Preprints.org

Article

Not peer-reviewed version

CLSTM-MT：Encryption Traffic Classification Based on CLSTM and Mean Teacher Collaborative Learning

XiaoZong Qiu , Guo Hua Yan , LiHua Yin *

Posted Date: 30 December 2024

doi: 10.20944/preprints202412.2397.v1

Keywords: Encrypted Traffic Classification; Convolutional Neural Network (CNN); Bidirectional Long Short-Term Memory (BiLSTM); Semi-Supervised Learning; Deep Learning

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

CLSTM-MT: Encryption Traffic Classification Based on CLSTM and Mean Teacher Collaborative Learning

XiaoZong Qiu [1], GuoHua Yan [2] and LiHua Yin [2,*]

[1] Cyberspace Institute of Advanced Technology,GuangZhou University,Guangzhou 510006,China
[2] Cyberspace Institute of Advanced Technology,GuangZhou University,Guangzhou 510006,China
* Correspondence: yinlh@gzhu.edu.cn

Abstract: The identification and classification of traffic is of great significance for maintaining network security, optimizing network management and providing reliable service quality. These functions not only help prevent malicious activities such as network attacks and illegal intrusions, but also effectively support the reasonable allocation of network resources and improve user experience. However, although the wide application of network traffic encryption technology enhances the security of data transmission, it also makes the content of traffic difficult to be directly analyzed, resulting in the existing identification technology is inefficient in the face of encrypted traffic and difficult to accurately classify. This not only affects the maintenance of network security, but also limits the further improvement of network service quality. Therefore, developing efficient and accurate encryption traffic identification methods has become an urgent problem to be solved. However, the existing work still has three main inherent limitations: (1) The potential relationship between the flow load feature and the sequence feature is ignored in the feature extraction process. (2) To adapt to the characteristics of different protocols to ensure the accuracy and robustness of encrypted traffic identification. (3) Training effective deep learning models requires large amounts of manually labeled data. This study aims to propose a method of encrypted traffic recognition based on CLSTM (a combination of 2-conv CNN and BiLSTM) and Mean Teacher collaborative learning. By detecting the fusion features of traffic load features and sequence features, the accuracy and robustness of encrypted traffic identification are improved, and the dependence of the model on labeled data is reduced. The experimental results show that the proposed CLSTM-MT collaborative learning method not only outperforms the traditional methods in the task of encrypted traffic identification and classification, but also improves the performance of the model by using only a small amount of labeled data when the cost of data labeling is high.

Keywords: Encrypted Traffic Classification; Convolutional Neural Network (CNN); Bidirectional Long Short-Term Memory (BiLSTM); Semi-Supervised Learning; Deep Learning;

1. Introduction

With the rapid development of Internet technology, network traffic has become an indispensable part of modern society. Encryption technology is widely used in data transmission to protect user privacy and data security. However, this also presents many new challenges for network traffic identification. Factors such as the diversity of encryption algorithms [3], data security and privacy, and dynamically changing traffic patterns have increased the difficulty of identifying encrypted traffic.

The existing encryption traffic identification methods [1] mainly include feature-based methods, machine learning-based methods, and deep learning-based methods. The feature-based method relies on manual selection and manual extraction of explicit features. Although this method is intuitive and easy to implement, its performance is easily affected by changes in the encryption algorithm, resulting in low recognition accuracy. Although the method based on machine learning

© 2024 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

can automatically learn features from data, reduce the dependence on specific features, and improve the flexibility of the model, it still faces the problems of difficult training and high cost of data annotation when dealing with encrypted traffic. However, deep learning-based methods rely heavily on manual annotation of data during model training. In particular, when new encryption protocols or algorithms are encountered, the robustness and adaptability of existing models are often insufficient, and they cannot respond to new situations quickly and effectively. Therefore, there is an urgent need to develop a new method that combines high accuracy and low-cost data annotation requirements with strong robustness and adaptability to accurately identify encrypted application traffic.

The objective of this study is to propose a method for encrypted application traffic identification based on CLSTM (a combination of CNN and BiLSTM) and Mean Teacher collaborative learning, aiming to: 1. Improve recognition accuracy: By leveraging the strong feature extraction capabilities of CNN and BiLSTM, combined with the advantages of Mean Teacher in semi-supervised learning, to enhance recognition accuracy. 2. Enhance robustness and adaptability: By utilizing consistency constraints from unlabelled data to improve the model's robustness and adaptability in recognizing different types of encrypted traffic. 3. Reduce the cost of traffic labeling: By using the Mean Teacher framework to train the model with a small amount of labeled data and a large amount of unlabeled data, thereby reducing overall training costs. To achieve these objectives, we designed the following scheme: 1. Dataset preparation: Collect and preprocess encrypted traffic data from publicly available datasets. 2. CLSTM model design: Construct a feature extraction model combining CNN and BiLSTM. 3. Integration of the Mean Teacher framework: Integrate the Mean Teacher framework into the CLSTM model. 4. Experimental validation: Design experiments to verify the effectiveness of the proposed method, perform ablation studies, and compare its performance with existing methods. Through the above methods, we aim to propose a novel approach that excels in the field of encrypted traffic identification, not only improving the accuracy of traffic recognition but also maintaining good robustness and adaptability while reducing the cost of traffic data labeling.

The remainder of this paper is organized as follows. In Section 2, we analyze and summarize related work in the field of encrypted traffic classification. In Section 3, we introduce the system architecture of CLSTM-MT, detailing the data preprocessing module and the classification process of the model. In Section 4, we provide a detailed description of the experimental environment, the datasets used, the evaluation metrics, and conduct an assessment and visualization analysis of the experimental results. Finally, in Section 5, we conclude the paper.

2. Related Work

2.1. Rule-Based Methods

Rule-based methods [12,13] typically rely on the port number or protocol identifier of the traffic to identify the application, such as HTTP traffic on port 80 or port 443. However, these methods are almost ineffective in encrypted traffic identification [4] because encryption protocols (such as TLS/SSL) obscure port information and protocol flags.

2.2. Statistical Feature-Based Methods

Statistical feature-based methods analyze statistical features (such as packet sizes and time intervals) in the traffic for classification. APPScanner [12] uses statistical features of packet sizes to train a random forest classifier, while BIND [13] also utilizes temporal statistical features. Although these features are prominent in non-encrypted traffic, the encryption of data in encrypted traffic makes statistical features less reliable. Therefore, these methods face challenges in encrypted traffic identification.

2.3. Machine Learning-Based Methods

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Machine learning-based methods [5,6] utilize machine learning algorithms, such as Support Vector Machines (SVM) and Random Forests, to learn traffic features. Early researchers used payload data and statistical features [7,8], both of which were suitable for identifying specific scenarios in complex traffic. Hao et al. [9] proposed an improved SVM network traffic classification method by calculating individual feature weights and parameter values for each binary SVM classifier. Despite their good performance in non-encrypted traffic identification, these methods still face challenges in encrypted traffic identification due to the lack of clear feature patterns. Consequently, traditional traffic identification methods often perform poorly in encrypted traffic recognition.

2.4. Deep Learning-Based Methods

A review of existing traffic identification methods reveals that traditional port and protocol-based, statistical feature-based, and machine learning-based methods have significant limitations in encrypted traffic identification.

Deep learning-based methods have shown significant advantages in traffic classification tasks. Yang et al. [10] proposed a method using Convolutional Neural Networks (CNN) to extract features from encrypted traffic, achieving significantly higher classification performance compared to traditional machine learning algorithms. Liu et al. [11] introduced a multi-layer encoder-decoder structure capable of deeply mining the underlying sequential characteristics of traffic flows. Yao et al. [15] modeled time-series network traffic using Recurrent Neural Networks (RNN) and introduced an attention mechanism.

Since data encryption does not alter the overall structure of the data flow, even after encryption, network traffic remains a sequence of data with a start and end point. Therefore, Convolutional Neural Network (CNN) models remain effective for classifying encrypted traffic. However, encryption can scramble request information in certain parts of the traffic protocol, making it difficult to identify features across different data segments. Thus, this study considers incorporating the recognition of temporal features in encrypted traffic to improve traditional deep learning models. In this domain, recurrent neural networks (RNN) and their improved versions, such as BiLSTM, have shown better performance. Therefore, in this study, we combine stacked bidirectional BiLSTM to learn the temporal features of spatial features extracted by CNN.

2.5. Semi-Supervised Learning Methods

Additionally, we consider introducing semi-supervised learning methods to reduce the dependency of deep learning models on labeled datasets. Mean Teacher [17] is a general framework for semi-supervised learning that enhances model robustness and adaptability by leveraging unlabeled data, providing a new solution for encrypted traffic identification. Shi et al. [18] designed a lightweight encrypted traffic classifier based on CNNs, converting traffic data into grayscale images as input and using a semi-supervised learning framework to improve the precision of MT-CNN with only a small amount of labeled data. Alam et al. [19] combined CNNs with autoencoders to develop unsupervised machine learning techniques for detecting anomalies in network traffic. Although these methods reduce the model's dependency on labeled data to some extent, they do not consider the potential relationships between traffic payload features and sequence features.

Therefore, this study aims to combine the advantages of CLSTM (a combination of 2-conv CNN and BiLSTM) and Mean Teacher to propose a more effective method for encrypted traffic identification, addressing the limitations of traditional methods in encrypted traffic recognition.

3. Methodology

3.1. System Architecture of CLSTM-MT

In this section, we will explain the system architecture of the proposed CLSTM-MT model. The overall structure of the encrypted traffic classifier based on the CLSTM-MT model is illustrated in

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Figure 1. The model is divided into two parts: the student model and the teacher model. Both models share the same network architecture but differ in how their parameters are updated.The CNN component of the model consists of six layers: a convolutional layer, a pooling layer, another convolutional layer, another pooling layer, a flattening layer, and a fully connected Softmax layer. The BiLSTM component of both models defines two bidirectional classifiers to process the sequential data of the traffic. The parameters include the input dimension, hidden layer dimension, and number of layers. The fully connected layer maps the output to the required categories for the classification task.

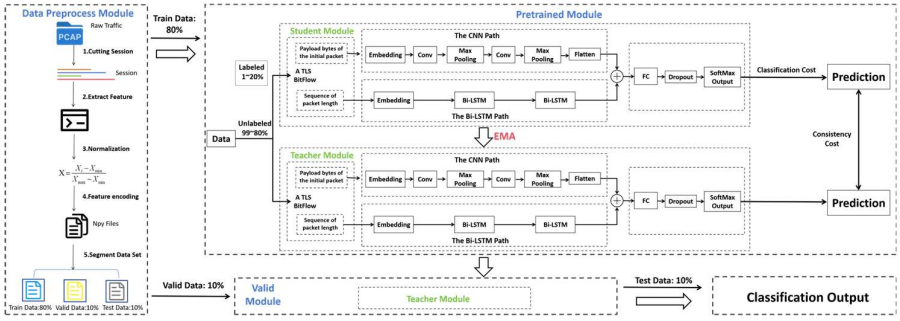

Figure 1. Overview of CLSTM-MT: The architecture of CLSTM-MT is primarily divided into three components: the data preprocessing module, the model training module, and the model validation module. Raw PCAP traffic is processed by the data preprocessing module into four npy feature files. These files are then split into training, validation, and testing datasets in an 8:1:1 ratio.The training dataset is further divided such that a small portion (1% to 20%) of the labeled data is used as input for model training, while a large portion of the unlabeled data (the remaining 99% to 80%) is used as input for model validation. This approach helps train our model effectively.

The input data format for the model consists of four npy feature files, representing the preprocessed traffic data. During training, the data is normalized using the min-max method, mapping the data values from the range [0, 255] to [0, 1]. The output of the model is the classification result for the samples.

3.2. Data Preprocessing Module

Data preprocessing is the first step in traffic classification and is a critical step to ensure the effectiveness of model training. We have taken the following measures to process the rawPCAPtrafficdata:

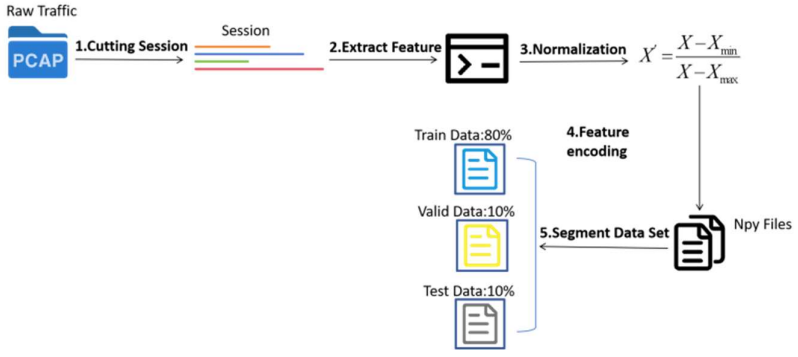

Figure 2. Data preprocessing module: This module is mainly composed of five parts, which are session segmentation, feature extraction, data normalization, feature coding and data set segmentation.

1)Session Segmentation: The purpose of segmentation is to reduce continuous PCAP streams based on sessions. The two most common forms of traffic representation are sessions and flows.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Sessions are traffic units divided based on a five-tuple (source IP, source port, destination IP, destination port, protocol). Flows are very similar to sessions but only contain traffic in one direction, meaning that the source IP/port and destination IP/port cannot be swapped. In this study, continuous PCAP files are segmented based on sessions, converting incoming traffic files into PCAP format data. Each session is limited to a maximum of 100,000 packets and 100,000 bytes to facilitate subsequent uniform processing.

2)Feature Extraction: Before extracting features, we anonymize the packets and select the most useful features for traffic identification.

a. Packet Anonymization: To ensure the normal use of traffic data, we anonymize the IP and MAC addresses of every packet in each session to protect user privacy and the usability of the traffic data.

b. Feature Selection: We extract the following features:Sequence Length Feature (Sequence): The length of the sequence.Payload Feature (Payload): The first Byte_Num bytes of the payload from the first Packet_Num packets.Statistical Feature (Statistic): Statistical features of the traffic.

We set Packet_Num to 4 and Byte_Num to 256 because the initial few packets of network traffic typically contain a large amount of key information, such as TCP/UDP ports, protocol type, sequence numbers, and flags, which are very useful for traffic type identification. Extracting partial data from the first few packets rather than the entire data stream can significantly reduce computational load while still retaining sufficient information for classification.

3)Data Normalization: To enable the model to better learn the underlying relationships in the data, we normalize packet payload data, sequence data, and statistical data to the [0, 1] range using the normalization formula shown in Equation (1).

$$X = \frac{X_i - X_{\min}}{X_{\max} - X_{\min}} \qquad (1)$$

4)Feature Encoding: The processed features are converted into numerical form and saved as npy files to facilitate model processing.

5)Dataset Splitting: Finally, the preprocessed data is split into training and validation sets according to the required proportions.

3.3. Model Design

1) CNN architecture and its role in encryption application traffic identification

Through multi-layer convolution and pooling operations, CNN convolutional neural networks gradually reduce the spatial dimension of encrypted traffic data features while increasing the number of channels and finally carry out Softmax classification through the full connection layer, which can be effectively used to extract local features of encrypted traffic data. Specifically, CNN first automatically extracts relevant features, such as packet load and statistics in traffic data, through the convolutional layer. Then the feature dimension is reduced by the pooling layer, the most important feature information is retained, and finally the classification decision is made by the fully connected layer. The architecture of the model is described below:

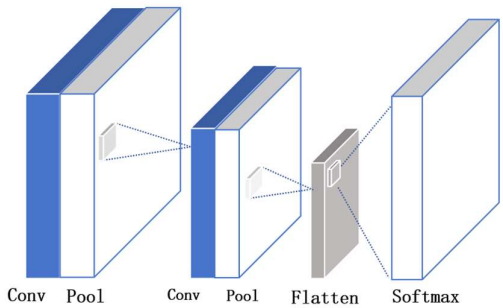

Figure 3. CNN model architecture diagram.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Let xi∈Rk be the k-dimensional vector associated with the ith byte of traffic in a session or flow. A session or flow of length n can be represented as:

$$x_{1:n} = x_1 \oplus x_2 \oplus \ldots \oplus x_n$$

where $\oplus$ denotes the concatenation operator. Generally, let xi:i+j represent the traffic bytes xi,xi+1,…,xi+j.A single convolution operation involves a filter w∈R, which is applied to a window of h traffic bytes to produce a new feature.For example, the feature ci generated by the following formula:

$$c_i = f\left(w \cdot x_{i:i+h-1} + b\right)$$

where f i is an activation function (e.g., ReLU), w is the filter, xi:i+h−1 is the window of h traffic bytes, and b is the bias term.

$$c = \left[c_1, c_2, \ldots, c_{n-h+1}\right]$$

Where c∈R. Then, we apply a max-over-time pooling operation on the feature map, and the maximum value c^=max{c}is used as the next feature. We use multiple convolutional layers and multiple pooling layers to extract features. These features form the penultimate layer and are passed to a fully connected Softmax layer, which produces the final output as the probability distribution of labels for the input session or flow.

Since data encryption does not alter the overall structure of the data flow, even after encryption, the data flow remains a sequence with a start and an end. Therefore, Convolutional Neural Network (CNN) models remain effective for the classification of encrypted traffic, and the classification model for encrypted traffic should still be based on CNNs. However, encryption can completely scramble the request information in certain parts of the traffic protocol, making it difficult to recognize features within different data segments. As a result, the performance of CNN models is inevitably affected.

2) BiLSTM Architecture and Its Role in Encrypted Application Traffic Identification

To address the limitations of Convolutional Neural Network (CNN) technology, we need to improve the model based on the characteristics of encrypted traffic. The temporal features of encrypted traffic primarily manifest in packet length sequences and inter-packet time intervals. The linear trend of the time interval sequence can be considered as the superposition of a horizontal smoothing equation and a trend equation, generally forming a linear recurrence sequence. The packet length sequence, on the other hand, exhibits segmental similarity. Because VPN encryption is achieved through connections to specific websites and nodes, session packets tend to exhibit specific lengths. Combined with the unique handshake mechanisms of encrypted traffic, this results in a sequence that is the superposition of a horizontal smoothing equation and a seasonal smoothing equation.By leveraging the memory of previous numerical features to find the general term of the sequence and predict data, we consider integrating Bidirectional Long Short-Term Memory (BiLSTM) to improve the model.

BiLSTM, a special type of Recurrent Neural Network (RNN), is particularly effective in handling time-series data in encrypted application traffic identification. BiLSTM can effectively process time-series traffic data by using its internal memory cells to remember long-term information, thereby automatically learning the temporal dependencies in the traffic data. Specifically, BiLSTM reads the traffic data sequence and processes it over a series of time steps. At the final layer of the network, it outputs a fixed-length vector that contains the key features of the entire traffic sequence (Sequence). This feature vector is then fed into a fully connected layer for classification decisions.

During the training phase, for a given time step t, the mini-batch input is Lt∈Rn×d, where nn is the number of sequence examples. In the BiLSTM architecture, we assume the forward and backward hidden states at this time step are $\overrightarrow{H}$t∈Rn×h and $\overleftarrow{H}$t∈Rn×h, respectively. Here, hh denotes the number of hidden units. We compute the forward and backward hidden state updates as follows:

$$\overrightarrow{H}_t = \phi\left(L_t \overrightarrow{W}_{lh} + \overrightarrow{H}_{t-1} \overrightarrow{W}_{hh} + \overrightarrow{b}_h\right) \quad (2)$$

$$\overleftarrow{H}_t = \phi\left(L_t \overleftarrow{W}_{lh} + \overleftarrow{H}_{t+1} \overleftarrow{W}_{hh} + \overleftarrow{b}_h\right) \quad (3)$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Here, φ is the activation function of the hidden layer. The weight parameters $\overrightarrow{W}lh$、$\overrightarrow{W}hh$、$\overleftarrow{W}lh$ and $\overleftarrow{W}hh$, and the bias parameters $\overrightarrow{b}h$ and $\overleftarrow{b}h$ are all model parameters. Then, the forward and backward hidden states$\overrightarrow{H}t$ and $\overleftarrow{H}t$ are concatenated to form the hidden state Ht∈Rn×2d.

To further enhance the learning capability of the BiLSTM network, we stack two layers of BiLSTM in the CLSTM-MT model. The hidden state Ht from the first bidirectional layer is passed as input to the second bidirectional layer. Finally, the output layer computes the output Ot using the hidden state Ht from the second layer:

$$O_t = H_t W_{hq} + b_q \quad (4)$$

where Whq and bq are the weight and bias parameters of the output layer, and q is the number of outputs. Since our BiLSTM architecture is a sequence-to-vector RNN model, we use the final output vector Ot as the output of the BiLSTM path. This output is then combined with the features extracted by the CNN path to form a fused representation. The fused features are finally fed into a fully connected layer for classification prediction.

3.4. Mean Teacher Framework Integration

In the task of traffic classification using deep learning models, obtaining a large amount of labeled data is a tedious and time-consuming process. The Mean Teacher framework is a semi-supervised learning method that improves model performance through the interaction between a teacher network and a student network. The parameters of the teacher network are updated using exponential moving average (EMA), while the student network is updated using conventional gradient descent. By maintaining the stability of the teacher network, it guides the learning of the student network.Specifically, during training, the student model is updated based on the total loss, which is the sum of the classification loss and the consistency loss. The parameters of the teacher model are updated using the EMA of the student model's parameters. The EMA calculation is given by Equation (5):

$$\theta_t^{'} = \alpha\theta_{t-1}^{'} + (1-\alpha)\theta_t \quad (5)$$

where θt′is the parameter of the teacher model at time step t, θt−1′is the parameter of the teacher model at the previous time step, θt is the parameter of the student model at time step t, and α is the smoothing coefficient, set to 0.9 in this study. After the training process is complete, we use the teacher model as the model for the validation part.

4. Experimental Evaluation

4.1. Experimental Setup

4.1.1. Data Preparation

In order to verify the effectiveness of the proposed encryption application traffic identification method based on CLSTM-Mean Teacher collaborative learning, we selected the ISCXVPN2016 data set, a publicly available encrypted traffic data set on the Internet [16]. This type of data set contains a variety of common encryption application traffic, covering different types of web applications, and we divide traffic into 14 label categories according to the type of application it belongs to. During training, we divided the incoming data set into a training data set, a verification data set, and a test data set in a ratio of 8:1:1. Among them, the part of the training set is again in the proportion used in the experiment (0.1:9.9; 0.5:9.5; 1:9; 1.5:8.5; 2:8) is divided into labeled data and unlabeled data. Based on this, the validity of encrypted traffic classification on the virtual private network is evaluated. The specific categories of the ISCXVPN2016 traffic dataset are shown in Table 1.

Table 1. ISCXVPN2016 Dataset Label.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Label	Class	Count
1	Aim_Chat	1340
2	Email	5000
3	FaceBook	5000
4	FTPS	5000
5	Hangout	5000
6	ICQ	823
7	Netflix	5000
8	SFTP	5000
9	Skype	5000
10	Spotify	5000
11	Torrent	5000
12	Vimeo	5000
13	VoipBuster	5000
14	Youtube	5000

4.1.2. Equipment Requirements

During the training phase, we use the Adam optimizer to train the model, setting the learning rate lrlr to 0.003 to optimize the entire network with 50 epochs and a batch size of 64. This method is implemented using PyTorch 1.9.0 and trained on a PC equipped with an Intel® Core™ i9-11900K @ 3.50 GHz, 64 GB of RAM, and an NVIDIA GeForce RTX 3090 GPU.

4.1.3. Evaluation Metrics

To comprehensively evaluate the performance of the model, we use four evaluation metrics: Accuracy (AC), Precision (PC), Recall (RC), and F1 Score (F1).

- Accuracy (AC): The proportion of correctly classified samples out of the total number of samples.
- Precision (PC): The proportion of correctly classified positive samples out of all predicted positive samples.
- Recall (RC): The proportion of correctly classified positive samples out of all actual positive samples.
- F1 Score (F1): The harmonic mean of precision and recall.

The specific formulas for each metric are as follows:

$$\mathrm{AC} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$\mathrm{PC} = \frac{TP}{TP + FP} \quad (7)$$

$$\mathrm{RC} = \frac{TP}{TP + TN} \quad (8)$$

$$\mathrm{F1} = \frac{2 * PC * RC}{PC + RC} \quad (9)$$

TP (True Positive): The number of samples that are correctly predicted as positive and are actually positive.

FP (False Positive): The number of samples that are incorrectly predicted as positive but are actually negative.

FN (False Negative): The number of samples that are incorrectly predicted as negative but are actually positive.

TN (True Negative): The number of samples that are correctly predicted as negative and are actually negative.

4.2. Experimental Results Compared to Baseline Models

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

We used five baseline models to validate the performance of the CLSTM-MT model:1-conv CNN: The 1-conv CNN method uses a one-dimensional convolutional layer to directly extract features from the processed NPY files and then classifies the encrypted traffic using these features. 2-conv CNN: The 2-conv CNN method is very similar to the 1-conv CNN, but it uses two convolutional layers and two pooling layers in the convolutional part to extract and pool features. 3-conv CNN: Unlike one-dimensional and two-dimensional convolutions, the 3-conv CNN uses three convolutional layers and three pooling layers in the feature extraction part. BiLSTM: Additionally, we used a Bidirectional Long Short-Term Memory (BiLSTM) network to extract features from the sequential data of the traffic, to evaluate the impact of sequential features on the experimental results. CNN-BiLSTM: By combining the CNN and BiLSTM networks, we integrated the local features and sequential data features of the traffic to validate the effectiveness of this combined network in traffic data processing.

From the experimental results in Table 2, we can observe that as the number of convolutional layers in the CNN increases, the model's performance tends to decline. This result is reasonable. Therefore, we selected the 2-conv CNN, which performed better in classification among the three convolutional models, as part of our base model algorithm. Furthermore, by using the BiLSTM network and adding the sequential data of the traffic as input features to the model, we observed a significant improvement in performance. Consequently, we chose the combined 2-conv CNN and BiLSTM network as the base model architecture for both the student and teacher models in the CLSTM-MT Mean Teacher framework.

Table 2. The performance of each model under different proportions of labeled traffic.

Model	ISCXVPN2016					FLOPs (10^6)	Params (10^6)
	1%	5%	10%	15%	20%		
1-conv CNN	16.8%	31.5%	47.6%	49.8%	52.9%	285.7	5.7
2-conv CNN	27.2%	33.6%	50.8%	52.6%	54.1%	65.3	5.2
3-conv CNN	26.7%	33.9%	46.6%	50.4%	53.2%	93.2	8.5
BiLSTM	30.3%	55.2%	57.7%	62.1%	58.2%	63.8	4.7
2-conv CNN+BiLSTM	32.4%	40.3%	52.9%	55.2%	62.7%	71.9	10.9
CLSTM-MT	**62.1%**	**80.3%**	**83.9%**	**93.4%**	**97.8%**	**34.9**	**4.1**

4.3. Compared with the Experimental Results of Other Advanced Models

To further evaluate the performance of our model, we also compared our model framework with various other methods, including:

- Rule-Based Method: FlowPrint [0]
- Statistical Feature-Based Method: APPScanner [12]
- Machine Learning-Based Method: SVM [9]
- Semi-Supervised and Deep Learning-Based Method: MT-CNN [18]

To better balance the performance of our model with others, the experiment used 20% labeled traffic. From Table 3, it can be seen that compared to other models, the MT-CNN and CLSTM-MT models exhibit superior overall classification performance. However, one limitation of MT-CNN is that it focuses solely on converting network traffic into a graph for type identification, without considering the temporal sequence features of the traffic. This leads to lower recall rates, lower F1 scores, and relatively poorer classification performance compared to the proposed CLSTM-MT model.

Table 3. Compare with other advanced methods.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Method	Accuary	Precision	Recall	F1-Score
FlowPrint	0.6163	0.6697	0.6651	0.6673
AppScanner	0.6266	0.4864	0.5198	0.5030
SVM	0.6767	0.5152	0.5153	0.5150
MT-CNN	0.9329	0.9492	0.9173	0.9330
CLSTM-MT	**0.9784**	**0.9788**	**0.9784**	**0.9783**

Table 3 shows that on the "ISCXVPN2016" dataset, the accuracy of CLSTM-MT reaches as high as 97.84%. The 2-conv CNN + BiLSTM model, combined with the Mean Teacher framework, achieves better traffic classification results under different labeled traffic data compared to the baseline models. This not only significantly improves the accuracy of traffic identification but also substantially reduces the required running time and model parameter size. In summary, compared to the other five methods, our model framework achieves the highest overall accuracy, precision, recall, and F1 score.

Among them, the identification accuracy, recall rate and F1 score of each traffic type are shown in Table 4.

Table 4. Classification results of different types of traffic.

Class	**CLSTM-MT**			MT-CNN		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
AIM_Chat	1.00	0.91	0.95	0.91	0.92	0.91
Email	1.00	0.96	0.98	0.93	0.95	0.94
Facebook	1.00	0.93	0.96	1.00	0.91	0.95
FTPS	1.00	0.94	0.97	0.96	0.92	0.94
Hangout	0.99	0.99	0.99	0.94	0.96	0.95
ICQ	1.00	0.91	0.95	0.93	0.93	0.93
Netflix	0.91	0.99	0.95	0.90	0.94	0.92
SFTP	1.00	0.86	0.93	0.92	0.84	0.88
Skype	0.98	0.99	0.99	0.96	0.98	0.97
Spotify	0.95	0.87	0.91	0.92	0.84	0.88
Torrent	0.98	0.99	0.99	0.88	0.86	0.87
Vimeo	0.95	0.95	0.95	0.91	0.93	0.92
VoipBuster	1.00	1.00	1.00	0.92	0.96	0.94
Youtube	0.94	0.98	0.96	0.90	0.92	0.91

4.4. Ablation Experiments and Results

In order to determine the effectiveness of each component of CLSTM-MT, we conducted ablation experiments on it to better verify the effectiveness of the model performance. As can be seen from Table 5, the Mean Teacher semi-supervised model architecture has a huge impact on model performance. Thanks to the extraction of traffic features by the CNN convolutional network and the processing of traffic sequence data by the BiLSTM network, the Mean Teacher architecture can exert the best performance.

Table 5. The results of ablation experiment were compared.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

Model	ISCXVPN2016				
	1%	5%	10%	15%	20%
Ours	**62.1%**	**80.3%**	**85.4%**	**93.4%**	**97.8%**
Ours w/o 2-conv CNN	51.1%	54.6%	69.4%	73.5%	85.6%
Ours w/o LSTM	53.2%	62.3%	75.3%	81.1%	91.2%
Ours w/o MT	32.4%	40.3%	52.9%	55.2%	62.7%

4.5. Analysis of Visual Experiment Results

In the field of traffic classification, confusion matrix is usually used to verify the classification results of different traffic data, so as to evaluate the classification performance of the model. The performance confusion matrix of CLSTM-MT is shown in Figure 4, with the dark elements on the main diagonal indicating that the CLSTM-MT model performs well for the classification of each application with a small confusion error.

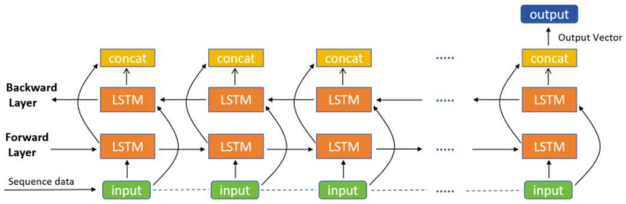

Figure 4. LSTM model architecture diagram.

Meanwhile, the training process of accuracy and loss of the model on the ISCXVPN2016 data set is shown in Figure 5. It can be seen that the accuracy rate of the model is 65.2%, and the loss is 1.87 at epoch 1. The accuracy of the model gradually converges to 96%, and the loss converges to less than 0.1. The above results show that this scheme has high experimental accuracy and model loss, and can perform well in encryption application traffic identification tasks with less data in the case of high cost of data annotation.

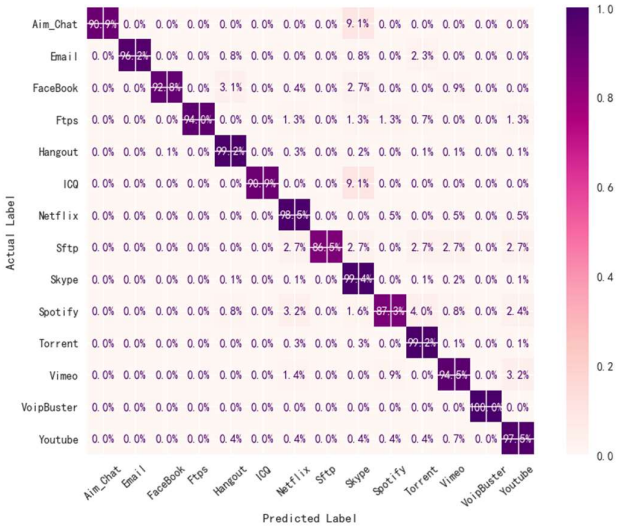

Figure 5. Recall rate confusion matrix of classification performance of ISCXVPN2016 traffic dataset by CLSTM-Mean Teacher model.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

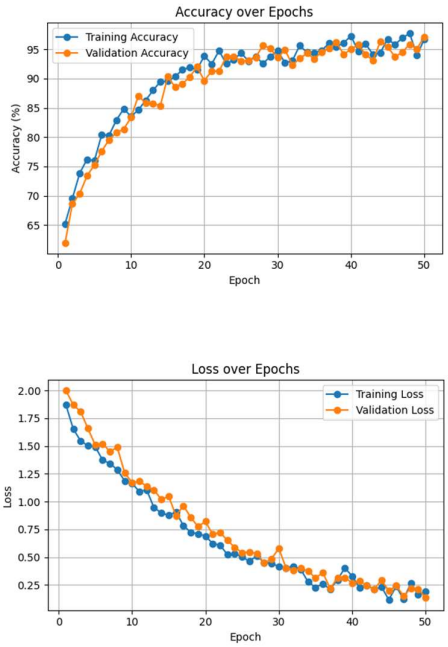

Figure 6. Convergence process of accuracy and loss on ISCXVPN2016 dataset.

5. Conclusions

This study aims to solve the challenges of traffic identification in cryptographic applications and proposes a collaborative learning method based on CLSTM-Mean Teacher. Through detailed experimental validation and result analysis, we demonstrate the significant advantages of this approach in improving the accuracy of traffic identification for encryption applications. The experimental results show that the proposed collaborative learning method based on CLSTM-Mean Teacher performs well in the task of encryption application traffic identification. The model is significantly better than the baseline method in accuracy, recall, and F1 scores. The main contributions of this study are: By combining the powerful feature extraction capability of CLSTM and the advantages of the Mean Teacher framework in semi-supervised learning, a new encryption application traffic identification method is proposed, which not only improves the identification accuracy of encrypted traffic but also has strong robustness and adaptability and can effectively identify and classify encrypted traffic. And it can significantly reduce the model's dependence on labeled data. It has important theoretical significance and application value for many fields, such as network security, traffic management, and service quality control.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: In this section, This work was supported by the National Key R&D Program of China (No. 2022YFB3104100).

References

1. S. Rezaei, X. Liu, Deep learning for encrypted traffic classification: An overview, IEEE Commun. Mag.57 (5) (2019) 76–81.
2. Alberto Dainotti, Antonio Pescape, and Kimberly C Claffy. "Issues and future directions in traffic classification". In: IEEE network 26.1 (2012), pp. 35–40.
3. Liangchen Chen et al. "Research status and development trends on network encrypted traffic identification". In: Netinfo Secur.(03) (2019), pp. 19–25.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 December 2024 doi:10.20944/preprints202412.2397.v1

4. Soheil Hassas Yeganeh et al. "Cute: Traffic classification using terms". In: 2012 21st International Conference on Computer Communications and Networks (ICCCN). IEEE. 2012, pp. 1–9.
5. Perdisci, R., Lee, W., & Jajodia, S. (2014).A deep-learning approach to detecting encrypted malicious web traffic.In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1271–1280). ACM.
6. Chen, L., Li, Y., & Jiang, X. (2020).A deep convolutional neural network approach for encrypted traffic classification. Computer Communications, 155, 151–161.
7. Arash Habibi Lashkari., Gerard Draper Gil., Mohammad Saiful Islam Mamun., and Ali A. Ghorbani. 2017. Characterization of Tor traffic using time based features. In International Conference on Information Systems Security and Privacy. 253–262.
8. Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. In Network and Distributed System Security Symposium (NDSS).
9. Shengnan Hao, Jing Hu, Songyin Liu, Tiecheng Song, Jinghong Guo and Shidong Liu, "Improved SVM method for internet traffic classification based on feature weight learning," 2015 International Conference on Control, Automation and Information Sciences (ICCAIS), Changshu, 2015, pp. 102-106.
10. Ying Yang, Cuicui Kang, Gaopeng Gou, Zhen Li, and Gang Xiong. 2018. TLS/SSL encrypted traffic classification with autoencoder and convolutional neural network. In IEEE International Conference on High Performance Computing and Communications. 362–369.
11. Chang Liu, Longtao He, Gang Xiong, Zigang Cao, and Zhen Li. 2019. FS-Net: a flow sequence network for encrypted traffic classification. In IEEE International Conference on Computer Communications (INFOCOM). 1171–1179.
12. Thijs van Ede, Riccardo Bortolameotti, Andrea Continella, and et al. 2020. FlowPrint: Semi-Supervised Mobile-App Fingerprinting on Encrypted Network Traffic.
13. A. Madhukar, C. Williamson, A longitudinal study of P2P traffic classification, in: 14th IEEE International Symposium on Modeling, Analysis, and Simulation, IEEE, 2006, pp. 179–188.
14. Adil Fahad, Abdulmohsen Almalawi, Zahir Tari, Kurayman Alharthi, Fawaz S. Al-Qahtani, and Mohamed Cheriet. 2019. SemTra: A semi-supervised approach to traffic flow labeling with minimal human effort. Pattern Recognition 91 (2019), 1–12.
15. Haipeng Yao et al. "Identification of encrypted traffic through attention mechanism based long short term memory". In: IEEE transactions on big data 8.1 (2019), pp. 241–252.
16. G. Draper-Gil, A.H. Lashkari, M.S.I. Mamun, A.A. Ghorbani, Characterization of encrypted and VPN traffic using time-related features, in: ICISSP 2016 - Proceedings of the 2nd International Conference on Information Systems Security and Privacy, no, Icissp, 2016, pp. 407–414.
17. Tarvainen A, Valpola H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results[J]. Advances in neural information processing systems, 2017, 30.
18. K. Shi, Y. Zeng, B. Ma, Z. Liu and J. Ma, "MT-CNN: A Classification Method of Encrypted Traffic Based on Semi-Supervised Learning," GLOBECOM 2023 - 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia, 2023, pp. 7538-7543.
19. Shumon Alam, Yasin Alam, Suxia Cui, and Cajetan M Akujuobi. Unsupervised network intrusion detection using convolutional neural networks. In 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), pages 0712–0717. IEEE, 2023.
20. Alberto Dainotti, Antonio Pescape, and Kimberly C Claffy. "Issues and future directions in traffic classification". In: IEEE network 26.1 (2012), pp. 35–40.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.