IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 16, 2021, accepted November 9, 2021, date of publication November 15, 2021,
date of current version December 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3128181

A Network Traffic Classification Method Based on Graph Convolution and LSTM

YANG PAN[1], **XIAO ZHANG**[1], **HUI JIANG**[3], **AND CONG LI**[2,3]

[1]China Energy Science and Technology Research Institute Company Ltd., Wuhan Branch, Wuhan 430077, China
[2]Wuhan City College, Wuhan 430083, China
[3]School of Computer Science, Wuhan University, Wuhan 430072, China

Corresponding author: Cong Li (16127097@qq.com)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61772386.

ABSTRACT In the identification of normal and abnormal traffic flows, Convolutional Neural Network (CNN) is commonly used to extract spatial features of network traffic at present. However, its limitation is that the one-dimensional form of traffic flow data needs to be converted into two-dimensional form, without considering the potential spatial correlation between traffic flows. In view of the potential correlation between network traffic flows, this paper proposes a classification method based on graph convolution and Long-Short Term Memory (LSTM). First, perform data preprocessing on the traffic flow data, then use the graph convolutional network to extract the spatial features of spatial topology and use LSTM to extract its temporal features. Finally, the performance of the algorithm is evaluated on the sampled UNSW-NB15 data set. Experimental results show that the proposed method can effectively extract the potential features of network traffic data. Compared with other methods such as feature selection, bidirectional LSTM (BiDLSTM) and CNN-LSTM, it proves the effectiveness of the proposed algorithm and performs better in classification performance.

INDEX TERMS Deep learning, feature extraction, graph convolution, network traffic classification.

I. INTRODUCTION

With the rapid development of Internet, artificial intelligence and big data technology, the network scale and network traffic are also increasing rapidly. However, while the Internet optimizes our lifestyle, the security issues it causes have increasingly become a major threat to national security [1], [2]. Network traffic classification technology can effectively identify normal and abnormal traffic flows in the network environment, so as to reduce the impact of abnormal traffic [3].

Deep learning methods can adaptively extract deep features from network traffic, avoiding a series of complex operations such as feature engineering, and the extracted features are usually more discriminative than feature selection methods. A typical extraction method such as Convolutional Neural Network (CNN) uses filters to extract local features of the network traffic [4], [5]. In this method, the one-dimensional form of the network traffic data needs to be converted into the two-dimensional form. Only the correlation between features is considered, and the correlation between traffic flow data is not considered. In addition to the relationship between the features within a traffic flow, there will also be a certain correlation between the traffic flows, such as the temporal correlation between the current traffic and the past traffic, and the spatial correlation between the traffic with the same source IP or the same destination IP. Therefore, designing a deep learning model with better feature extraction ability has important worthiness for researching. Through the above analysis, this paper proposes a network traffic classification method based on graph convolution and Long-Short Term Memory (LSTM). This method uses the good topology extraction ability of the graph convolution model to extract the spatial features of network traffic data, and uses the LSTM model to extract its temporal features.

The main contributions of this paper are as follows:

(1) We propose a network traffic classification method based on graph convolution and LSTM, which can improve the accuracy of traffic classification, increase the detection rate of abnormal traffic, and reduce the false alarm rate of normal traffic.

(2) In order to evaluate the performance of the proposed network traffic classification model, we not only evaluate the overall metrics of the model, but also calculate

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

the metrics in each class and compare them with feature selection methods and other deep learning models (such as CNN-LSTM, CNN, etc.). The evaluation process uses the sampled UNSW-NB15 as the benchmark data set.

The rest of the paper is organized as follows. Section II reviews the related works in the field. Section III introduces the detailed construction process of the proposed traffic classification model. The experimental comparison and performance evaluation are presented in Section IV. Section V concludes the work and makes an outlook.

II. RELATED WORK

In the field of deep learning, related researchers apply deep learning algorithms to the classification of network traffic, such as applying restricted Boltzmann machines to the classification of DoS traffic [6], using Artificial Neural Network (ANN) to detect the malicious traffic [7], the application of deep belief network in network traffic classification [8] and so on. Since the network traffic data itself also has potential temporal and spatial features, the temporal feature is reflected in the current and past traffic flows, and the spatial feature is reflected in the topological correlation between the traffic flows. Therefore, the spatial and temporal features also have a certain influence on the recognition of normal and abnormal traffic. Relevant researchers have applied CNN to the spatial feature extraction of network traffic, and have achieved certain achievements [9], [10]. Riyaz and Ganapathy [5] proposed a feature selection method based on conditional random fields and linear correlation coefficients to select the most contributing features, and then used the CNN model for further feature extraction to improve the performance of network traffic recognition. Xu *et al.* [11] proposed the LSTMs-AE model, which combines LSTM with the AutoEncoder (AE). The model utilizes LSTM's time series feature extraction ability and AE's feature representation learning ability to improve performance. Azizjon *et al.* [12] used the 1D-CNN model for supervised learning of network traffic temporal features, and through experiments to verify that its performance is better than traditional machine learning models such as random forest and SVM. After preprocessing the original traffic data, Xu [13] used image processing technology to convert traffic data into grayscale images, and then used CNN to convolve the grayscale images of traffic to extract the correlation between features. Ling [14] processed the spatial features of the data by using multiple CNNs with different scale convolution kernels, and combined with LSTM to extract temporal features. Imrana *et al.* [15] proposed the bidirectional LSTM (BidLSTM) model for the classification of abnormal traffic, and verified its performance to be better than LSTM and other models.

Applying LSTM to the extraction of network traffic features can effectively extract the time series features between traffic flows. Although the application of CNN to the extraction of traffic spatial features also has a certain performance improvement, however CNN is more suitable for processing Euclidean structural data such as images. The form of network traffic data is usually a one-dimensional structure, and the spatial relationship between traffic flows is more similar to a topology structure. Graph convolution model [16] has a good feature extraction capability for topological structure and has been widely applied in some fields. Zhao *et al.* [17] proposed a combination of graph convolutional network and Gated Recurrent Unit (GRU) to extract the temporal and spatial features of traffic roads and make more accurate predictions of road traffic flow. The results show that its performance is better than traditional time series regression models such as ARIMA and SVR. Yao *et al.* [18] construct a single text graph for the corpus based on word co-occurrence and document word relationship, and then learn the text graph convolutional network for the corpus. Compared with other methods, the performance of this model is more prominent.

By analyzing the application status and limitations of the above works, the graph convolution model is still in the exploratory phase. In the field of network security, the application of graph convolution model in network traffic feature extraction has important research significance.

In summary, many feature extraction methods have been proposed in recent years, but most of them still have some limitations, such as:

- The CNN model used for network traffic spatial feature extraction mainly considers the relationship between network traffic data features, and does not consider the spatial relationship between traffic flows.
- The experimental results of some methods are mainly evaluated on overall metrics, which need to be further verified from multiple metrics.

Based on the above studies and findings, we propose a network traffic classification method based on graph convolution and LSTM, and evaluate the performance of each class on multiple metrics. By using graph convolution and LSTM to extract temporal and spatial features of network traffic data, we find that the proposed method has a certain degree of improvement in the performance of normal and abnormal traffic compared with other methods.

III. NETWORK TRAFFIC CLASSIFICATION MODEL BASED ON GRAPH CONVOLUTION AND LSTM

A. SGC MODEL

Graph Convolutional Network (GCN) is widely used in learning graph representation [19]–[21], which can extract spatial features of topological structures. SGC (Simple Graph Convolutional) [22] has made some optimizations on the basis of GCN, which removes the complex nonlinear transformation on GCN, and greatly reduces the computational time complexity of the model through pre-calculation. This section mainly introduces the process of SGC on the classification problem. Let an undirected graph be denoted as $G = (V, A)$, where $V \epsilon R^n$ represents the node sets $\{v_1, v_2, \ldots, v_n\}$ of the graph, $A \in R^{n\times n}$ represents the adjacency matrix of G, and this matrix is a symmetric matrix. The element a_{ij} in A indicates whether there is an edge between the nodes v_i and v_j, if it exists, it is 1. Let $D = diag\,(d_1, d_2, \ldots, d_n)\,, d_i = \sum_j a_{ij}$

denotes the degree matrix of the node, and this matrix is a diagonal matrix.

Each node v_i in the graph has a corresponding d-dimensional feature vector $x_i \in R^d$, then the feature matrix $X \in R^{n\times d}$ contains the feature vector of n nodes, and each node in the graph belongs to a specific class. According to the adjacency matrix and the nodes of known class, then the class to which the node of the unknown class belongs can be predicted.

For the k-th graph convolutional layer, let $H^{(k-1)}$ represents the input of the k-th layer, and let $H^{(k)}$ represents the output node representation of the k-th layer. Then we can get $H^{(0)} = X$, where X is the input to the first graph convolutional layer. In the feature propagation process of the k-th layer of SGC, the hidden feature representation $\bar{h}_i^{(k)}$ of node v_i is the average value of its local neighbors. The update rules are as follows:

$$\bar{h}_i^{(k)} = \frac{1}{d_i+1} h_i^{(k-1)} + \sum_{j=1}^{n} \frac{a_{ij}}{\sqrt{(d_i+1)\left(d_j+1\right)}} h_j^{(k-1)} \tag{1}$$

where $h_i^{(k-1)}$ represents the hidden feature representation of node v_i of the $(k-1)$-th layer, d_i is the degree of node v_i. If there is an edge between node v_i and v_j, then a_{ij} is 1, and the feature representation of node v_j will affect node v_i. The coefficients of $h_i^{(k-1)}$ and $h_j^{(k-1)}$ in equation (1) can be expressed by matrix multiplication as follows:

$$S = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} \tag{2}$$

Equation (2) has $\tilde{A} = A + I$, where $I \in R^{n\times n}$ is the identity matrix, $\tilde{D}$ is the degree matrix of matrix $\tilde{A}$, and S represents the normalized matrix after adding self-circulation. Then the operation of equation (1) can be expressed as follows:

$$\begin{aligned} \bar{H}^{(k)} &= SH^{(k-1)} \\ &= \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(k-1)} \end{aligned} \tag{3}$$

where $H^{(k-1)}$ represents the matrix formed by the feature representations of all nodes at the $(k-1)$-th layer. After local smoothing, each layer corresponds to a learnable weight matrix Θ. The original GCN does the following nonlinear transformation to $\bar{H}^{(k)}$ in equation (4):

$$H^{(k)} = \sigma\left(\bar{H}^{(k)} \Theta^{(k)}\right) \tag{4}$$

where σ is the activation function, $\Theta^{(k)}$ is the weight matrix of the k-th layer. In SGC, the non-linear transformation is removed to speed up the calculation, and it becomes a linear transformation as shown below:

$$H^{(k)} = \bar{H}^{(k)} \Theta^{(k)} \tag{5}$$

Therefore, equation (5) can be further transformed, as shown in the following equation:

$$\begin{aligned} H^{(k)} &= \bar{H}^{(k)} \Theta^{(k)} \\ &= SH^{(k-1)} \Theta^{(k)} \\ &= SSH^{(k-2)} \Theta^{(k-1)} \Theta^{(k)} \\ &= S \ldots SSX \Theta^{(1)} \Theta^{(2)} \ldots \Theta^{(k)} \\ &= S^k X \Theta^{(1)} \Theta^{(2)} \ldots \Theta^{(k)} \end{aligned} \tag{6}$$

For the above equation, S is determined by $\tilde{A}$ and $\tilde{D}$. Then S^k can be pre-calculated, which involves the multiplication of multiple sparse matrices and can greatly reduce the time complexity of model training.

B. LSTM MODEL

As a special RNN, LSTM is mainly used to solve the long-term dependency problem of RNN [15], [23]. LSTM avoids the problem of gradient disappearance by complicating the structure of the hidden layer unit. The basic unit of LSTM is shown in Fig.1.

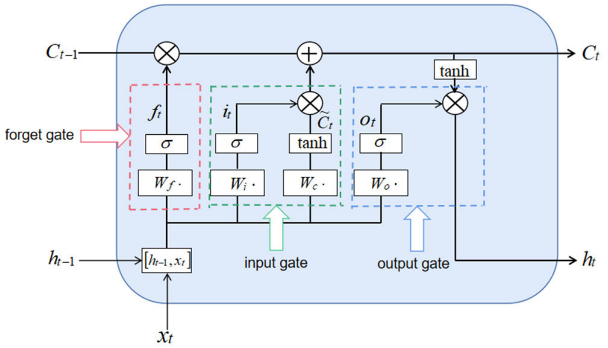


FIGURE 1. LSTM unit structure.

The uniqueness of the LSTM model lies in the three gate control structures, which are the forget gate, input gate and output gate in the figure above. The functions of these three gate structures are as follows:

(1) Forget gate: it is used to control whether the unit is forgotten, that is, the state of the upper hidden unit is forgotten in the current LSTM unit with a certain probability, corresponding to the following equation:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{7}$$

where σ represents the sigmoid activation function, W_f and b_f are the corresponding weight and bias respectively, h_{t-1} is the output of the hidden layer at time $t-1$, and x_t is the input at time t. After processing by sigmoid function, the value of f_t falls into the interval (0, 1), which represents the probability of forgetting the state of the previous hidden unit.

(2) Input gate: responsible for processing the input of the current sequence, corresponding to the following equation:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{8}$$

$$\tilde{C}_t = tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right) \tag{9}$$

where W_i and W_c are weight matrices, and b_i and b_c are bias. The input gate is divided into two parts, which use sigmoid and *tanh* activation functions respectively, and the results of these two parts are multiplied to update the unit state.

(3) Output gate: responsible for outputting the hidden state h_t at time t, as shown in the following equations:

$$o_t = \sigma \left(W_o \cdot [h_{t-1}, x_t] + b_o \right) \tag{10}$$

$$h_t = o_t * \tanh \left(C_t \right) \tag{11}$$

where W_o and b_o are the corresponding weight and bias. It can be seen that o_t is calculated from the previous hidden layer h_{t-1} and the input x_t of this layer through the sigmoid function. C_t is the current (time t) unit state, which is calculated by the following equation:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{12}$$

C. STRUCTURE DESIGN OF NETWORK TRAFFIC CLASSIFICATION MODEL BASED ON SGC AND LSTM

When classifying normal and abnormal traffic flows, there will be spatial features (the topological structure correlation between traffic flows) and temporal features (the correlation between current and past traffic flows). In order to extract potential features of traffic flows, this section proposes a network traffic classification model based on graph convolution and LSTM (SGC-LSTM) to improve the normal and abnormal traffic recognition performance. Fig.2 shows the structure of the proposed SGC-LSTM model, which includes the SGC graph convolutional layer, LSTM layer, fully connected layer and output layer. Firstly, the original data is preprocessed, and the topological graph is constructed according to the correlations between traffic flows. Then input the preprocessed data into the SGC model, extract its spatial feature representation, and input the output of SGC into the LSTM layer to extract temporal feature representation. After the LSTM layer, a fully connected layer and an output layer are added for model training.

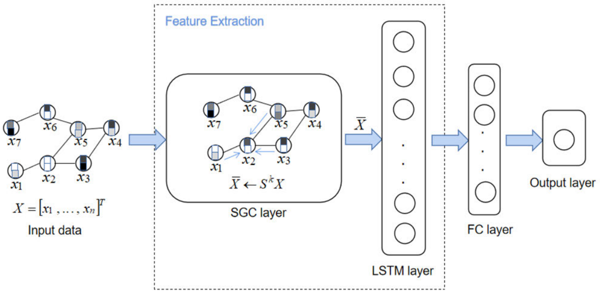


FIGURE 2. SGC-LSTM model structure.

1) INPUT DATA PROCESSING

For numerical features, because different features have different measurement methods, in order to avoid the impact of measurement on the data, it is necessary to standardize the data. Assuming that there are m records in the data set, X_{ij} represents the value of the j-th feature of the i-th record, where $1 \leq i \leq m$, then the features are standardized as follows:

$$\hat{X}_{ij} = \frac{X_{ij} - MEAN_j}{STD_j} \tag{13}$$

where $MEAN_j$ represents the average value of the j-th feature data in the data set, represented by equation (14), and STD_j represents the standard deviation of the j-th feature data, as shown in equation (15):

$$MEAN_j = \frac{1}{m} \sum_{i=1}^{m} X_{ij} \tag{14}$$

$$STD_j = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} \left(X_{ij} - MEAN_j \right)^2} \tag{15}$$

Then the input matrix can be constructed. Assuming that there are n records in the training set, and the feature dimension of each record is d, then the training feature matrix X of the input model can be expressed as follows:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \cdot & \cdot & \cdot & \cdot \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}_{n \times d} \tag{16}$$

After constructing the feature matrix, the next step is to construct the adjacency matrix of the graph, that is, to establish the edge relationship between traffic flows. There are four fields among the features of network traffic flows such as 'srcip', 'dstip', 'sport', and 'dsport', which indicate the source IP address, destination IP address, source port, and destination port respectively. A connection can be established between the traffic flows based on these fields, and the experiments in this paper are based on the following four hypothetical rules for establishing a connection between traffic flows. For traffic flow A and flow B, if they meet one of the following rules, then an undirected edge is established between them.

① A['srcip'] = B['srcip'] and A['sport'] = B['sport']
② A['srcip'] = B['dstip'] and A['sport'] = B['dsport']
③ A['dstip'] = B['srcip'] and A['dsport'] = B['sport']
④ A['dstip'] = B['dstip'] and A['dsport'] = B['dsport']

After the connection is established, the adjacency matrix A can be obtained as follows:

$$A = \begin{bmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & 0 & a_{23} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & 0 \end{bmatrix}_{n \times n} \tag{17}$$

The elements of matrix A satisfy $a_{ij} = a_{ji}, 1 \leq i, j \leq n$, where a_{ij} and a_{ji} represent whether there is a connection between the i-th and the j-th traffic flow. If it exists, then $a_{ij} = a_{ji} = 1$, otherwise, $a_{ij} = a_{ji} = 0$.

2) SGC-LSTM FEATURE EXTRACTION LAYER

The SGC layer is mainly reflected in the local smoothing of nodes and their neighbors. First, the matrix S is constructed. It can be seen from equation (2) that S matrix is related to the

$\tilde{A}$ and $\tilde{D}$ matrices, where $\tilde{A}$ is A plus an identity matrix, and its form is as follows:

$$\tilde{A} = \begin{bmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & 1 & a_{23} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & 1 \end{bmatrix}_{n\times n} \tag{18}$$

The element of $\tilde{A}$ on the diagonal is always 1, which means that each traffic flow node must be connected to itself. Then the form of matrix D is as follows:

$$D = \begin{bmatrix} d_1 & 0 & 0 & \dots & 0 \\ 0 & d_2 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & d_n \end{bmatrix}_{n\times n} \tag{19}$$

The calculation process of $\tilde{D}$ is $\tilde{D} = D + I$, as shown is equation (20):

$$\tilde{D} = \begin{bmatrix} \tilde{d}_1 & 0 & 0 & \dots & 0 \\ 0 & \tilde{d}_2 & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & \tilde{d}_n \end{bmatrix}_{n\times n} \tag{20}$$

where $\tilde{d}_j = d_j + 1, 1 \leq j \leq n$, then the form of S can be multiplied by equation (21), as shown at the bottom of the page.

The output $\bar{X}$ of the $k - th$ layer of SGC is used as the input of LSTM, and the output of the LSTM layer is obtained through a series of operations in the LSTM unit.

3) FULLY CONNECTED LAYER AND TRAINING PROCESS

In summary, the traffic classification algorithm based on the SGC-LSTM model is summarized as Algorithm 1. The input of the fully connected layer is the output of the LSTM layer, and the number of nodes in the fully connected layer is set to 32. This paper mainly discusses whether there is a spatial correlation between abnormal and normal traffic. Therefore, the experiment will be discussed on the binary-classification problem, so the activation function of the output layer can use the sigmoid function.

The SGC-LSTM model is trained by minimizing the cross-entropy loss function. Assuming there are n samples, let z_i represents the score of the i-th sample as a positive example, y_i represents the true class of the i-th sample, and σ represents the sigmoid activation function. Then the form of the cross-entropy loss function is as follows:

$$\begin{aligned} loss\,(z, y) = \frac{1}{n}\sum_{i=1}^{n}[-\,(y_i * log\,(\sigma\,(z_i)) \\ +\,(1 - y_i) * log\,(1 - \sigma\,(z_i)))] \end{aligned} \tag{22}$$

In the training process, RMSProp optimizer is used to optimize the parameters of the model. For the hyperparameters in the SGC-LSTM model, we conducted 100 iterations of training on the data set, and selected the hyperparameters with the highest accuracy as the optimal parameters. The hyperparameters settings of the model are as follows: the number of SGC model graph convolutional layers and LSTM model layers are set to 3, the number of nodes in each layer of LSTM is 32, the value of 'epoches' is set to 500, learning rate is set to 0.01, and 'batch_size' is set to 64. Considering that over-fitting may occur in the training stage, a dropout of 0.1 is used for the LSTM layer and the fully connected layer.

D. MODEL EVALUATION

The KDD99 and NSL-KDD datasets have been used as benchmark datasets in the field of network security, and have made important contributions to the development of network security [24]–[26]. However, a lot of current research has shown that for the current network environment, these data cannot fully reflect network traffic and modern

$$\begin{aligned} S = &\begin{bmatrix} \tilde{d}_1^{\,-\frac{1}{2}} & 0 & 0 & \dots & 0 & \\ 0 & \tilde{d}_2^{\,-\frac{1}{2}} & 0 & \dots & 0 & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\ \cdot \\ 0 & 0 & 0 & \dots & \tilde{d}_n^{\,-\frac{1}{2}} & \end{bmatrix} \\ &\times \begin{bmatrix} 1 & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & 1 & a_{23} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \dots & 1 \end{bmatrix} \begin{bmatrix} \tilde{d}_1^{\,-\frac{1}{2}} & 0 & 0 & \dots & 0 \\ 0 & \tilde{d}_2^{\,-\frac{1}{2}} & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot\ \cdot & \cdot\ \cdot \\ 0 & 0 & 0 & \dots & \tilde{d}_n^{\,-\frac{1}{2}} \end{bmatrix} \\ &= \begin{bmatrix} \tilde{d}_1^{\,-1} & \tilde{d}_1^{\,-\frac{1}{2}} a_{12} \tilde{d}_2^{\,-\frac{1}{2}} & \tilde{d}_1^{\,-\frac{1}{2}} a_{13} \tilde{d}_3^{\,-\frac{1}{2}} & \dots & \tilde{d}_1^{\,-\frac{1}{2}} a_{1n} \tilde{d}_n^{\,-\frac{1}{2}} \\ \tilde{d}_2^{\,-\frac{1}{2}} a_{21} \tilde{d}_1^{\,-\frac{1}{2}} & \tilde{d}_2^{\,-1} & \tilde{d}_2^{\,-\frac{1}{2}} a_{23} \tilde{d}_3^{\,-\frac{1}{2}} & \dots & \tilde{d}_2^{\,-\frac{1}{2}} a_{2n} \tilde{d}_n^{\,-\frac{1}{2}} \\ \cdot\ \cdot\ \cdot & \cdot\ \cdot & \cdot\ \cdot\ \cdot & \cdot\ \cdot & \cdot\ \cdot \\ \tilde{d}_n^{\,-\frac{1}{2}} a_{n1} \tilde{d}_1^{\,-\frac{1}{2}} & \tilde{d}_n^{\,-\frac{1}{2}} a_{n1} \tilde{d}_1^{\,-\frac{1}{2}} & \tilde{d}_2^{\,-\frac{1}{2}} a_{23} \tilde{d}_3^{\,-\frac{1}{2}} & \dots & \tilde{d}_n^{\,-1} \end{bmatrix}_{n\times n} \end{aligned} \tag{21}$$

Algorithm 1 : SGC-LSTM Training

```
Input: Sampled UNSW-NB15 dataset, RMSProp, lr,
        batch_size, dropout
Output: SGC-LSTM Model
load dataset
for data in training and test sets do
    Extract Features(X)
    Extract Labels(Y)
end
scale features with $\hat{X}_{ij} = \frac{X_{ij} - MEAN_j}{STD_j}$
establish matrix A and D based on connection rules
calculate S based on $\tilde{A}$ and $\tilde{D}$, initialize $H = S$
for $i$ from $2 \rightarrow k$ do
    $H = HS$
end
get the output $\bar{X} = HX$
input $\bar{X}$ into the LSTM layer
add a fully connected layer, whose value is 32
add a dropout, whose value is 0.1
get cross-entropy loss by $z_i$ and $y_i$
update parameters by RMSProp with loss
```

low-occupancy attacks. The UNSW-NB15 data set is collected by the Australian Cyber Security Centre, which can better reflect the situation in the network environment. Based on this, the experiment in this paper uses the UNSW-NB15 data set [27]. Due to the large scale of the graph convolution node and the limitation of machine memory, the experiment is carried out on the UNSW-NB15 training set and test set with 20% data in stratified sampling, and the proportion of sample classes in the training set and test set was retained. After sampling 20% of the data set, the number of samples in some attack class is smaller. Therefore, the experiment is mainly verified on the binary-classification problem. The distribution of normal and abnormal traffic flows in the training set and test set after sampling is shown in Table 1.

TABLE 1. Distribution of training and test set sampled by 20%.

	normal	abnormal
training set	11200	23868
test set	7400	9066

1) CONFUSION MATRIX AND METRICS

The confusion matrix describes the number of samples in the data set that are correctly and incorrectly classified by the classifier, and is often used in classification problems [28]. Take the abnormal class as Positive and the normal class as Negative. Then the form of confusion matrix is shown in Table 2.

Accuracy refers to the ratio of samples whose predicted class is consistent with the actual class,

TABLE 2. Confusion matrix.

		predict	
		abnormal class (Positive)	normal class (Negative)
actual	abnormal class (Positive)	TP	FN
	normal class (Negative)	FP	TN

which is expressed as follows:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{23}$$

The precision of abnormal class refers to the ratio of the true abnormal samples to the overall traffic records identified as abnormal, as shown in equation (24) below:

$$precision = \frac{TP}{TP + FP} \tag{24}$$

The recall of abnormal class refers to the ratio of the number of abnormal records correctly classified as abnormal to the overall abnormal samples. It can also be called Detection Rate (DR), as shown in the following equation:

$$DR = recall = \frac{TP}{TP + FN} \tag{25}$$

The $f1 - score$ is a comprehensive metric of precision and recall, expressed as follows:

$$f1 - score = \frac{2 * precision * recall}{precision + recall} \tag{26}$$

The false alarm rate refers to the percentage of normal traffic classified as abnormal traffic, as shown in the following equation:

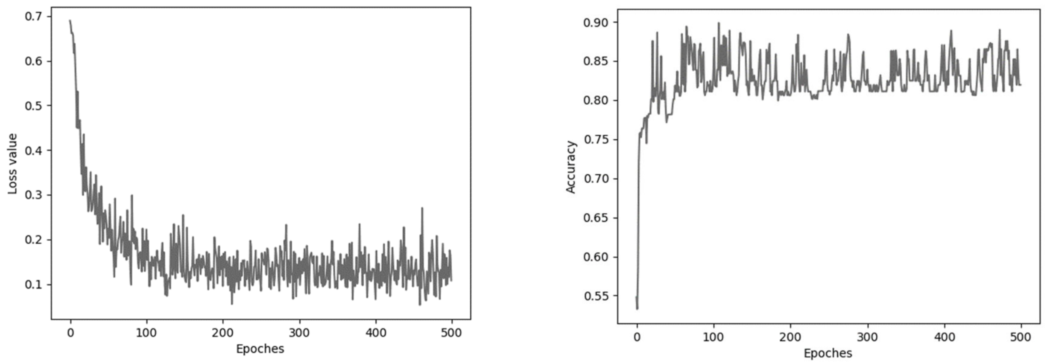
$$FAR = \frac{FP}{FP + TN} \tag{27}$$

2) ROC CURVE AND PR CURVE

Receiver operating characteristic (ROC) curve is a common metric of machine learning classification problems and can be used as a measure of classifier performance. This paper uses AUCROC to represent the area enclosed by the ROC curve and the x-axis. Precision-Recall (PR) curve is also used as a common evaluation metric for classification problems. ROC curve is not sensitive to class distribution, but PR curve can capture the impact of class distribution on model performance [29], [30]. In this paper, $AUCPRC_i$ is used to represent the area of PR curve of the model on class i. If the value of $AUCPRC_i$ is larger, then the model performs better on class i.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

First, the adjacency matrix of traffic flows is established according to the rules in Section III. The topological graph contains 51,534 nodes, which is the total number of training and test set samples. After calculation, 51,534 nodes create about 36.5 million undirected edges, and the number of elements with the value of 1 in the adjacency matrix is



(a) The curve of loss value.

(b) The curve of training set accuracy.

FIGURE 3. SGC-LSTM model training process curves.

about 73 million, which are stored in the form of sparse matrix. During the training process, feature extraction of SGC layer and LSTM layer was carried out on the data of 35,068 nodes in the training set.

A. EXPERIMENTAL RESULTS

The loss value and the accuracy curve of the training set during 500 iterations of the SGC-LSTM model are shown in Fig.3 (a) and Fig.3 (b) respectively. It can be seen from Fig.3 (a) that the loss value of the SGC-LSTM model in the training process slowly decreases after about 50 iterations. After 500 iterations, the loss value fluctuates between 0.1 and 0.2, and the accuracy of the training set fluctuates around 0.85, and then save the SGC-LSTM model. For follow-up comparison experiments, this section inputs the original training and test set data into the saved SGC-LSTM model, extracts the node values of the last layer of SGC and the last layer of LSTM as the new feature representation of the data set, and uses Xgboost as the classification model to evaluate the performance. The confusion matrix of Xgboost model on the new test set extracted by SGC-LSTM is shown in Fig.4, and the metrics are shown in Table 3.

TABLE 3. The metrics of the Xgboost model on new test set extracted by SGC-LSTM.

class	precision	recall	$f1-score$
normal (0)	0.88	0.94	0.91
abnormal (1)	0.95	0.90	0.92
macro avg	0.92	0.92	0.92
weighted avg	0.92	0.92	0.92

As can be seen from Table 3, after feature extraction of SGC-LSTM model, $f1-score$ of normal class and abnormal class in test set are around 0.9, with little difference. The model has higher recall in normal class and higher precision in abnormal class. Fig.5 (a) and Fig.5 (b) are the PR curve of each class and ROC curve of Xgboost model on new test set extracted by SGC-LSTM.

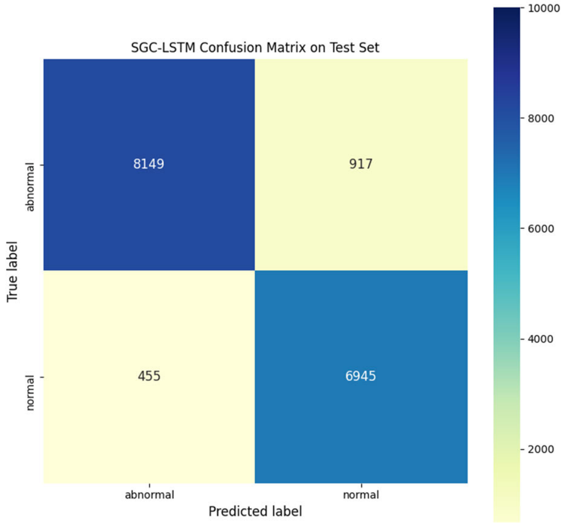


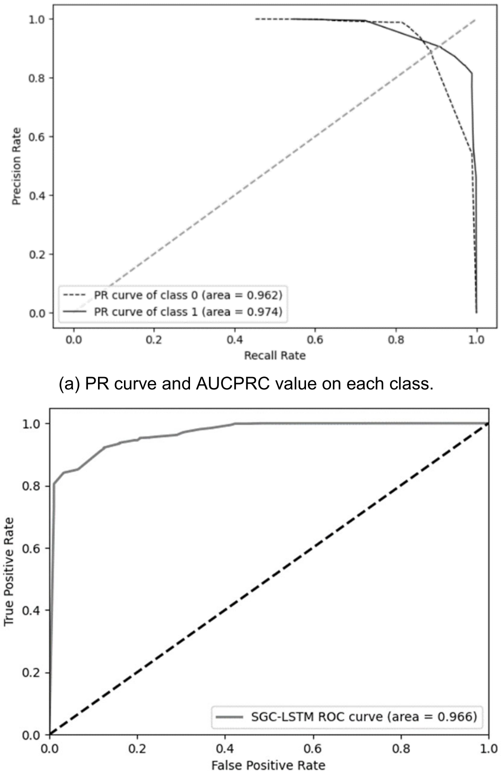
FIGURE 4. Confusion matrix of Xgboost on new test set data extracted by SGC-LSTM.

In Fig.5 (a), the AUCPRC value of the model on all classes are above 0.95, and the model performs better on class 1 (abnormal class). Experimental comparison will be carried out later in this paper, firstly, the extracted features of SGC-LSTM model are compared with the features selected by feature selection method to verify the effectiveness of the proposed method, and then compared with other deep learning methods.

B. COMPARISON TO FEATURE SELECTION METHOD

In order to verify the effectiveness of the SGC-LSTM model in the feature extraction of network traffic data, the experiments in this part mainly compare the performance of SGC-LSTM with feature selection method. The feature selection method in the experiment uses the Sigmoid_PIO algorithm proposed by Hadeel *et al.* [31]. The feature subset selected by this method contains a total of 13 features, and the feature information is shown in Table 4.

The original training set and test set are input into the trained SGC-LSTM model, and the hidden nodes of the SGC layer and LSTM layer are extracted as new feature



(a) PR curve and AUCPRC value on each class.

(b) ROC curve and AUCROC value.

FIGURE 5. Evaluation curves of Xgboost on new test set extracted by SGC-LSTM.

TABLE 4. Feature subset selected by Sigmoid_PIO algorithm.

features	explanations
sbytes	Number of bytes from source to destination.
dbytes	Number of bytes from destination to source.
dttl	Destination to source time to live.
sloss	Source packets retransmitted or dropped.
smeansz	Mean of the flow packet size transmitted by src
res_bdy_len	The content size of the data transferred from the server's http service.
sjit	Source jitter.
djit	Destination jitter.
sintpkt	Source inter-packet arrival time.
ct_flw_http_mthd	No. of flows that has methods such as Get and Post in http service.
is_ftp_login	If the ftp session is accessed by user and password then 1 else 0.
ct_ftp_cmd	No. of flows that has a command in ftp session.
ct_dst_ltm	No. of connections of the same destination address in 100 connections according to the last time.

representations of the new training and test data. The t-SNE method [32] is used to map the new training set and test set features from high-dimensional to two-dimensional planes, and compare them with feature selection method. Fig.6 (a) and Fig.6 (b) are the new training and test set feature visualization results extracted by SGC-LSTM respectively. Fig. 7 (a) and Fig. 7 (b) are the feature subset visualization results of the original training set and test set under the feature selection method respectively. It can be seen from Fig.6 (a) that the normal class (green points) and the abnormal class (red points) overlap less on the two-dimensional plane, while there is a large amount of overlap between two class in Fig.7 (a). Similarly, there is significantly less overlap in Fig.6 (b) than in Fig.7 (b). Based on the four figures, the new training and test set data processed by SGC-LSTM have a more obvious distinction in features.

In order to compare the performance of SGC-LSTM and feature selection method from metrics, the comparison experiment in this section inputs the new features extracted by SGC-LSTM and feature subset constructed by feature selection method into Xgboost model for evaluation. Table 5 shows the comparison between the proposed SGC-LSTM and feature selection method on AUCPRC and AUCROC metrics.

TABLE 5. Comparison of SGC-LSTM and feature selection method on AUCPRC and AUCROC metrics.

methods	$AUCPRC_0$	$AUCPRC_1$	AUCROC
SGC-LSTM	**0.962**	**0.974**	**0.966**
feature selection [31]	0.947	0.957	0.951

When Xgboost is used as the base classifier, the features extracted by SGC-LSTM are superior to the feature selection method in all metrics. Compared with the feature selection method, the SGC-LSTM method has improved about 1.5% in all metrics. The comparison results of accuracy, DR and FAR between SGC-LSTM and feature selection method are shown in Table 6. As can be seen from the table, compared with the feature selection method, the accuracy of the SGC-LSTM model is improved by about 5%. There is little difference between two methods in DR, but SGC-LSTM method is 53% lower than feature selection method in FAR.

TABLE 6. Comparison of SGC-LSTM and feature selection method on accuracy, DR and FAR.

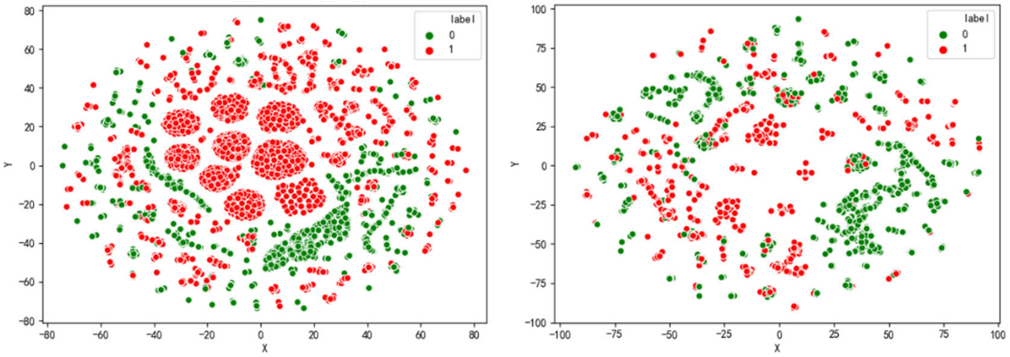
methods	accuracy	DR	FAR
SGC-LSTM	**91.67%**	**89.89%**	**6.15%**
feature selection [31]	86.75%	88.24%	15.09%

C. COMPARISON TO OTHER DEEP LEARNING MODELS

This section compares the proposed method with other commonly used deep learning network traffic feature extraction models such as CNN, BiDLSTM and CNN-LSTM to verify the performance of the proposed method.

The CNN-LSTM model is currently a widely used model in the classification of normal and abnormal traffics. CNN is used to extract spatial features of network traffic flows, and LSTM is used to extract temporal features. The model structure is shown in Fig.8.

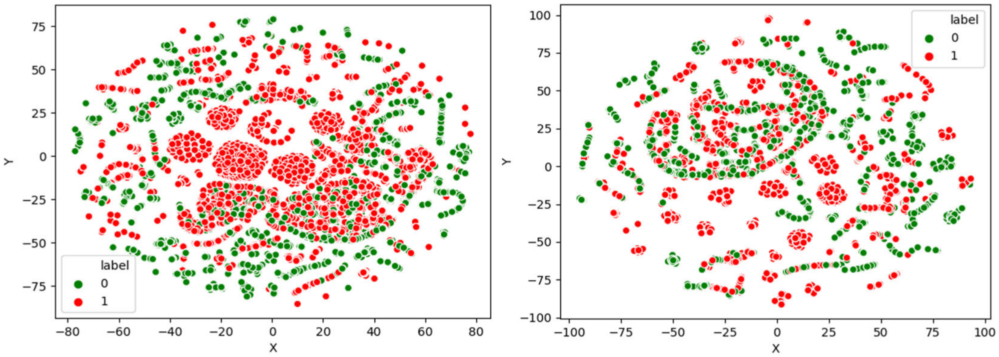
The CNN part of the CNN-LSTM model [33] includes a convolutional layer 1, a pooling layer 2, a convolutional layer 3, a pooling layer 4, and the final fully connected layer. The LSTM part that follows includes two layers. The CNN that is additionally compared in the experiment has the same structure with the CNN-LSTM model. In addition, this paper



(a) Visualization of new training set features.

(b) Visualization of new test set features.

FIGURE 6. Visualization of features extracted by SGC-LSTM under t-SNE.



(a) Visualization of training set features.

(b) Visualization of test set features.

FIGURE 7. Visualization of features selected by feature selection method under t-SNE.

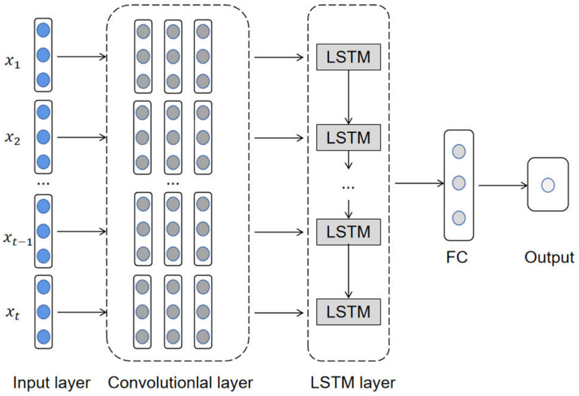


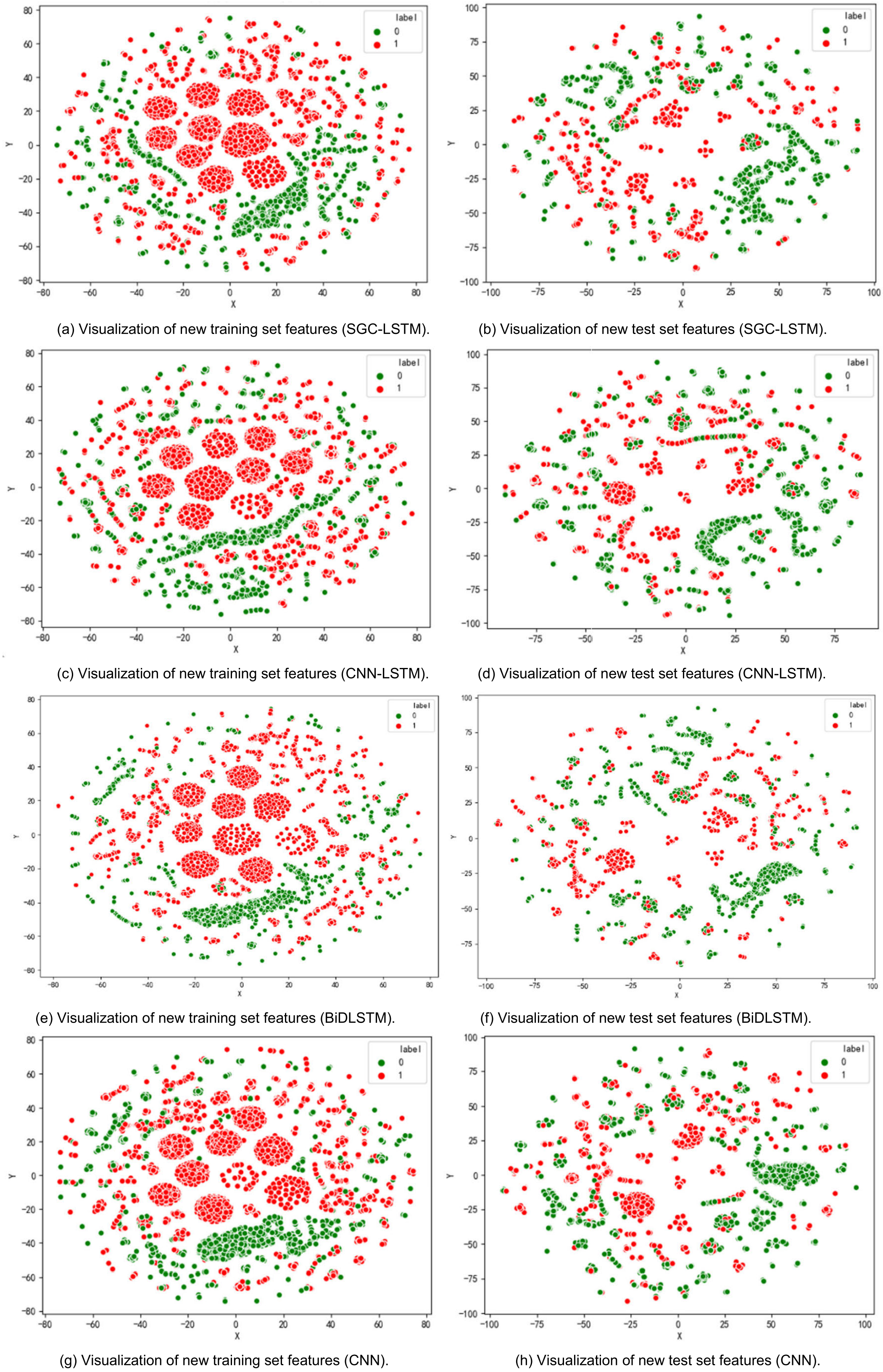
FIGURE 8. The architecture of CNN-LSTM model.

also compares the proposed method with BiDLSTM [15], whose structure contains one embedding layer, two bidirectional LSTM layers, and two fully connected layers.

After extracting the new features of the original training set and test set from each model, the visualization results of the t-SNE method are shown in Fig.9. The performance of the four feature extraction models cannot be seen intuitively from the figure, and it needs to be compared with various classification metrics.

The base classifier still uses Xgboost, the metrics of four feature extraction models on the normal and abnormal class are shown in Fig.10 and Fig.11 respectively.

As can be seen from the above figure, the recall of abnormal class and the precision of normal class of SGC-LSTM are lower than CNN-LSTM model, however it is better than the other three models in other metrics. The CNN-LSTM method is better than CNN and BiDLSTM methods in all metrics. The overall performance of the BIDLSTM model is better than the CNN model with little difference. The comparison of AUCPRC and AUCROC results of each model is shown in Table 7. The CNN model mainly relies on multiple convolution kernels to extract the local spatial features of traffic flows, and the BiDLSTM model mainly relies on the memory unit to extract the temporal features. The performance of these two models is not much different on the three metrics in the table. The CNN-LSTM model has both CNN's spatial feature extraction capabilities and LSTM temporal feature extraction capabilities. Compared with CNN and BiDLSTM, CNN-LSTM has improved by about 0.6% on all metrics. However, the CNN model has certain limitations in non-image structure feature extraction. The SGC-LSTM model performs best on three metrics, and compared to the CNN-LSTM model, its metrics are improved by about 0.4%. Table 8 shows the comparison of accuracy, DR and FAR of



(a) Visualization of new training set features (SGC-LSTM).

(b) Visualization of new test set features (SGC-LSTM).

(c) Visualization of new training set features (CNN-LSTM).

(d) Visualization of new test set features (CNN-LSTM).

(e) Visualization of new training set features (BiDLSTM).

(f) Visualization of new test set features (BiDLSTM).

(g) Visualization of new training set features (CNN).

(h) Visualization of new test set features (CNN).

FIGURE 9. Visualization of the training and test features extracted by each deep learning model under t-SNE.

these four feature extraction models on the test set. Compared with CNN and BiDLSTM, the CNN-LSTM method has better performance. In terms of DR, the CNN-LSTM method is better than the other three models, reaching 90.96%, which is about 1.2% higher than the SGC-LSTM method. However, in terms of accuracy and FAR, the SGC-LSTM method

TABLE 7. Comparison of SGC-LSTM and other deep learning methods on AUCPRC and AUCROC metrics.

models	$AUCPRC_0$	$AUCPRC_1$	AUCROC
CNN	0.950	0.961	0.955
BiDLSTM [15]	0.952	0.962	0.956
CNN-LSTM [33]	0.958	0.970	0.963
SGC-LSTM	**0.962**	**0.974**	**0.966**

TABLE 8. Comparison of SGC-LSTM and other deep learning methods on accuracy, DR and FAR.

models	accuracy	DR	FAR
CNN	88.05%	88.15%	12.08%
BiDLSTM [15]	88.74%	88.50%	10.96%
CNN-LSTM [33]	90.22%	**90.96%**	10.69%
SGC-LSTM	**91.67%**	89.89%	**6.15%**

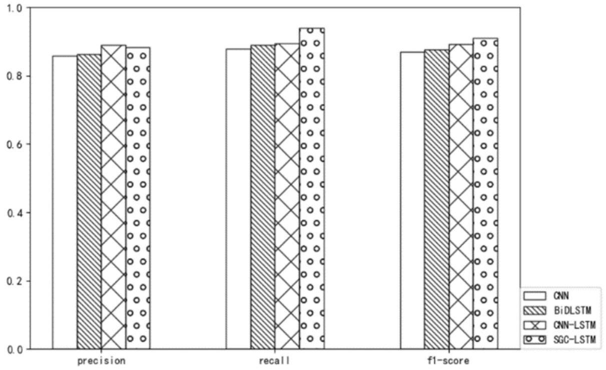


FIGURE 10. Comparison of metrics between SGC-LSTM and other models on normal class.

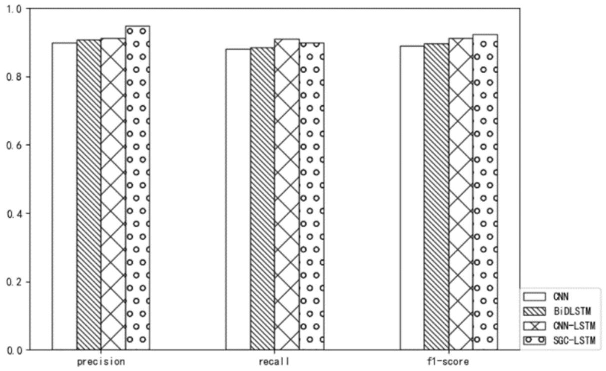


FIGURE 11. Comparison of metrics between SGC-LSTM and other models on abnormal class.

performs best, and its FAR is reduced by about 43% compared to the CNN-LSTM method.

V. CONCLUSION

This paper studies the classification of network traffic, proposes the establishment rules of network traffic topology graph structure, and proposes a network traffic classification method based on graph convolution and LSTM. This method first processes the data with the graph convolution layer, extracts its spatial features, and then combines the LSTM model to extract its potential temporal features. On the sampled UNSW-NB15 data set, it is compared with feature selection and other commonly used deep learning methods (such as CNN, BiDLSTM and CNN-LSTM) to verify the performance and effectiveness of the proposed method. There are also some shortcomings and areas to be optimized in the experiment. When building a topological graph for network traffic data, the more the number of nodes, the more undirected edges are established, and the greater the amount of matrix operations involved, which is a big challenge to the memory size and computing power of the machine. This article provides an idea for using graph convolution model in network traffic environment, exploring the relationship between normal and abnormal traffic flows, and the correlations between traffic flows can be further explored in the future.

REFERENCES

[1] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.

[2] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, p. e4150, 2021.

[3] G. D'Angelo and F. Palmieri, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction," *J. Netw. Comput. Appl.*, vol. 173, Jan. 2021, Art. no. 102890.

[4] J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, p. 916, Jun. 2020.

[5] B. Riyaz and S. Ganapathy, "A deep learning approach for effective intrusion detection in wireless networks using CNN," *Soft Comput.*, vol. 24, no. 22, pp. 17265–17278, Nov. 2020.

[6] S. Seo, S. Park, and J. Kim, "Improvement of network intrusion detection accuracy by using restricted Boltzmann machine," in *Proc. 8th Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Tehri, India, Dec. 2016, pp. 413–417.

[7] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Exp.*, vol. 4, no. 2, pp. 95–99, Jun. 2018.

[8] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114170.

[9] Y. Yang, *Research on Convolutional Neural Network Intrusion Detection Model Based on Network Traffic Feature Map*. Hangzhou China: Hangzhou Dianzi Univ., 2020.

[10] S. Z. Lin, Y. Shi, and Z. Xue, "Character-level intrusion detection based on convolutional neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.

[11] Y. Xu, Y. Tang, and Q. Yang, "Deep learning for IoT intrusion detection based on LSTMs-AE," in *Proc. 2nd Int. Conf. Artif. Intell. Adv. Manuf.*, Oct. 2020, pp. 64–68.

[12] M. Azizjon, A. Jumabek, and W. Kim, "1D CNN based network intrusion detection with normalization on imbalanced data," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 218–224.

[13] Y. Xu, "A research of intrusion detection based on image processing within the framework of deep learning," M.S. thesis, Univ. Electron. Sci. Technol. China, Chengdu, China, 2020.

[14] Y. Ling, *Research on Intrusion Detection System Model Based on Deep Neural Network*. Hangzhou, China: Hangzhou Dianzi Univ., 2020.

[15] Y. Imrana, Y. Xiang, L. Ali, and Z. Abdul-Rauf, "A bidirectional LSTM deep learning approach for intrusion detection," *Expert Syst. Appl.*, vol. 185, Dec. 2021, Art. no. 115524.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[17] L. Zhao, Y. Song, C. Zhang, Y. Liu, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2019.

[18] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 7370–7377.

[19] N. Khan, U. Chaudhuri, B. Banerjee, and S. Chaudhuri, "Graph convolutional network for multi-label VHR remote sensing scene recognition," *Neurocomputing*, vol. 357, pp. 36–46, May 2019.

[20] X. Tian, C. H. Q. Ding, S. Chen, B. Luo, and X. Wang, "Regularization graph convolutional networks with data augmentation," *Neurocomputing*, vol. 436, pp. 92–102, May 2021.

[21] J. Wu, S.-H. Zhong, and Y. Liu, "Dynamic graph convolutional network for multi-video summarization," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107382.

[22] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, Jun. 2019, pp. 6861–6871.

[23] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3469–3477, May 2021.

[24] P. Singh and A. Tiwari, "A review intrusion detection system using KDD'99 dataset," *Int. J. Eng. Res. Technol.*, vol. 3, no. 11, pp. 1103–1108, 2014.

[25] Y. Hamid, V. R. Balasaraswathi, L. Journaux, and M. Sugumaran, "Benchmark datasets for network intrusion detection: A review," *Int. J. Netw. Secur.*, vol. 20, no. 4, pp. 645–654, 2018.

[26] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in *Proc. ACM Workshop Wireless Secur. Mach. Learn.*, 2020, pp. 25–30.

[27] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[28] A. A. Salih and I. Duhok, "Evaluation of classification algorithms for intrusion detection system: A review," *J. Soft Comput. Data Mining*, vol. 2, no. 1, pp. 31–40, Apr. 2021.

[29] H. R. Sofaer, J. A. Hoeting, and C. S. Jarnevich, "The area under the precision-recall curve as a performance metric for rare binary events," *Methods Ecol. Evol.*, vol. 10, no. 4, pp. 565–577, Apr. 2019.

[30] A. S. Qureshi, A. Khan, N. Shamim, and M. H. Durad, "Intrusion detection using deep sparse auto-encoder and self-taught learning," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3135–3147, Apr. 2020.

[31] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Syst. Appl.*, vol. 148, Jun. 2020, Art. no. 113249.

[32] V. D. M. Laurens and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 2605, pp. 2579–2605, 2008.

[33] P. Sun, P. Liu, Q. Li, C. Liu, X. Lu, R. Hao, and J. Chen, "DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system," *Secur. Commun. Netw.*, vol. 2020, pp. 1–11, Aug. 2020.

YANG PAN received the master's degree from Wuhan University, Wuhan, China. His research interests include network security, artificial intelligence, and automation and control.

XIAO ZHANG received the bachelor's degree from the Jilin Institute of Chemical Technology, Jilin, China. His research interests include network security and automatic control of power plant.

HUI JIANG received the master's degree from the School of Computer Science, Wuhan University, Wuhan, China. His current research interests include machine learning and network security.

CONG LI is currently pursuing the Ph.D. degree with the School of Computer Science, Wuhan University. He is an Associate Professor of Wuhan City College. His current research interests include network security, artificial intelligence, and network behavior cognition.