SSL/TLS ENCRYPTED TRAFFIC APPLICATION LAYER PROTOCOL AND SERVICE CLASSIFICATION

Kunhao Li, Bo Lang, Hongyu Liu and Shaojie Chen

State Key Laboratory of Software Development Environment, Beijing, China

ABSTRACT

Network traffic protocols and service classification are the foundations of network quality of service (QoS) and security technologies, which have attracted increasing attention in recent years. At present, encryption technologies, such as SSL/TLS, are widely used in network transmission, so traditional traffic classification technologies cannot analyze encrypted packet payload. This paper first proposes a two-level application layer protocol classification model that combines packets and sessions information to address this problem. The first level extracts packet features, such as entropy and randomness of ciphertext, and then classifies the protocol. The second level regards the session as a unit and determines the final classification results by voting on the results of the first level. Many application layer protocols only correspond to one specific service, but HTTPS is used for many services. For the HTTPS service classification problem, we combine session features and packet features and establish a service identification model based on CNN-LSTM. We construct a dataset in a laboratory environment. The experimental results show that the proposed method achieves 99.679% and 96.27% accuracy in SSL/TLS application layer protocol classification and HTTPS service classification, respectively. Thus, the service classification model performs better than other existing methods.

KEYWORDS

SSL/TLS, HTTPS, Protocol Classification, Service Classification.

1. INTRODUCTION

SSL/TLS encryption technology has the advantages of high security and low cost and is widely used for secure communication of network applications. The protocol and service classification of encrypted network traffic are the basis of network service quality and network security technologies, which have received increasing attention. Since most of the content of the packets transmitted is encrypted, traditional traffic classification technology, such as deep packet inspection (DPI), has difficulty detecting SSL/TLS traffic[1]. To solve the above problems, some researchers have focused on machine learning based methods. Because different applications and protocols have different functions, the statistical features of the generated traffic data are also different. Machine learning methods can find these differences and classify the traffic. Even though traffic is encrypted, its statistical features are still not affected, so the method can identify it.

In recent years, deep learning has made great achievements in computer vision and natural language processing. In the field of computer networks, technology has also attracted attention. Compared with traditional machine learning methods, this method does not require cumbersome

David C. Wyld et al. (Eds): EMSA, SEA, AIFU, NLCAI, BDML, BIoT, NCOM, CLOUD, CCSEA, SIPRO - 2022
pp. 237-252, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.120621

feature engineering. Instead, network traffic packets are directly inputted into the neural network, and the convolutional layers extract features to complete the classification task.

At present, research on application layer protocol classification of SSL/TLS encrypted traffic is still lacking. For service classification, traditional machine learning methods usually only extract features from the time and length of the network flow, while these methods do not make full use of the semantics of the packet content; the existing deep learning-based methods only use the first few packets of the SSL/TLS flow. The content is not portrayed from the global level of the flow. This paper comprehensively analyzes the characteristics of SSL/TLS single packet and session data and proposes a two-level application layer protocol classification model combining single packet and session. This model extracts features, such as entropy and randomness, from the ciphertext in a single packet and then classifies the protocol. According to the labels of packets in the same session, we build a voting model to determine the traffic protocol. For the problem of HTTPS service classification, we propose a method fusing the global session features and time sequence features, which fully utilizes the encrypted network flow information and improves the task's accuracy. The contributions of this paper mainly include the following:

1) We propose an SSL/TLS application layer protocol classification method combining ciphertext features and a voting mechanism. The method first extracts ciphertext features and uses a machine learning model to complete single-packet protocol determination. Then we use a voting scheme to realize the application layer protocol classification of SSL/TLS sessions.

2) We propose an HTTPS service classification method based on feature engineering and deep learning. This method establishes a CNN-LSTM model to extract the time-series features of the packets in the SSL session and merges them with the global features of the session.

3) In a laboratory environment, we construct the dataset from many sources, including Chrome, Foxmail, FileZilla, etc. We apply the two methods mentioned above to the dataset. The accuracy of the application layer protocol classification method achieves 99.679% and the accuracy of the HTTPS service classification method reaches 96.27%, which is better than the existing machine learning and deep learning methods.

The rest of this paper is organized as follows. Section 2 describes related work. Section 3 introduces the details of our proposed methods. Section 4 presents experimental results. Finally, the paper is concluded in Section 5.

2. RELATED WORK

In the early Internet, every application/protocol used a fixed port number assigned by the Internet Assigned Numbers Authority (IANA)[2]. Therefore, according to the port field in the TCP/UDP header, the application types and protocol types of flow can be classified. For example, HTTPS uses port 443, and SMTPS uses port 456. In recent years, port-based methods have not been more effective they as previously were, because dynamic ports are widely used and new applications have emerged continuously. DPI classifies traffic through pattern matching on the payload in the packet, but it is still difficult to adapt to the encrypted network environment.

At present, research on the application layer protocol classification of network encrypted traffic is still lacking. Some network encryption traffic service classification methods have emerged, mainly including traditional machine learning-based and deep learning-based methods.

Traditional Machine Learning Methods: Because the statistical features of the traffic generated by different applications or services have certain differences in the spatial and temporal dimensions, machine learning methods can utilize the features to classify traffic. Such methods usually include two steps: feature extraction and model training. Features are mainly composed of packet length features, packet ordering features, and packet timing features, which include the number of packet bytes, the packets' time interval, and the flow duration, etc. The models mainly include KNN, SVM and random forest, etc. These models work well on small datasets and do not rely on hardware. However, feature engineering requires much time and professional knowledge to support.

Lashkari et al.[3] regarded unidirectional and bidirectional encrypted traffic flow as the units and extracted timing-related features such as flow duration and packet time interval to train KNN and C4.5 models, which classify different services of encrypted traffic. Dominik et al.[4] used SVM to distinguish whether HTTPS traffic is a mail service. They extracted features, including the duration of the session, the different patterns of daily/weekly traffic usage, and the inherent periodicity.

Deep Learning Methods: This kind of method can automatically learn features and classify encrypted traffic. It does not rely on complex and high-cost feature engineering. The methods can directly deal with packet data and achieve good classification performance.

Wei et al.[5] applied the end-to-end method to classify encrypted traffic for the first time. They proposed a one-dimensional CNN method. Lotfollah et al.[6] first removed the ethernet header and conducted normalization of the packet. Then, they designed SAE and one-dimensional CNN models to classify the service type of traffic. Mingze et al.[7] proposed a text-based convolutional neural network (Text-CNN). He et al.[8] proposed an image-based convolutional neural network (Image-based CNN). They are also better than traditional machine learning methods in service classification.

In addition, RNNs and their variant models have also achieved satisfactory results in service classification. Zhuang et al.[9] combined a CNN with a LSTM and extracted the packet features and the sequence features to classify the encrypted traffic service. Haipeng et al.[10] proposed two models, an attention-based LSTM and a hierarchical attention network (HAN) to model sequential traffic. Liu et al.[11] proposed attention-based bidirectional GRU networks to solve the problem of HTTPS traffic classification. The bidirectional GRU layer is used to extract the forward and backward features of the byte sequence in the session, and the attention layer assigns weights according to the contribution of features to the classification.

3. METHODOLOGY

There are still relatively few achievements in the current encryption traffic classification studies for application layer protocol identification. In addition, service classification methods cannot comprehensively describe the characteristics of network flows. Therefore, we propose a two-level traffic classification framework to solve these problems. For the application layer protocol classification task, we consider the characteristics of ciphertext and propose a classification method combining single packet features and session features; for the service classification task, we extract the global features and time sequence features of the flow, and propose a CNN and LSTM-based classification model, which describes the flow from a comprehensive perspective.

3.1. Framework

The SSL/TLS protocol consists of two layers (as shown in Table 1). The bottom layer is the SSL/TLS record protocol, which is responsible for encrypting packets with a symmetric key. The upper layer is the SSL/TLS handshake protocol, which is divided into four subprotocols:

Handshake Protocol, Change Cipher Spec Message Protocol, Alert Message Protocol and Application Data Protocol.

Table 1. Structure of SSL/TLS protocol

Record Layer		
Content Type	Version	Length
Handshake Protocol (Content Type= 0x16)		
Change Cipher Spec Message (Content Type = 0x14)		
Application Data (Content Type = 0x17)		
Alert Message (Content Type = 0x15)		

The SSL/TLS record protocol consists of content type, version and length fields. The content type field represents the subtype of the recording protocol. The version field represents the version of the SSL/TLS protocol. Content type and length fields represent the type and length of the remaining packet content, respectively. For instance, if Content Type=0x16, the rest content is the content of the handshake protocol; if Content Type=0x17, the rest is the encrypted data in the transmission phase.

Figure 1 shows the framework of the two-layer classification model proposed in this paper. We regard SSL/TLS sessions as the detection units. For application layer protocol classification, we trained the protocol classification model for the five most widely used encrypted protocols in the current network environment: HTTPS, FTPS, SMTPS, IMAPS, and POPS. For service classification, FTPS, SMTPS, IMAPS and POPS only carry a single service (FTPS is used for file transfer, and the other three protocols are used for mail). Therefore, these protocols can be directly output as service types. Only the HTTPS protocol carries multiple services(browser, streaming, etc.), so we focus on HTTPS service classification. Therefore, we propose a convolutional and recurrent neural network-based model combining global and sequential features (CRNN-CGSF) to realize HTTPS service classification.

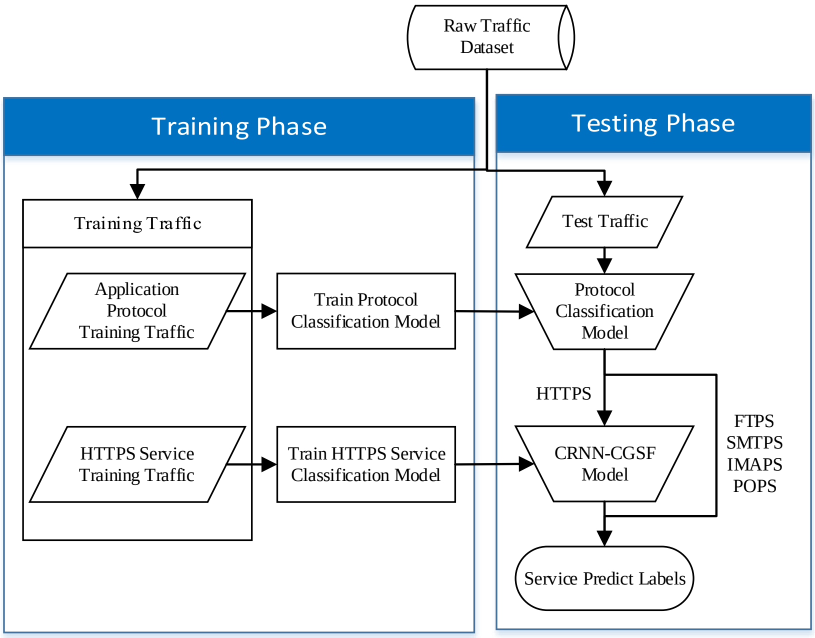
Raw Traffic Dataset
Training Phase
Testing Phase
Training Traffic
Application Protocol Training Traffic
Train Protocol Classification Model
HTTPS Service Training Traffic
Train HTTPS Service Classification Model
Test Traffic
Protocol Classification Model
HTTPS
CRNN-CGSF Model
FTPS
SMTPS
IMAPS
POPS
Service Predict Labels

Figure 1. Framework of application protocols and services classification architecture

As shown in Figure 1, our method is divided into a training phase and a testing phase. In the training phase, the dataset is divided into an application layer protocol training dataset and an HTTPS service training dataset. The protocol classification model and the HTTPS service classification model (CRNN-CGSF) are independently trained. In the testing phase, the protocol classification model is used to identify the application layer protocol, and then the sessions classified as HTTPS are further used to classify the service using the CRNN-CGSF model. Finally, the services of all sessions are output.

Our detection unit is the session. Therefore, the raw flow first needs to be restored to a session before detection. We define the session based on a four-tuple <source IP, source port, destination IP, destination port> (because the protocols are all TCP, so there is no need to express the protocol) bidirectional flow.

According to the protocol specification, handshake phase packets and application data protocol packets should be in a complete session. The standard handshake phase should conform to <client_hello, server_hello, server_hello_done, client_key_exchange, change_cipher_spec> mode or <client_hello, server_hello, change_cipher_spec> mode. Incomplete SSL sessions usually do not have complete handshake phase information or data transmission due to being truncated or from network delays. In addition, we discard this type of flow.

3.2. Application Layer Protocol Classification

In this paper, the application layer protocols include HTTPS, FTPS, SMTPS, IMAPS, and POPS, and their plaintext protocols (HTTP, FTP, SMTP, IMAP, and POP) have different format specifications according to the RFC. Thus, we believe that the data will have different distributions in randomness and entropy after encryption to distinguish different application layer protocols.

The application layer protocol detection framework for SSL/TLS encrypted traffic is shown in Figure 2. The input of the framework is a preprocessed SSL session. Detection is mainly divided into three steps: feature extraction, single-packet classification, and voting:

(1) Feature extraction. We extract all application data protocol packets in every session because these packets contain SSL/TLS header information and are the first packets of a single forward or backward flow in the encrypted data transmission phase. Therefore, the encrypted data of these packets provide the most sufficient format information of the corresponding plaintext protocol. Second, we extract the packets' application data field (i.e. encrypted data) and perform feature extraction on each encrypted data.

(2) Single packet classification. The features extracted from each packet are input into the classifier. The classifier will output the application layer protocol label corresponding to each application data protocol packet in the session.

(3) Voting. Since only one application layer protocol is used in the same session, we vote on the prediction result of a single packet and select the application layer protocol with the highest frequency as the application layer protocol used in the session.

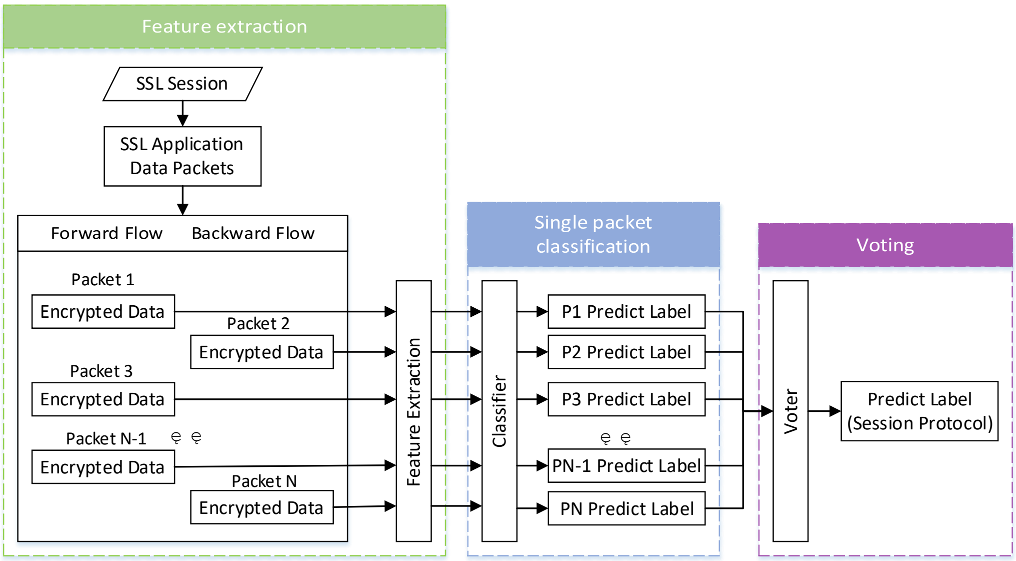

Figure 2. Framework of the application protocol detection

3.2.1. Feature Design

We refer to the randomness detection scheme defined in the National Institute of Standards and Technology (NIST) standards[12] and the ciphertext entropy theory mentioned in the literature [13] and design the features including randomness measurement, entropy, length and bidirectional flow.

Randomness measurement is an important indicator for evaluating the randomness of ciphertext. When the plaintext content of different packet formats is encrypted, the distribution of these characteristic values is still different. Entropy can be used to indicate the uniformity of the ciphertext's byte distribution. The more uniform the byte distribution is, the higher the entropy is[14]. We also use the length feature because different protocols have different length distributions. For example, the lengths of FTPS-Data packets and HTTPS packets are usually hundreds of bytes or even reach the MTU. The lengths of other protocol packets are relatively short. The bidirectional flow features are mainly for FTPS-Data packets, which are all unidirectional in the SSL/TLS encrypted transmission stage, while other protocols are usually bidirectional. The details of these features are shown in Table 2:

Table 2. List of features of the SSL/TLS application package

Feature Type	Feature	Description
Randomness Measurement	Frequency	To detect the proportion of zeroes and ones for the entire sequence.
	Frequency within a Block	To detect the proportion of ones within M-bit blocks.
	Runs	To detect the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits.
	Discrete Fourier Transform	To detect the peak heights in the Discrete Fourier Transform of the sequence.
	Non-overlapping Template Matching	To detect the number of occurrences of pre-specified target strings.

	Overlapping Template Matching	To detect the number of occurrences of pre-specified target strings.
	Linear Complexity	To detect the length of a linear feedback shift register (LFSR)
	Serial	To detect the frequency of all possible overlapping m-bit patterns across the entire sequence.
	Approximate Entropy	To detect the frequency of all possible overlapping m-bit patterns across the entire sequence.
	Cumulative Sums	To detect the maximal excursion (from zero) of the random walk defined by the cumulative sum of adjusted (-1, +1) digits in the sequence.
	Random Excursions	To detect the number of cycles having exactly K visits in a cumulative sum random walk.
	Longest Run of Ones in a Block	To detect the longest run of ones within M-bit blocks.
Entropy	Byte entropy of inter-packet	To measure the randomness of the byte frequency between ciphertext binary packets.
Length	Packet Length	The length of the current packet.
	SSL packet length	The Content-Type field of SSL handshake packet’s header.
Bidirectional	Bidirectional	Indicates if the encrypted data transmission is bidirectional or not.

3.2.2. Packet Classification Model

In the selection of classification models, we consider the currently popular machine learning algorithms, which are mainly divided into three categories:

(1) **Traditional machine learning algorithm**: We use C4.5[17], KNN[18], LR and SVM[19]. These algorithms have the advantages of fitting for nonlinear classification, supporting for numerical and discrete data, and preventing overfitting, and are widely used.

(2) **Integrated learning algorithm**: We use Random Forest[20], Vote, Adaboost, GBDT and XGBoost. Multiple weaker learners integrate these algorithms. Compared with single learners, they usually reach higher accuracy. Moreover, the robustness and generalization ability of these models have also been improved.

(3) **Neural network algorithm**: MLP and DNN are used in this paper. MLP is an artificial neural network with a forward structure. It overcomes the weakness that a single-layer perceptron cannot recognize linear inseparable data. DNN is an improvement over MLP, and overcomes the problem of gradient disappearance caused by the increase of the number of network layers in the multilayer perceptron. In addition, it has more types of activation functions.

3.3. Service Classification

Among the existing service classification methods, machine learning methods only use the time and length features of the network flow. Deep learning methods only focus on the content of the first few packets of the session. Both of them lack a macro description of the entire network flow. Therefore, we propose the CRNN-CGSF model that integrates the global features of the session with the packet time sequence information in the session to solve HTTPS service classification. The model architecture is shown in Figure 3:

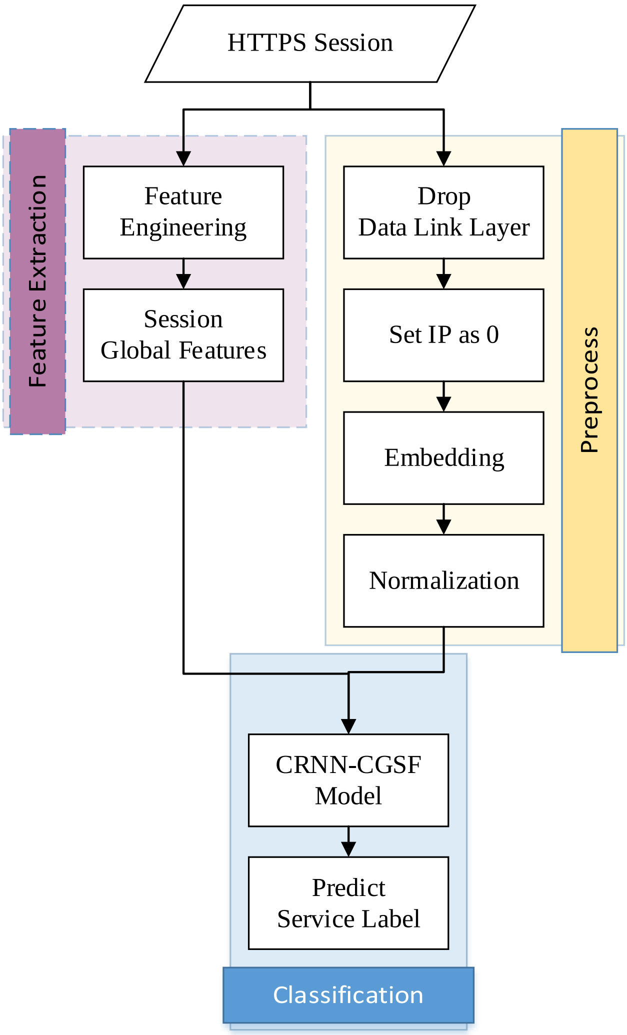

Figure 3. Framework of HTTPS service detection

We first conduct preprocessing and then perform feature extraction and regularization of HTTPS sessions. As shown in Figure 3, the left branch extracts the global features of the session, and the right branch preprocesses the packets in the session.

(1) **Global feature extraction**. We refer to [3] to extract time-related features, including the duration of the flow (duration), forward interarrival time (fiat), backward interarrival time (biat), flow interarrival time (flowiat), active time (active) , idle time (idle), flow bytes per second (fb_psec) and flow packets per second (fp_psec), totaling 23 dimensions.

(2) **Packet preprocessing**. The packet's data link layer (Ethernet frame) contains the MAC address and the IP version. The MAC address is the host identifier and is useless for the task of network traffic classification, although it may affect the classification results; we only pay attention to the ipv4 version of the network traffic, so the IP protocol version is also useless. Thus, we discard the Ethernet frame. The source IP and destination IP in the network layer are unnecessary information, so we replace these fields of the IP header with zeroes. To reduce the input dimension of the model, we convert the bits in the data packet into bytes. Then we conduct normalization, that is, we divide all byte values by 255 and map them to the [0,1] interval; the purpose of this is to obtain a better computing performance.

(3) **Service classification model**. We input the extracted global features of the session and the preprocessed packets into the classification model. The model outputs the predicted HTTPS service label. The model structure is shown in Figure 4:

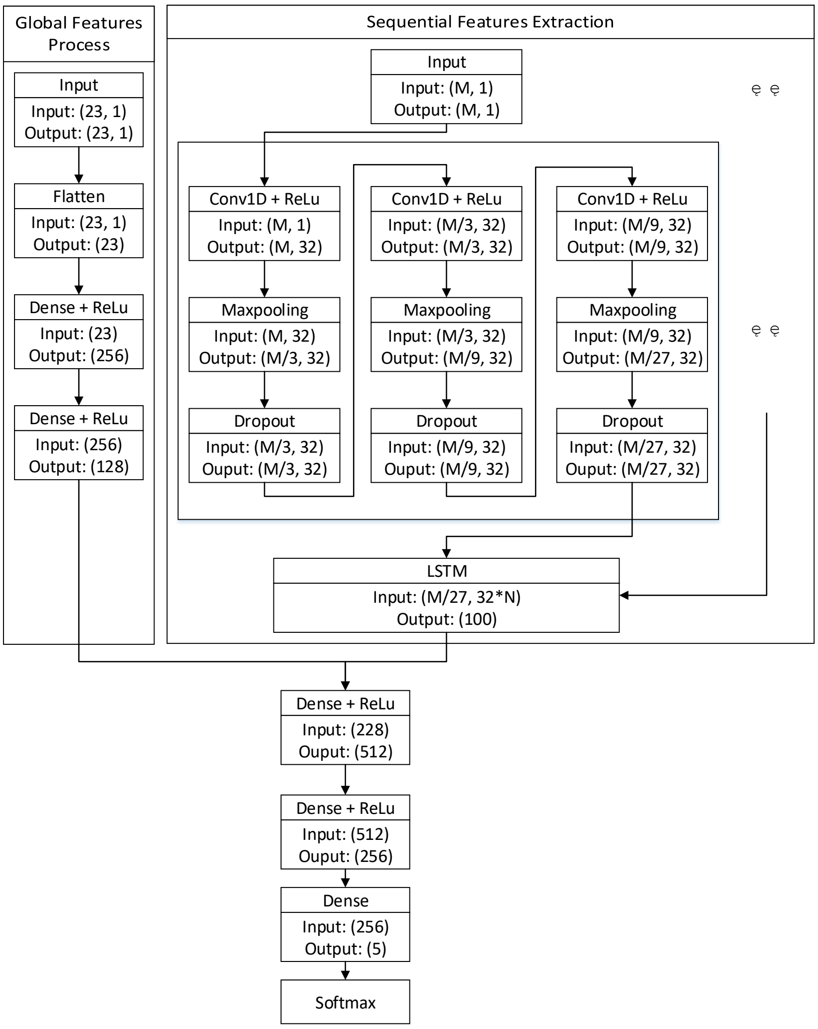

Figure 4. Architecture of the CRNN-CGSF model

In Figure 4, the left branch of the model processes the global features of the session, whose input dimensionality is 23*1. First, the input is flattened to 23 dimensions. Then through two FC layers, the output is 128-dimensions vector. The multiple branches on the right extract the time sequence features from packets. The input is the first M byte stream of the first N packets of the session. The first N packets are selected because the first few packets of the session are in the handshake phase and contain certain semantic information. When the encrypted communication is turned on, the semantics will be greatly weakened. We use CNN to extract the features of each packet byte stream. The CNN layer includes three concatenated Conv1D, Maxpooling, and Dropout. After that, the features of these sequential packets are concatenated and input into LSTM to extract the time sequence features. Then, 100-dimensional time sequence features and 128-dimensional global feature vectors are integrated and input to three FC layers. Finally, the classification result is output through Softmax.

4. EXPERIMENTS

4.1. Dataset

At present, most of the encrypted traffic classification datasets are used for widened meaning protocol classification (SSL, SSH, Tor, etc.). Our task is to subdivide the traffic of SSL/TLS, so we collect the corresponding network traffic as a dataset in a laboratory environment. We use Wireshark to capture the traffic, and the captured packets are stored in pcap.

We browse a large number of web pages with Chrome and Firefox to collect HTTPS traffic. For FTPS traffic, we use two personal computers, using one as an FTP server (encrypted by SSL) and the other as an FTP client. We use FileZilla on the client and remote access the FTP server for file upload and download operations. We enable SMTP/POP and SMTP/IMAP services in a personal QQ mailbox and a 163 mailbox and then use Foxmail to send, receive, and delete mails to collect SMTPS, POPS and IMAPS traffic.

We collect browser service traffic through Chrome and Firefox, but not all the traffic generated by the browser belongs to the browser service. For example, if we use NetEase Cloud Music, the traffic generated, by the browser, that transmits multimedia content belongs to the streaming service traffic. For other services, we use specific applications to collect the traffic data.

The total size of the dataset is 4.7GB and it contains 51 pcap files. The specific content of the dataset is shown in Table 3:

Table 3. List of captured protocols and applications

Service	Protocol	Content	Session Number
Browser	HTTPS	Chrome、Firefox	5908
Streaming	HTTPS	QQ Music、NetEase CloudMusic、Tencent Video	1596
Chat	HTTPS	Weibo Chat、Skype	1280
Mail	HTTPS	buaamail、Tom Mail	1070
	SMTPS	Foxmail	581
	POPS	Foxmail	961
	IMAPS	Foxmail	708
File Transfer	HTTPS	Skype、Baidu Netdisk、115 Pan	1068

	FTPS	FileZilla	378(FTPS-Control)
			2088(FTPS-Data)

We split the dataset into two subdatasets according to two types of tasks: dataset-protocol and dataset-service. The former is used for the SSL/TLS application layer protocol classification task, and the latter is used for the HTTPS service classification task. The traffic of these two datasets is marked with application layer protocol labels and service labels in sessions.

In the upper application layer classification task, we divide FTPS into FTPS-Control and FTPS-Data. The reason is that the plaintext protocol formats of these two FTPS packets are completely different, so there is a great difference in indicators such as entropy and randomness metrics. We can regard it as two subprotocols of FTPS.

4.2. Indicators and Experimental Settings

To evaluate the classification effects of different models, we use the accuracy rate (Acc) to evaluate the overall effect of multiclassification, using precision (Pr), recall (Rc) and F1 scores (F1) that comprehensively consider accuracy and recall to evaluate the effect of a certain type of classification in multiclassification.

We set up 4 groups of experiments:

- **Experiment 1**: We compare the classification effects of different machine learning algorithms on the SSL application layer protocol. As mentioned in Section 4.1, FTPS traffic can be divided into FTPS-Control and FTPS-Data. Therefore, according to FTPS as one class, or divided into two classes, we designed two schemes of five classes and six classes.
- **Experiment 2**: We compare our proposed CRNN-CGSF model with the existing service classification methods, and verify that our method has a higher accuracy.
- **Experiment 3**: We explore the impact of the input dimensions of the CRNN-CGSF model on the performance of HTTPS service classification. The input dimension is determined by the number of intercepted session packets N and the number of intercepted packet bytes M.
- **Experiment 4**: We explore the impact of introducing global features on the classification results of the HTTPS service.

4.3. Results and Analyses

4.3.1. Application layer protocol classification (Experiment 1)

Since the dataset is unbalanced in categories (as shown in Table 3), we sampled the dataset to train the model better. For the single-packet detection experiment, we select 8000 data from each protocol (for the case where the FTPS-DATA samples in the six categories are less than 8000, we select 1500 samples). Then, we divide them into a training set and a test set at a ratio of 4:1. We select 150 sets of session data from each protocol as the test set for the session detection experiment.

We use the machine learning algorithms mentioned in Section 3.2.2 to train the classifiers. The test results are shown in Table 4, 5, and 6:

Table 4. Accuracy of different traditional machine learning methods

		C4.5	KNN	LR	SVM
5 classes	Packet	**0.74406**	0.73496	0.43156	0.39971
	Session	**0.98500**	0.90866	0.69054	0.23491
6 classes	Packet	**0.74496**	0.72971	0.44264	0.41071
	Session	**0.98772**	0.92292	0.66243	0.33378

Table 5. Accuracy of different ensemble learning methods

		RF	VOTE	ADA	GBDT	XGBoost
5 classes	Packet	0.80659	0.81494	0.59589	0.79595	**0.81828**
	Session	0.99670	0.99679	0.94098	**0.99688**	0.99679
6 classes	Packet	0.80734	0.81518	0.56525	0.79764	**0.81871**
	Session	0.99554	0.99628	0.78296	0.99576	**0.99650**

Table 6. Accuracy of different neural network methods

		MLP	DNN
5 classes	Packet	0.55717	**0.69721**
	Session	0.78580	**0.90317**
6 classes	Packet	0.59157	**0.70423**
	Session	0.84665	**0.91438**

In traditional machine learning methods, C4.5 performs far better than other methods. In the six-classification task, the accuracy of single-packet detection and session detection reaches 0.74496 and 0.98772, respectively. In the ensemble learning methods, RF, VOTE, GBDT and XGBoost have similar accuracies in session detection tasks. XGBoost performs best in single-packet detection and six-classification session detection; GBDT performed best in five-classification session detection, with an accuracy that is 0.009% higher than XGBoost. In neural network methods, DNN is better than MLP because DNN has deeper layers and can better fit the data. Still, DNN has limited improvement performance and is far inferior to C4.5, KNN and integrated learning methods.

From the detection point of view, the accuracy of session detection is much higher than that of single-packet detection. Because the accuracy of single-packet detection reaches a certain height, the incorrect single-packet classification is corrected after voting. In terms of methods, integrated learning algorithms are generally better than traditional machine learning algorithms. Integrated learning can combine multiple single learners with a certain strategy, which greatly improves generalization performance. Among all the methods, XGBoost is the most comprehensive. The XGBoost single-packet detection and session detection results of each type of protocol are shown in Table 7, and the confusion matrix of the experimental results is shown in Figure 5:

Table 7. Detailed experimental results of XGBoost (5 classes)

	Packet			**Session**		
	Pr	**Rc**	**F1**	**Pr**	**Rc**	**F1**
FTPS	0.86496	0.79813	0.83020	0.99379	1.00000	0.99689
HTTPS	0.91086	0.89623	0.90349	0.99371	0.98750	0.99060
IMAPS	0.85053	0.75000	0.79711	1.00000	0.99375	0.99687
POPS	0.77748	0.89286	0.83118	0.99375	0.99375	0.99375
SMTPS	0.73533	0.78320	0.75851	0.99379	1.00000	0.99689

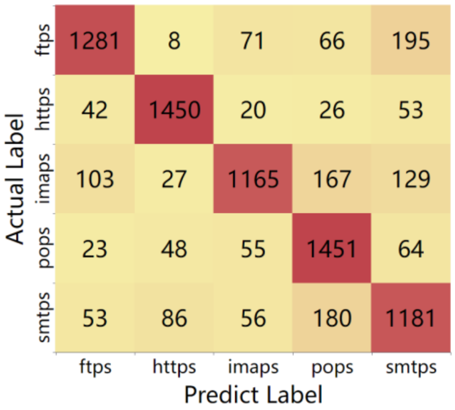

Figure 5. Confusion matrix of the single packet detection result of XGBoost

Observing the results of Table 7 and Figure 5, it can be seen that XGBoost has the best recognition effect on the HTTPS protocol in single-packet detection. The accuracy, recall and F1 score indicators reach 0.91086, 0.89623 and 0.90349, respectively; for the SMTPS protocol, the recognition effect is the worst, but its F1 score also reaches 0.75851. In terms of session detection, XGBoost has a very good classification effect for each protocol, and the F1 score can be stabilized above 0.99.

4.3.2. Comparative experiment on service classification methods (Experiment 2)

We compared the six methods in the five current papers[3,4,5,13,14] containing the same kind of research with our own method. The experimental results are shown in Table 8:

Table 8. Comparative experimental results of HTTPS encrypted traffic service classification

Method	**Acc**
Our method(CRNN-CGSF)	**0.9627**
C4.5[3]	0.8905
KNN[3]	0.7030
1D-CNN[5]	0.9410
SAE[6]	0.9406
LSTM[15]	0.9080
nnDPI[16]	0.9401

According to Table 8, the accuracy of deep learning methods (1D-CNN[5], SAE[6], LSTM[15] and nnDPI[16]) is better than traditional machine learning methods (C4.5[3] and KNN[3]). Among these methods, our model has the best accuracy, reaching 0.9627, which is 0.0217 higher than the second-highest 1D-CNN.

4.3.3. CRNN-CGSF model input dimension experiment (Experiment 3)

To obtain better results in the classification of HTTPS services, we explored the influence of N (flow size) and M (intercept length) on the model classification effect. N represents the number of packets we select in the session. If the value of N is too small, the information in the handshake phase will be incomplete; if the value of N is too large, encrypted data will be used, which will have a certain negative impact on the model performance. M indicates how many bytes we choose from each packet. If the value of M is too small, then the information extraction of each packet will be insufficient; if the value of M is too large, it will have a certain impact on the computational performance overhead of the model.

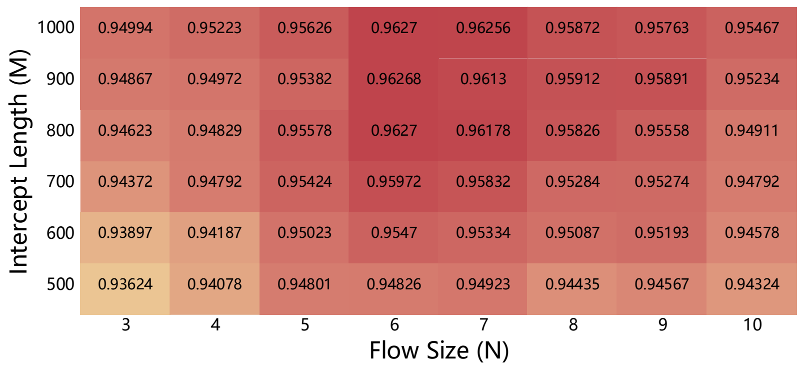

Figure 6. Thermodynamic diagram under different flow sizes and intercept lengths

Figure 6 shows that the accuracy of the model increases as the value of M increases. This is because the longer the intercepted byte length of a single packet is, the richer the information will be. The accuracy of the model first increases with increasing N and then decreases with further increasing N. N=6 can be regarded as a turning point because if the stream length is too long, then model will use the packets of encrypted data transmission, and accuracy will decrease. Because the packet carries ciphertext, it interferes with the model performance. The model achieves the maximum value at N=6, M=800 and N=6, M=1000. Considering the calculation performance of the model, we choose N=6 and M=800.

4.3.4. Validity experiment of introducing global features (Experiment 4)

To verify that the global features are effective, we performed a comparative experiment on global features. The model that does not contain global features is called CRNN-UOSF (convolutional and recurrent neural networks using only sequential features). Compared with the CRNN-CGSF, CRNN-UOSF eliminates the global feature input layer (Input), the flattened layer (Flatten) and the fully connected layer (Dense). The rest of the structure is the same. The experimental results are shown in Table 9:

Table 9. Comparison of experimental results between CRNN-CGSF and CRNN-UOSF

Method	**Acc**
CRNN-CGSF	**0.9627**
CRNN-UOSF	0.9504

The model's accuracy without global features is 0.9504, and the model's accuracy with global features is 1.23% higher than that without global features. Therefore, it can be concluded that the introduction of the global features of the session enables the model to better characterize the session.

5. Conclusions

This paper focuses on the application layer protocol classification and service classification of SSL/TLS encrypted traffic. We first extract features such as randomness and entropy of encrypted data for application layer protocol classification and then use a machine learning model to judge single packets. After that, we utilize voting mechanisms to realize application layer protocol classification for SSL/TLS sessions. The experimental results show that XGBoost has the best comprehensive detection effect. We propose the CRNN-CGSF model combining session global features and packet time sequence features for HTTPS service classification. The model uses CNN and LSTM to effectively utilize the packet byte stream information. In addition, we improve the accuracy of the model by introducing the global features of the session. The experimental results show that the accuracy of our method can reach 96.27%, which is better than the existing traditional machine learning and deep learning methods. Our method can provide preliminary traffic analysis results in network management and QoS and can provide basic support for further analysis procedures. In future work, we will focus on the classification of new versions of protocols such as HTTP2 and QUIC. In addition, our experiments are based on the dataset collected in the laboratory, which may lead to the limitations of our model. We will pay more attention to the traffic in different network environments and further improve the generalization capabilities and robustness of our model.

REFERENCES

[1] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, “A survey on encrypted traffic classification,” in International Conference on Applications and Techniques in Information Security, (Berlin, Heidelberg), pp. 73–81, 2014.

[2] Moore A W, Papagiannaki K. Toward the accurate identification of network applications[C]//International Workshop on Passive and Active Network Measurement. Springer, Berlin, Heidelberg, 2005: 41-54.

[3] Lashkari A H, Draper-Gil G, Mamun M , et al. Characterization of Encrypted and VPN Traffic Using Time-Related Features[C]// The International Conference on Information Systems Security and Privacy (ICISSP). 2016.

[4] Schatzmann D, Mühlbauer W, Spyropoulos T, et al. Digging into HTTPS: flow-based classification of webmail traffic[C]//Proceedings of the 10th ACM SIGCOMM conference on Internet measurement. 2010: 322-327.

[5] Wei W , Ming Z , Wang J , et al. End-to-end encrypted traffic classification with one-dimensional convolution neural networks[C]// 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). IEEE, 2017.

[6] Lotfollahi M , Zade R S H , Siavoshani M J , et al. Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning[J]. Soft Computing, 2017.

[7] Song M, Ran J, Li S. Encrypted traffic classification based on text convolution neural networks[C]//2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT). IEEE, 2019: 432-436.

[8] He Y , Li W . Image-based Encrypted Traffic Classification with Convolution Neural Networks[C]// 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC). IEEE, 2020.

[9] Zhuang Z , J Ge, Zheng H , et al. Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network[C]// 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2018.

[10] Yao H, Liu C, Zhang P, et al. Identification of encrypted traffic through attention mechanism based long short term memory[J]. IEEE Transactions on Big Data, 2019.

[11] Liu X , You J , Wu Y , et al. Attention-Based Bidirectional GRU Networks for Efficient HTTPS Traffic Classification[J]. Information ences, 2020, 541.

[12] Rukhin A . A statistical test suite for random and pseudorandom number generators for cryptographic applications[J]. NIST Special Publication 800-22, 2000.

[13] Mishra S, Bhattacharjya A . Pattern analysis of cipher text: A combined approach[C]// International Conference on Recent Trends in Information Technology. IEEE, 2013.

[14] Wang Y, Zhang Z, Guo L, et al. Using entropy to classify traffic more deeply[C]//2011 IEEE Sixth International Conference on Networking, Architecture, and Storage. IEEE, 2011: 45-52.

[15] Vu L , Thuy H V , Nguyen Q U , et al. Time Series Analysis for Encrypted Traffic Classification: A Deep Learning Approach[C]// 2018 18th International Symposium on Communications and Information Technologies (ISCIT). 2018.

[16] Bahaa M , Aboulmagd A , Adel K , et al. nnDPI: A Novel Deep Packet Inspection Technique Using Word Embedding, Convolutional and Recurrent Neural Networks[C]// 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES). 2020.

[17] Quinlan J R. Improved use of continuous attributes in C4.5[J]. Journal of artificial intelligence research, 1996, 4: 77-90.

[18] Abeywickrama T, Cheema M A, Taniar D. K-nearest neighbors on road networks: a journey in experimentation and in-memory implementation[J]. arXiv preprint arXiv:1601.01549, 2016.

[19] Nello Cristianini and John Shawe-Taylor. 1999. An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods. Cambridge University Press, New York, NY, USA.

[20] Tin, Kam. The random subspace method for constructing decision forests.[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1998, 20(8):832-832.