ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

Research on Network Traffic Classification Method Based on Dual-branch Spatial-Temporal Attention

Qian Wang[1,a,*], Tianbo Xu[1,b]

[1]*Wuhu Institute of Technology, Wuhu City, 241003, China*
[a]*2598961564@qq.com,* [b]*101394@whit.edu.cn*
[*]*Corresponding author*

Abstract: *Network traffic classification plays an important role in network monitoring and network management. Deep learning methods have become an effective traffic classification method due to its ability to automatically extract features. In the article, a two-branch spatio-temporal attention-based model is designed to represent bi-directional flows using multiple dimensional uniform packets as samples, 2DCNN with channel and spatial attention to extract spatial features, and LSTM with many-to-one attention mechanism to extract temporal features. The article conducts experiments on the public traffic dataset USTC-TFC2016. The results show that the classification performance of the model is better compared to the ablation experiments.*

Keywords: *traffic classification; flow characterization methods; deep learning; spatial-temporal attention*

1. Introduction

Traffic classification plays an important role in many network applications such as quality of service (QoS), pricing, resource allocation, and malware detection. While the development of new network technologies, such as encryption and port obfuscation, has brought additional challenges to network traffic classification [1]. The development of traffic classification techniques consists of three main phases: (1) port number-based, (2) load-based deep packet inspection (DPI), and (3) machine learning algorithms based on flow statistical feature mining and (4) end-to-end deep learning based algorithms. The dynamic port number and encrypted traffic technologies have been applied to current network data communication on a large scale, making the classification accuracy of port number-based and load-based DPI methods no longer up to the requirements [2][3]. Machine learning algorithms have to explore the statistical characteristics of data flows, which requires specific prior knowledge and additional manual operations. Due to superior classification performance and the fact that they do not require manual feature extraction, in recent years, it has been an increasing interest in classifying network traffic by deep learning method. However, most of these deep learning-based methods do not exploit all the feature information involved in the package headers and loads.

In this paper, firstly, we propose a data flow representation method, which preprocesses the packet according to its headers and data load, filtering out the packets that are not helpful to the classification task and removing the fields in the packet headers that do not contain information about the classification characteristics, and then only retains the valid packet headers and load information of the packets of each data flow. Secondly, for packets of a data flow have time sequence characteristics in packet headers (the field information in packet headers can reflect the packet arrival time order) and the spatial characteristics in data loads (user data of different applications have different characteristics), a dual-branch network FC-STAM (Flow Classifier based Spatial-Temporal Attention Module) is designed. For the packet headers, the LSTM network [4], is used to mine their time-sequence features, while the attention mechanism [5] is introduced to help select time-sequence features. For the packet load, the 2DCNN, is used to mine its spatial feature information, while the spatial attention module CBAM [6] is introduced to helps the CNN to extract and select useful spatial feature information through the joint action of channel attention and spatial attention. Lastly, FC-STAM was applied to the public traffic dataset USTC-TFC2016[7], and the effectiveness of the method was verified in terms of accuracy, precision, recall and model training time.

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

2. Network Model Design and Implementation

2.1 Motivation

Bidirectional data flows are used as the sample of classification, which contain information about the interaction between two sides, having more temporal and spatial characteristics. The variability of the communication patterns of different data flows is mainly reflected in the packet header field and data load. The temporal features of packet header implied the interactive pattern between two sides. Data matrices constructed from data load of different packets have different spatial characteristics.

Based on the above idea, proposed a data flow representation method with both the temporal features of packet headers and the spatial features of loads, and designed a two-branch network based on spatial-temporal attention module to extract the temporal and spatial features of the data flow separately, then concatenate and input these features to the classification module for classification.

2.2 Model Design

The model schematic of the FC-STAM method consists of three main parts: (1) a pre-process module, (2) a feature extraction module based on a spatial-temporal attention mechanism and (3) a classification module.

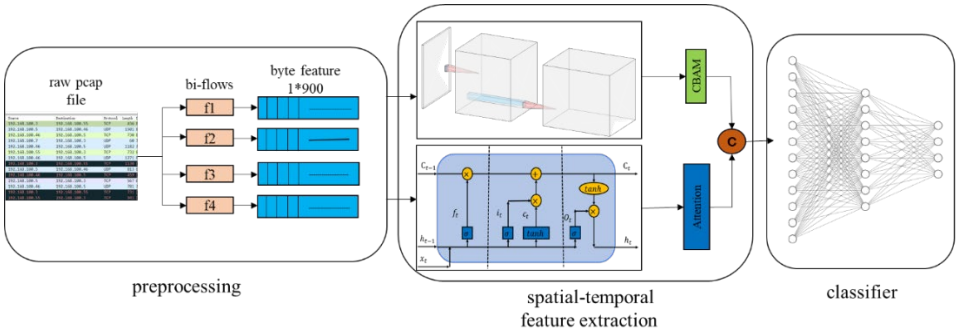


Figure 1: Model schematic of FC-STAM

The main functions of modules in the diagram are as follows.

(1) Pre-processing module: From the original PCAP file, use the Splitcap tool to split the data flow according to a five-tuple (source IP address, destination IP address, source port number, destination port number and protocol), filter out the data flow that without useful information according to specified conditions, and dump the valid data flows into an IDX file as input for the subsequent feature extraction module.

(2) Feature extraction module: The output of the pre-process module is input into feature extraction module from two branches. CNN and CBAM branch networks to extract spatial features; LSTM and Attention branch networks to extract temporal features. Then, the two features are concatenated as the output of the feature extraction module.

(3) Classification module: The output of the feature extraction module is classified using a multilayer fully connected network and softmax.

2.3 Pre-processing Module

The pre-process module consists of five steps: traffic filtering, traffic splitting, packet redefinition, sample generation and IDX file generation, as shown in the figure.

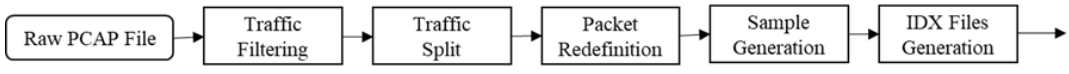


Figure 2: Packet pre-process flow

Traffic filtering: filtering out packets that are not useful for the classification task, such as SYN and FIN packets that establish and disconnect TCP connections, and DNS packets that resolve domain names to IP addresses. These packets are not relevant for application identification or traffic representation and

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

therefore need to be removed [8].

Traffic splitting: Using the Splitcap tool, the original pcap data file is splitted according to the five tuples of the packet header field (source IP address, destination IP address, source port number, destination port number and protocol), and the bidirectional flow is defined as a sample for traffic classification.

Packet redefinition: The original packet structure is shown in Figure 3(a) and (b), with different types of packet headers of different lengths, while some fields in the packet headers are of little value for traffic classification will be removed, and then these packets was redefined into a uniform format in four steps as follows:

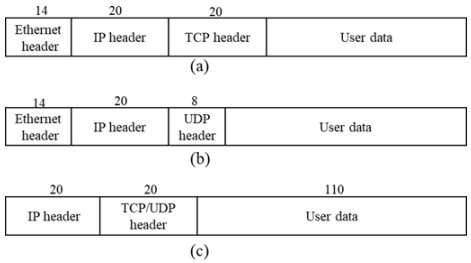
Remove the Ethernet frame header, which contains source MAC, destination MAC, type and FCS fields that provide little value for traffic classification.

Expand UDP header to the length of TCP by padding.

For the data load, the first 110 bytes are retained, which taking into account the number of packets within the flow and the input dimension. The purpose is to ensure that the inputs of the neural network contains the packet headers and load information as more as possible, while making the neural network training time suitable. If the load of some packets is not enough to 110 bytes, zero-padding is used.

As certain classes of traffic in some datasets are generated by specific IP addresses, in order to avoid model over-fitting due to the IP header, IP address in the IP header was set to zero

After the redefinition of the packets, the structure of all packets is unified as shown in Figure 3(c), with each packet being a fixed 150 bytes in length.



(a): Structure of TCP packets; (b): Structure of UDP packets; (c): Redefinition of packets

Figure 3: Packet format before and after redefinition

Each data flow consists of a number of uniform packets as an individual sample, as shown in Figure 4. These packets are fed into the feature extraction module, where the 2DCNN branch extracts the spatial features and the LSTM branch extracts the temporal features.

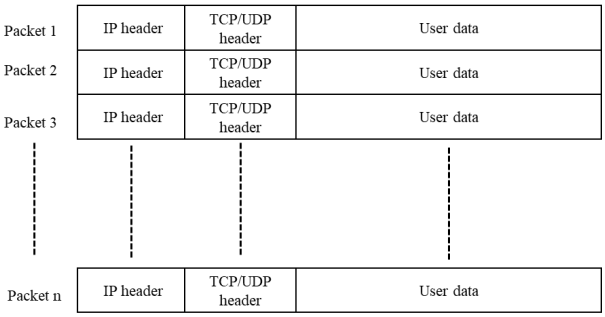


Figure 4: Schematic representation of the data flow

Sample generation: After representation of the data flow, the first 900 bytes of each data flow, i.e. 6 complete data packets, are intercepted as a sample. According consequent experiments, six packets provide enough spatial-temporal features for the data flow.

IDX file generation: The generated samples are output to an IDX format file, which is used as input for the subsequent feature extraction module.

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

2.4 Feature Extraction Module

2.4.1 Dual-branch Network Model Structure

The core of the FC-STAM method is a dual-branch network model based on spatial-temporal attention that shown in Figure 5. The LSTM branch is designed to extract time sequence features of data flow, while the 2DCNN branch is designed to extract spatial features of data flow.

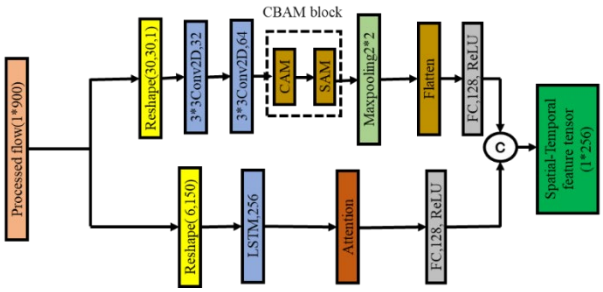


Figure 5: Dual branch network model structure

2.4.2 CBAM Module Implementation

The structure of the CAM is shown in Figure 6, where ⊗ denotes the product of elements and ⊕ denotes the summation of elements. For convolutional neural networks, each channel is a feature detector, so channel attention is concerned with what kind of features are meaningful for traffic classification.

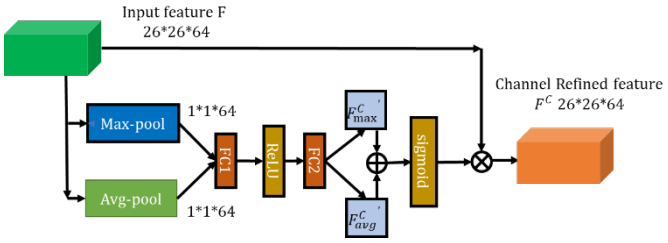


Figure 6: Structure of the CAM mechanism

The Feature map F ∈ $\mathbb{R}^{26*26*64}$ is output of the second convolutional layer of the 2DCNN branch in Figure 6. Channel refined feature FC and channel attention map M_C(F) ∈ $\mathbb{R}^{1*1*64}$ are calculated as follow:

$$F^C = M_c(F) \otimes F \tag{1}$$

Where $M_c(F)$is:

$$F^C_{pool} = W_1\delta\left(W_0\left(F^C_{avg}\right)\right) + W_1\delta\left(W_0\left(F^C_{max}\right)\right) \tag{2}$$

$$M_c(F) = \sigma\left(F^C_{pool}\right) \tag{3}$$

Firstly, obtain the average-pooling and the max-pooling feature map for all channels along the spatial axis, respectively, $F^C_{avg} \in \mathbb{R}^{1*1*64}$ and $F^C_{max} \in \mathbb{R}^{1*1*64}$.

Next, they are fed into a parameter-sharing two fully connected layers. The number of neurons in the first layer is 64/r, where r is the attenuation coefficient, set as 16, so W0 is weight matrix with shape (64,4). ReLU is used as activation function δ. The number of neurons of second FC layer is 64, so W_1 is weight matrix with shape (4, 64).

Next, these two processed feature maps are element-wise added to obtain the intermediate feature map $F^C_{pool} \in \mathbb{R}^{1*1*64}$.

Then, weight coefficients of all channels, $M_C(F)$, is calculated by normalizing F^C_{pool} through Sigmoid activation function.

Finally, the channel refined feature, $F^C \in \mathbb{R}^{26*26*64}$, is calculated by multiplying $M_C(F)$ with original

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

feature F.

The structure of the spatial attention module SAM is shown in Figure 7.

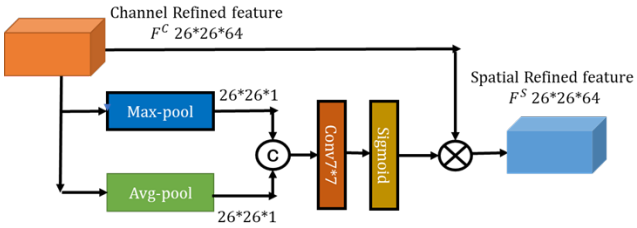


Figure 7: Structure of the SAM mechanism

In the figure, ⊗ denotes the product of elements, and © denotes the concatenation of vectors. Spatial refined feature FS and spatial attention map $M_S(\mathrm{F}) \in \mathbb{R}^{26*26*1}$ are calculated as follow:

$$F^S = M_S(F) \otimes F^C \tag{4}$$

Where $M_S(F)$ is:

$$F^S_{pool} = \left[F^S_{avg}; F^S_{max}\right] \tag{5}$$

$$M_S(F) = \sigma\left(f^{7\times7}\left(F^S_{pool}\right)\right) \tag{6}$$

σ represents the sigmoid operation, and $f^{7\times7}$ is a two dimensional convolution operations with convolution kernel 7.

Firstly, obtain the average-pooling and the max-pooling feature map for all pixel points along the channel axis, respectively, $F^S_{avg} \in \mathbb{R}^{26*26*1}$ and $F^S_{max} \in \mathbb{R}^{26*26*1}$.

Next, these two feature map are concatenated together to obtain the intermediate feature map $F^S_{pool} \in \mathbb{R}^{26*26*2}$.

Then, weight coefficients of all pixels, $M_S(F)$, is calculated by $f^{7\times7}$ operation and Sigmoid activation function.

Finally, the spatial refined feature, $F^S \in \mathbb{R}^{26*26*64}$, is calculated by multiplying $M_S(F)$ with channel refined feature F^C.

2.4.3 Temporal-attention Module Implementation

To extract the temporal features contained in the packet headers more effectively, LSTM and many-to-one attention mechanism is used in the LSTM branch. Functional diagram of many-to-one attention processing multiple packets within a data flow is shown in Figure 8.

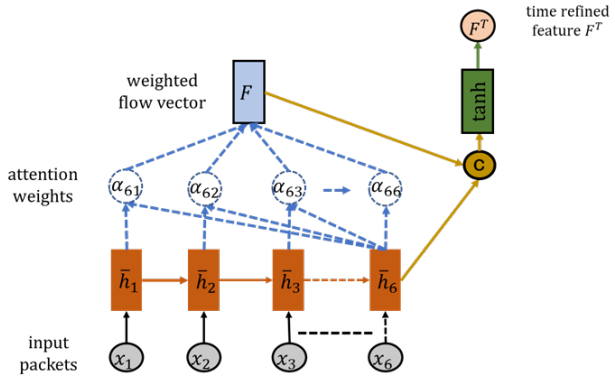


Figure 8: Functional diagram of many-to-one attention

An input sample of LSTM branch is 6*150 bytes for each data flow, including six packets headers and packets loads. By Reshaping (6, 150), $[x_1 \sim x_6]$ is inputted to LSTM unit at 6 time-steps. $\bar{h}_1 \sim \bar{h}_6 \in \mathbb{R}^{1*256}$, is the hidden layer state of LSTM unit at every time-step. Time refined feature F^T is calculated as following.

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

$$score(\bar{h}_6, \bar{h}_i) = \bar{h}_6 W \bar{h}_i^T \tag{7}$$

Equation 7 calculate the correlated score between *i*th hidden state and the last hidden state, where **W** is a trainable matrix.

$$\alpha_{6i} = \frac{exp(score(\bar{h}_6, \bar{h}_i))}{\sum_{i=1}^{6} exp(score(\bar{h}_6, \bar{h}_i))} \tag{8}$$

Equation 8 calculate the attention weight of time-steps by normalizing correlated score of 6 hidden state.

$$\mathrm{F} = \sum_6 \alpha_{6i} \bar{h}_i) \tag{9}$$

By summing up the multiplication of attention weight and hidden layer state, equation 9 calculate the weighted flow vector $\mathbf{F} \in \mathbb{R}^{1*256}$.

$$F^T = f(F, \bar{h}_6) = tanh(W_c[F; \bar{h}_6]) \tag{10}$$

Concatenating weight flow vector **F** and last hidden state, trained by one fully connected layer with *tanh* activation function. The output is the time refined feature $F^T \in \mathbb{R}^{1*256}$.

Many-to-one attention calculate weight for hidden state of every time-steps, helping LSTM to extract temporal feature effectively.

3. Experimentation and Discussion

3.1 Datasets

In this paper, we use USTC-TFC2016 public dataset for verifying performance of FC-STAM. The USTC-TFC2016 dataset includes 10 types of benign traffic and 10 types of malware traffic. The distribution of USTC-TFC2016 is shown in Table 1.

Table 1: Distribution of USTC-TFC2016 dataset

Category	Malware Traffic	Size(MB)	Category	Benign Traffic	Size(MB)
-	- Tinba	2.55	-	Facetime	2.4
0	Cridex	94.7	9	BitTorrent	7.33
1	Geodo	28.8	10	FTP	60.2
2	Htbot	83.6	11	Gmail	9.05
3	Miuref	16.3	12	MySQL	22.3
4	Neris	90.1	13	Outlook	11.1
5	Nsisay	281	14	Skype	4.22
6	Shifu	57.9	15	SMB	1206
7	Virut	109	16	Weibo	1618
8	Zeus	13.4	17	WorldOfWarcraft	14.9

By using the data flow representation method proposed in Section 3.3, the datasets were preprocessed separately to generate samples, and the training and testing sets were divided in a 9:1 ratio, and finally, the IDX files for training and testing were generated. Among them, the USTC-TFC2016 dataset has only 2 samples of Tinba class after preprocessing, which is not enough to support the training of the model. Therefore, we remove the Tinba class and accordingly remove the Facetime class which has the least number of samples in the benign traffic. After preprocessing, USTC-TFC2016 dataset consists of 9 benign classes and 9 malware classes.

3.2 Evaluation metrics

Based on the confusion matrix, the following metrics are calculated to evaluate the model's

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

classification effectiveness.

$$\text{precision} = \frac{TP}{TP+FP} \tag{11}$$

$$\text{recall} = \frac{TP}{TP+FN} \tag{12}$$

$$\text{F1} - \text{score} = \frac{2*\text{precision}*\text{recall}}{\text{precision}+\text{recall}} \tag{13}$$

Where TPs (True Positives) are the flows correctly classified, FPs (False Positives) are the flows incorrectly classified, FNs (False Negatives) are the flows incorrectly rejected, and TNs (True Negatives) are the flows correctly rejected.

3.3 Experimental results

3.3.1 Feature transformation and extraction results

To provide a graphical interpretation of how the intermediate layers in the network affect the features transformation, t-Distributed Stochastic Neighbor Embedding (t-SNE) [9] was used to visualize the high-dimensional data space in a two-dimensional plane. In this section, the USTC-TFC2016 dataset with a larger number of traffic categories is selected for subsequent dimensionality reduction and visualization experiments.

A) Feature map after pre-processing module

Figure 9 presents the features of data flow obtained through pre-process module in t-SNE way. The number in the figure is located center of the category.

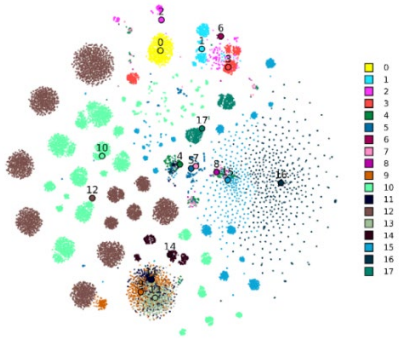


Figure 9: T-SNE representation of features after data preprocessed

As show in Figure 9, many categories overlap each other, e.g. 1,2,3 and 6, these four categories; 4,5,7,8 and 15, these five categories; 9,11 and 13, these three categories; 15 and 16, these two categories. Categories 0,10,12,14,17 do not overlap with other categories, but categories 10 and 12 are widely scattered.

Based on the above analysis, the byte feature extracted from the data flow of each category has a large overlap with each other, or widely scattered in the same category, which is not suitable for classification directly, requires further feature transformation and extraction by the deep learning network before accurate classification.

B) Spatial features map extracted by 2DCNN branch

Figures 10 presents the t-SNE visualization of the data flow feature after the spatial-temporal feature extraction module for the 4th experiment (full flow classification) respectively.

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

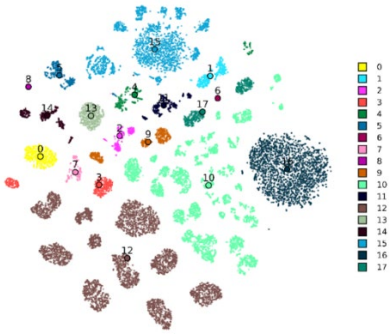


Figure 10: T-SNE representation of features of 2DCNN branch

Figure 10 shows the extracted feature by the 2DCNN branch. Compared with Figure 9, there is basically no longer any overlap between the categories, and 10,12,15,16 categories scattered widely still.

C) Temporal features map extracted by LSTM branch

Figure 11 shows the extracted feature by the LSTM branch. Compared with Figure 9, there is basically no longer any overlap between the categories too, and 10,12,15,16 categories scattered widely too.

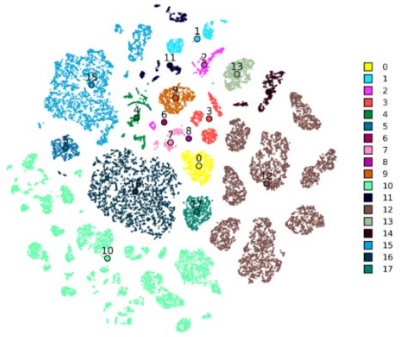


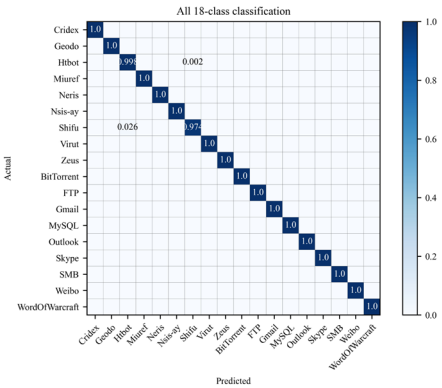
Figure 11: T-SNE representation of features of LSTM branch

Compared with Figure 10, the locations and distribution patterns of the same category of data flow are different, indicating that the 2DCNN and LSTM branches extracted different features from the data flows.

3.3.2 Classification module results

All traffic of the data set was used in this experiment, they were labeled with 0-17 according to its category, and then the number of neurons in the last layer of the classification module was set as 18 for full traffic classification.

The confusion matrix for the classification is given in Figure 12 and the evaluation metrics are given in Table 2.



ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

Figure 12: Confusion matrix of full flow classification for USTC-TFC2016

The results show that the FC-STAM model can accurately identify the different spatial-temporal features of benign and malware traffic. The results show that the accuracy of FC-STAM for full flow classification is very high, almost 100%. All samples of benign traffic are classified correctly, only a few malware samples of Shifu and Htbot categories were classified as incorrect sub-categories, which is considered to be probably due to the characteristics of the application data itself.

In order to compare the classification performance and time overhead of different models. The data flow presentation method proposed in 3.3 of this paper was used to generate sample data, compare the performance of FC-STAM with CBAM-CNN and Attention-LSTM by full flow classification on USTC-TFC2016 datasets. The evaluation metrics used in the comparison is F1-score. Figure 13 shows the classification evaluation metrics of the three models for traffic classifications in the USTC-TFC2016 dataset.

Table 2: Evaluation metrics for full flow classification for USTC-TFC2016 (%)

Benign				Malware	
Category	Recall	Precision	Category	Recall	Precision
Skype	100	100	Zeus	100	100
BitTorrent	100	100	Shifu	97.37	97.37
Gmail	100	100	Neris	100	100
Outlook	100	100	Cridex	100	100
WorldOfWarcraft	100	100	Nsis-ay	100	100
MySQL	100	100	Geodo	100	100
FTP	100	100	Miuref	100	100
SMB	100	100	Virut	100	100
Weibo	100	100	Htbot	99.79	99.79

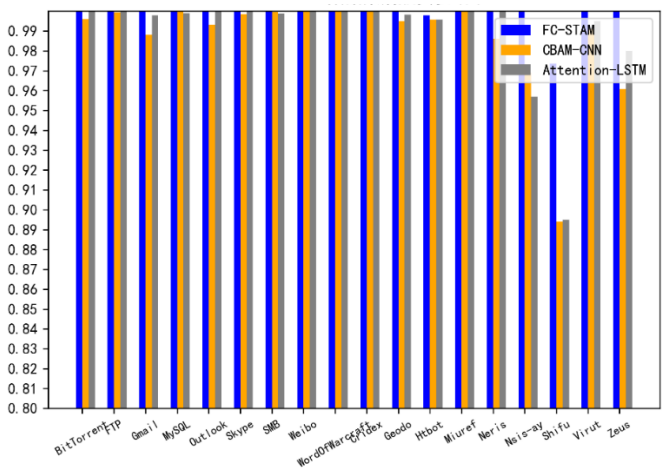


Figure 13: F1-score of each class of USTC-TFC dataset under different models

The results show that the traditional network models have shortcomings in classification performance due to the limitation of the number of samples in the malware traffic class. While the FC-STAM can fully exploit the spatial-temporal characteristics of the samples, and thus solve the problem of low classification accuracy of categories with a few samples. Indicating that FC-STAM can accurately classify malware traffic in the network communication environment where benign traffic dominates.

4. Conclusion

Based on the idea of different types of feature information contained in the packet headers and data loads, we propose a new method of data flows presentation, which redefines data packet into 20 bytes of IP headers, 20 bytes of TCP/UDP headers, and 110 bytes of load data. The data flow was presented with a two-dimensional matrix of packets arriving in order.

In order to extract spatial and temporal features from data flow samples, we design a two-branch network, which extracts spatial and temporal features parallel and didn't interfere with each other.

ISSN 2616-5775 Vol. 7, Issue 9: 1-10, DOI: 10.25236/AJCIS.2024.070901

2DCNN branch uses the channel attention mechanism to extract which feature in different channels is helpful for classification, uses the spatial attention mechanism to identify the location of the useful feature in the same channel. After extraction of spatial and temporal features from two branches separately, concatenating these two features into one feature vector, which contains all the spatial-temporal features of the data flow.

FC-STAM is validated on the USTC-TFC2016 dataset. The t-SNE presentation shows that the feature extraction module transforms and extracts the feature of data flow very well. The classification experiment results show that FC-STAM achieves 100% accuracy in anomalous traffic detection. It performs better than other models in less training time.

Acknowledgements

This work is supported by natural science research projects of Wuhu Institute of Technology (wzyzr202423, wzyzr202424, wzyzrzd202214) and Anhui Province universities natural science research project (KJ2021A1322, 2022AH052198).

References

[1] S. Rezaei and X. Liu, "Deep Learning for Encrypted Traffic Classification: An Overview," in IEEE Communications Magazine, vol. 57, no. 5, pp. 76-81, May 2019.

[2] A. Madhukar and C. Williamson, "A Longitudinal Study of P2P Traffic Classification," 14th IEEE International Symposium on Modeling, Analysis, and Simulation, Monterey, CA, USA, 2006, pp. 179-188.

[3] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin and J. Aguilar, "Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey," in IEEE Communications Surveys & Tutorials, vol. 21, no. 2, pp. 1988-2014 ,Secondquarter 2019.

[4] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997.

[5] Luong M.-T, Pham H and Manning C.D, "Effective approaches to attention-based neural machine translation," in Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, pp. 1412-1421.

[6] Woo S, Park J, Lee J.-Y and Kweon I.S, "CBAM: Convolutional block attention module," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11211 LNCS , pp. 3-19,2018.

[7] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng, "Malware traffic classification using convolutional neural network for representation learning," 2017 International Conference on Information Networking (ICOIN), Da Nang, Vietnam, 2017, pp. 712-717 .

[8] Hwang, Ren-Hung, Peng, Min-Chun, Nguyen, Van-Linh, Chang, Yu-Lun, "An LSTM Based Deep Learning Approach for Classifying Malware Traffic at the Packet Level," in Applied Sciences. 9(16), 3414, 2019.

[9] van der Maaten, Laurens & Hinton, Geoffrey, "Viualizing data using t-SNE," in Journal of Machine Learning Research. 9. 2579-2605, 2008.