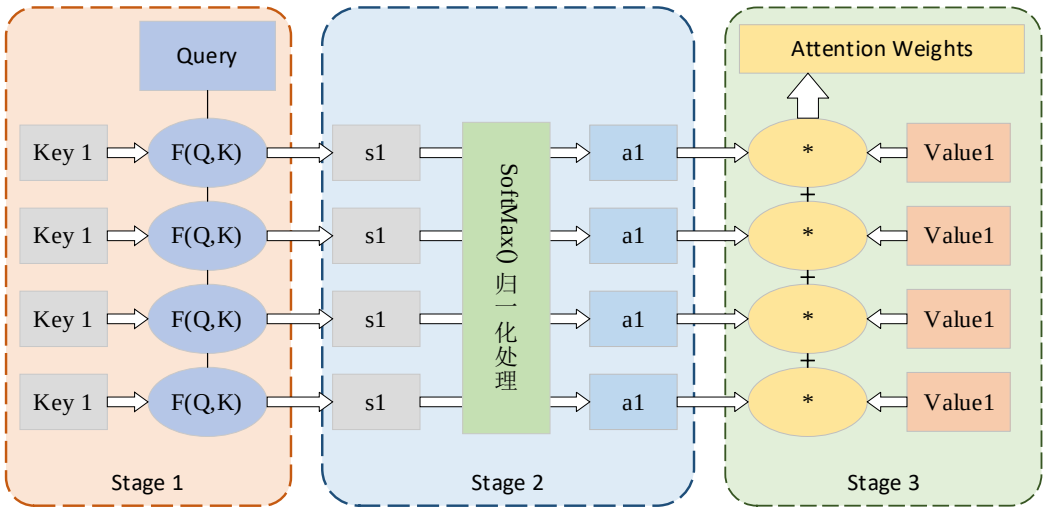


Figure 4. Attention Mechanism Calculation.

Stage 1: The model calculates the similarity between the Query (Q) and multiple Keys (K) through a function *F(Q, K)*, generating corresponding similarity scores *s1*. These correspond to different nodes (such as IP addresses and host information) and the edges between them (network flows) in the network traffic. The query result is Q, the set of keys is$K = \{K_1, K_2, \ldots, K_n\}$. The similarity score s1 calculation method is given by formula (8):

$$s_1 = F(Q,K) = [F(Q,K_1), F(Q,K_2), \ldots, F(Q,K_n)] \tag{8}$$

 doi:10.20944/preprints202510.1194.v1

Stage 2: The similarity score *s1* undergoes SoftMax normalization processing to generate attention weights a_1. These weights represent the importance of each key to the query. They reflect the relative importance of different attacking hosts and network flows in the graph structure, helping to identify potential malicious traffic patterns. The attention weight a_{1i} for the i-th key is calculated as shown in formula (9):

$$a_{1i} = SoftMax(s_{1i}) = \frac{e^{s_{1i}}}{\sum_{j=1}^{n} e^{s_{1j}}} \tag{9}$$

Stage 3: The attention weights a_1 are used to perform a weighted sum of the corresponding Values (V), obtaining the final attention output, which is used for further processing by the malicious traffic detection model. This integrates host features and network traffic characteristics, enhances the model's focus on key traffic features, and improves detection accuracy and efficiency. Assuming the set of Values V is $V = \{V_1, V_2, \ldots, V_n\}$, using attention weights a_1 for weighted summation yields the final attention output O as shown in formula (10):

$$o = \sum_{i=1}^{n} a_{1i} * V_i \tag{10}$$

4. Experimental Analysis

4.1. Experimental Environment and Settings

The experiment was based on the Windows 10 operating system, using the Python 3.8.19 compilation environment. The Scapy library was used for data preprocessing of packets in the ISCXVPN2016 dataset. The Torch_Geometric learning library under the PyTorch framework was selected for building the graph convolutional model. The cross-entropy loss function was used for the loss function, and the Scikit-learn machine learning framework was referenced for comparison.

4.2. Experimental Data Selection

The ISCXVPN 2016 dataset was used for training and research. The dataset was created by reading PCAP files and creating CSV files based on selected features, with a total data volume of 28GB. The ISCX dataset has 14 class labels, but the original traffic was not labeled. However, since Facebook_video can be labeled as "browser" and "streaming", it is not labeled. The final six types of labeled conventional encryption and protocol encapsulation traffic are shown in Table 1.

Table 1. ISCXVPN2016 Dataset Content.

Traffic Classification	Content
Web Browsing	Chrome and Firefox
Email	SMPTS, POP3S and IMAPS
Chat	ICQ, AIM, Skype, Facebook and Hangouts
Streaming	Vimeo and Youtube
File Transfer	Skype, FTPS and SFTP using Filezilla and an external servive
VoIP	Facebook, Skype and Hangouts voice calls (1h duration)
P2P	uTorrent and Transmission (Bittorrent)

A maximum of 1024 data entries were taken from each file in the dataset, with 5% used as the test set and the rest for the training set. The ISCXVPN 2016 dataset contained a total of 85,193 data entries, with 4096 in the test set and 81,097 in the training set. Training parameters were epoch=50, batch_size=4096. A cross-validation method was used, repeating the random splitting process of the dataset 10 times. The results of each experiment were counted, and the average of the 10 experiments was taken as the final result.

 doi:10.20944/preprints202510.1194.v1

4.3. Experimental Evaluation Metrics

The following four key performance indicators were adopted to evaluate the classification effect and performance of the model: Precision, F1-Score, Recall, and Accuracy. The calculation formulas are shown in (11)-(14):

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Rcall = \frac{TP}{TP + FN} \tag{12}$$

$$F_1 - Score = 2 * \frac{Pr\,e\,cision * Re\,c\,all}{Pr\,e\,cision + Re\,c\,all} \tag{13}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

To evaluate model efficacy, the key performance indicators in the above expressions are calculated based on the following definitions: True Positive (TP) represents the number of correctly identified malicious traffic samples; False Positive (FP) represents the number of normal traffic samples incorrectly marked as malicious; True Negative (TN) represents the number of correctly identified normal traffic samples; False Negative (FN) represents the number of unidentified malicious traffic samples.

4.4. Experiment and Analysis

4.4.1. Experimental Results

Experimental results show that the proposed GCN encrypted malicious traffic classification and detection model performs excellently in detecting SSL VPN encrypted traffic. Performance verification was conducted using 10-fold cross-validation, dividing the ISCXVPN dataset into 10 data subsets, using 9 as training model subsets and the remaining one as the validation subset, repeated 10 times, taking the final average performance as the result. The final model's overall recognition accuracy was 98.79%, indicating that the model can effectively extract and analyze malicious traffic from encrypted traffic. During the adjustment process of each parameter setting, the early stopping method was used to monitor the performance of the model's validation set results throughout the iteration process. A dynamic learning rate adjustment strategy was adopted to optimize training effectiveness and prevent overfitting. This model set the initial learning rate to 0.01 during training to ensure rapid convergence in the early stages while avoiding performance degradation and recall rate reduction caused by too high or too low a learning rate. Using this GCN classification detection model, classification detection was performed for six different categories of encrypted traffic: CHAT, EMAIL, FILE, P2P, STREAM, and VOIP. The experimental results are shown in Table 2. The detection precision, recall, and F1 value for the six different categories of encrypted traffic were all above 90%, especially for P2P encrypted traffic, where the metrics reached 100%. The method proposed in this paper can accurately identify different types of encrypted traffic with good generalization ability and discrimination performance.

Table 2. Performance Comparison of Encrypted Traffic Classification for Different Categories.

Traffic Category	Accuracy	Precision	Recall	F_1-score
CHAT	0.984	0.950	0.997	0.973
EMAIL	0.980	0.950	0.996	0.972
FILE	0.996	0.915	0.931	0.923
P2P	1.000	1.000	1.000	1.000

 doi:10.20944/preprints202510.1194.v1

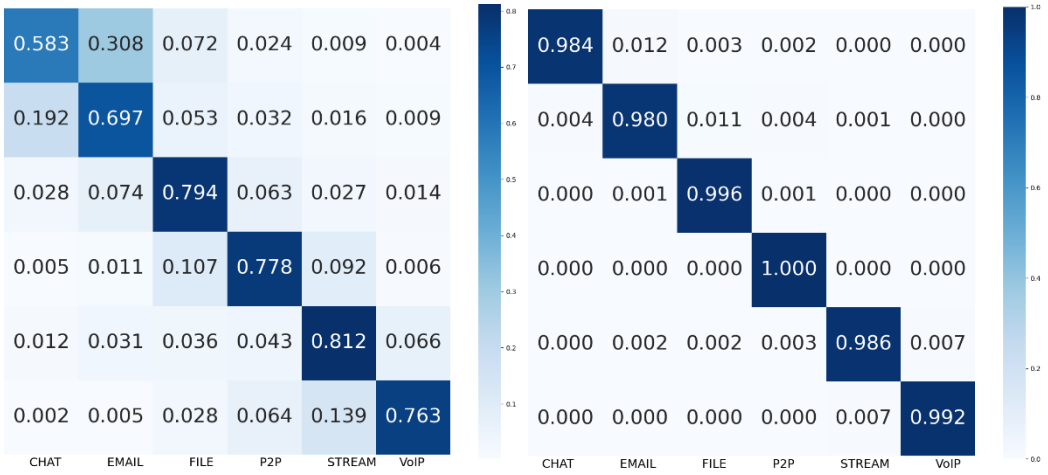
STREAM	0.986	0.921	0.948	0.935
VoIP	0.992	0.938	0.979	0.958
Overall	**0.9879**	**0.9478**	**0.9924**	**0.9696**

Table 3. Performance Comparison of Encrypted Traffic Classification for Different Categories.

Traffic Type		Category (Detection Result / Total Dataset)			Accuracy	Recall
Benign Traffic		67776/68678			98.69%	\
		Correctly Identified Malicious Flows	Misclassified Benign Flows	Unidentified Malicious Flows		
Malicious Traffic	CHAT	6504/6523	341/15425	19/6523	98.36%	99.71%
	EMAIL	7285/7312	386/13122	27/7312	97.98%	99.63%
	FILE	257/276	24/10071	19/276	99.58%	93.12%
	P2P	178/178	0/9849	0/178	100%	100%
	STREAM	422/445	36/3781	23/445	98.58%	94.83%
	VoIP	1744/1781	115/16493	37/1781	99.17%	97.92%
	Overall				98.79%	99.24%

4.4.2. Ablation Experiment

This method differs from traditional graph convolutional GCN neural networks by using an attention mechanism for final weight allocation and a classifier for result classification. To prove the effective improvement in overall detection accuracy after adding the attention mechanism, an ablation experiment was conducted. Figure 5(a) uses a single GCN model, combining hidden layers and output layers for feature extraction and output, without employing the attention mechanism. Figure 5(b) shows the GAT-CNN method used in this paper, which adds a multi-head attention mechanism after GCN spatiotemporal feature extraction for weight allocation. By comparing the confusion matrices of the models before and after adding the multi-head attention mechanism, it can be found that for large sample categories like Chat and Email, the multi-head attention mechanism can effectively reduce misjudgments. For small sample type classifications like P2P and File, the model also exhibits excellent classification performance.



(a) Without multi-head attention mechanism

(b) With multi-head attention mechanism

Figure 5. Ablation study on the use of a multi-head attention mechanism.

4.4.3. Performance Comparison

The proposed GAT-CNN encrypted malicious traffic classification and detection model was compared with the classic SVM model [15], Random Forest model [16], LSTM model [17], CNN model [18], Convolutional Recurrent Neural Network (CRNN) model [19], Graph Neural Network

 doi:10.20944/preprints202510.1194.v1

(GNN) model [20], GNN variant GCN-ETA [21], and GNN variant GCN-MHSA [22]. The detailed comparison is shown in Table 4.

Table 4. Performance Comparison of Malicious Traffic Detection Models.

Model	Accuracy	Precision	Recall	F1-score
Support Vector Machine (SVM)	0.923	0.924	0.867	0.895
Random Forest	0.903	0.909	0.892	0.907
Long Short-Term Memory (LSTM)	0.863	0.884	0.853	0.868
Convolutional Neural Network (CNN)	0.910	0.910	0.905	0.910
Convolutional Recurrent Neural Network (CRNN)	0.970	0.958	0.977	0.967
Graph Neural Network (GNN)	0.934	0.970	0.926	0.948
GCN-ETA	0.974	0.975	0.964	0.969
GCN-MHSA	0.963	0.968	0.971	0.970
GAT-CNN	**0.988**	**0.987**	**0.989**	**0.988**

In terms of detection accuracy, the GAT-CNN model demonstrates significant advantages. Compared to classic SVM and Random Forest models, the GAT-CNN not only covers a broader range of detection categories but also improves overall detection efficiency by 8 percentage points. It achieves performance improvements of 12.5% and 7.8% over traditional LSTM and CNN models, respectively. Furthermore, when compared to the computationally complex hybrid convolutional recurrent neural network (CRNN), the GAT-CNN model maintains a 2.8% performance advantage. Even against standard graph neural network models, which inherently possess advantages in traffic detection, the GAT-CNN achieves a 5.4% improvement in detection accuracy. While existing GCN variants (GCN-ETA and GCN-MHSA) perform well on packet capture datasets, their performance on finely classified encrypted traffic datasets remains inferior to the proposed model. Overall, the GAT-CNN model achieves superior results across three key metrics—precision, recall, and F1-score—demonstrating its effectiveness and excellence in the field of encrypted malicious traffic classification and detection. Although the model achieves over 96% recognition accuracy across different network traffic categories, the accuracy and F1-score for FILE-type traffic remain the lowest, with some FILE traffic being misclassified as VoIP. This limitation primarily stems from the fact that the FILE and VoIP categories have the smallest number of malicious traffic samples in the ISCX-VPN2016 dataset, leading to insufficient feature extraction. Future research will focus on incorporating datasets with more distinct features and larger sample sizes for cross-training and validation to improve overall accuracy and recognition effectiveness.

5. Conclusions

This paper proposes an encrypted malicious traffic classification and detection method that integrates Graph Convolutional Neural Networks with a multi-head attention mechanism. The approach utilizes graph-structured representation for network traffic data, enabling comprehensive extraction of both temporal and spatial characteristics from encrypted traffic flows, thereby achieving efficient detection and classification of encrypted malicious traffic. Compared with conventional methods, the proposed model addresses the limitations of existing detection technologies that exhibit excessive reliance on manual feature selection and rule databases, while demonstrating the following distinctive advantages:Firstly, detection can be completed without requiring traffic decryption, thereby effectively preserving data privacy. Secondly, through graph convolutional neural networks for traffic feature extraction, the obtained features are directly utilized for both model training and detection processes, significantly enhancing detection efficiency. Finally, experimental results

 doi:10.20944/preprints202510.1194.v1

demonstrate that compared to existing identification methods, the proposed model provides superior recognition accuracy and detection efficiency.

Author Contributions: Conceptualization, Liu Yanan and Wang Suhao; methodology, Liu Yanan and Wang Suhao; software, Liu Yanan and Wang Suhao; validation, Liu Yanan and Wang Suhao; formal analysis, Wang Suhao and Hou Tianhao; investigation, Zhang Zheng; resources, Zhang Zheng; data curation, Qiu Shuo and Wang Pengfei; writing—original draft preparation, Wang Suhao; writing—review and editing, Qiu Shuo and Ma Lejun; visualization, Wang Suhao; supervision, Zhang Zheng; project administration, Zhang Zheng; funding acquisition, Zhang Zheng. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by National Natural Science Foundation of China (under Grant 61902163), the Jiangsu "Qing Lan Project", Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Major Research Project: 23KJA520007), and Postgraduate Research & Practice Innovation Program of Jiangsu Province (No.SJCX25_1303).

Data Availability Statement: Where no new data were created.

Acknowledgments: We thank the School of Network Security of Jinling Institute of Technology for the support to this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TLS/SSL	Transport Layer Security/ Secure Sockets Layer
DDos	Distributed denial of service attack
SVM	Support Vector Machines
LSTM	Long Short-Term Memory
CNN	Convolutional Neural Networks
GNN	Graph Neural Networks
MLP	Multi-Layer Perceptrons

References

1. Dang Y.J., Li Q.N. Research on the Development of Foreign Artificial Intelligence Hotspot Security Technology[J]. Information Security And Communications Privacy, 2024(12):1-8, doi:10.3969/j.issn.1009-8054.2024.12.001.
2. Bian Y, Zheng F, Wang Y, et al. AsyncGBP+: Bridging SSL/TLS and Heterogeneous Computing Power With GPU-Based Providers[J]. IEEE Transactions on Computers, 2025(2):74, doi:10.1109/TC.2024.3477987.
3. Aguru A, Erukala S. OTI-IoT: A Blockchain-based Operational Threat Intelligence Framework for Multi-vector DDoS Attacks[J]. ACM Transactions on Internet Technology (TOIT), 2024, 24(3):31, doi:10.1145/3664287.
4. Mosakheil J.H., Yang K. PKChain: Compromise-Tolerant and Verifiable Public Key Management System[J]. IEEE Internet of Things Journal, 2025, 12(3):3130-3144, doi:10.1109/JIOT.2024.3478754.
5. Mei H.T., Cheng G, Zhu Y.L., Zhou Y.Y. Survey on Tor Passive Traffic Analysis[J].Ruan Jian Xue Bao/Journal of Software, 2025,36(01):253-288, doi:10.13328/j.cnki.jos.007182.
6. Hazman C, Guezzaz A, Benkirane S, et al. Enhanced IDS with Deep Learning for IoT-Based Smart Cities Security[J]. Tsinghua Science and Technology, 2024, 29(4):929-947, doi:10.26599/TST.2023.9010033.
7. Yogesh, Goyal M L. Deep learning based network intrusion detection system: a systematic literature review and future scopes[J]. International Journal of Information Security, 2024:1-31, doi:10.1007/s10207-024-00896-y.

 doi:10.20944/preprints202510.1194.v1

8. Surjeet D, Kumar U L, Neetu F, et al. Next-generation cyber attack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree [J]. Journal of Cloud Computing, 2023,12(1):137, doi:10.1186/s13677-023-00517-4.
9. Gou J.J., Li J.H., Chen C., et al. Network Intrusion Detection Method Based on Random Forest[J].Computer Engineering and Applications, 2020, 56(2): 82-88., doi:10.3778/j.issn.1002-8331.1903-0139.
10. Bakır H, Ceviz Ö. Empirical enhancement of intrusion detection systems: a comprehensive approach with genetic algorithm-based hyperparameter tuning and hybrid feature selection[J]. Arabian Journal for Science and Engineering, 2024, 49(9): 13025-13043, doi:10.1007/s13369-024-08949-z.
11. Wang Y. Advanced Network Traffic Prediction Using Deep Learning Techniques: A Comparative Study of SVR, LSTM, GRU, and Bidirectional LSTM Models[C]//ITM Web of Conferences. EDP Sciences, 2025, 70: 03021, doi:10.1051/itmconf/20257003021.
12. PL S, Emmanuel W R S, Rani P A J. Network traffic classification based--masked language regression model using CNN[J]. Concurrency and Computation: Practice and Experience,2024,36(22):e8223-e8223, doi:10.1002/cpe.8223.
13. Altaf T, Wang X, Ni W, et al. GNN-Based Network Traffic Analysis for the Detection of Sequential Attacks in IoT[J]. Electronics, 2024,13(12): 2274-2274, doi:10.3390/electronics13122274.
14. Zhao D., Yin Z.C., Cao Z.H., Lu Z.G. Malicious TLS Traffic Detection Based on Graph Representation[J].Journal of Information Security Research, 2024, 10(03):209-215, doi:10.12379/j.issn.2096-1057.2024.03.03.
15. Deng H., Yang A.M., Liu Y.D. P2P traffic classification method based on SVM[J].Computer Engineering and Applications,2008,(14):122-126, doi:10.3778/j.issn.1002-8331.2008.14.034.
16. Wang L., Feng H.M., Liu B., et al. SSL VPN Encrypted Traffic Identification Based on Hybrid Method[J]. Computer Applications and Software, 2019, 36(02):315-322, doi:10.3969/j.issn.1000-386x.2019.02.055.
17. Shi L., Shi S.S., Wen W.P. Research on APT Attack Detection Based on LSTM in Linux System[J]. Journal of Information Security Research, 2022,8(08):736-750, doi:10.12379/j.issn.2096-1057.2022.08.01.
18. Cheng H., Xie J.X., Chen L.H. CNN-based Encrypted C & C Communication Traffic Identification Method[J]. Computer Engineering, 2019,45(08):31-34+41, doi: 10.19678/j.issn.1000-3428.0051218
19. Xu H.P., Ma Z.W., Yi H., et al. Network Traffic Anomaly Detection Technology Based on Convolutional Recurrent Neural Network[J]. Netinfo Security, 2021, 21(7): 54-62, doi:10.3969/j.issn.1671-1122.2021.07.007.
20. Luo G.Y., Wang X.S, DAI J.Y. Random Feature Graph Neural Network for Intrusion Detection in Internet of Things[J]. Computer Engineering and Applications, 2024,60(21):264-273, doi:10.3778/j.issn.1002-8331.2312-0266.
21. Zheng J, Zeng Z, Feng T. GCN--ETA: High--Efficiency Encrypted Malicious Traffic Detection[J]. Security and Communication Networks, 2022(1): 4274139, doi:10.1155/2022/4274139.
22. Chen J, Xie H, Cai S, et al.GCN-MHSA: A novel malicious traffic detection method based on graph convolutional neural network and multi-head self-attention mechanism[J]. Computers & Security, 2024:147, doi:10.1016/j.cose.2024.104083.