Only Header: A Reliable Encrypted Traffic Classification Framework without Privacy Risk

Susu Cui (cuisusu@iie.ac.cn)
Institute of Information Engineering CAS: Chinese Academy of Sciences Institute of Information Engineering https://orcid.org/0000-0001-5249-5699

Jian Liu
Institute of Information Engineering CAS: Chinese Academy of Sciences Institute of Information Engineering

Cong Dong
Institute of Information Engineering CAS: Chinese Academy of Sciences Institute of Information Engineering

Zhigang Lu
Institute of Information Engineering CAS: Chinese Academy of Sciences Institute of Information Engineering

Dan DU
Institute of Information Engineering CAS: Chinese Academy of Sciences Institute of Information Engineering

Research Article

Keywords: encrypted traffic classification, capsule neural networks. twice segmentation mechanism, privacy risk

Posted Date: February 15th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-745961/v1

Noname manuscript No.
(will be inserted by the editor)

Only Header: A Reliable Encrypted Traffic Classification Framework without Privacy Risk

Susu Cui [1,2] · Jian Liu [1,2] · Cong Dong [1,2] · Zhigang Lu [1,2] · Dan Du [1] ✉

Received: date / Accepted: date

Abstract Encrypted traffic classification plays a critical role in network management, providing appropriate Quality-of-Service and Network Intrusion Detection. Conventional port-based and deep packet inspection (DPI) approaches cannot classify encrypted traffic effectively. Methods based on machine learning can classify encrypted traffic by extracting statistical features of the flow. However, they require manual extraction of features. Recent studies show that the approaches based on deep learning are compelling for the task. They can automatically learn raw traffic features without manual feature extraction. However, these studies still take the payload of encrypted traffic as the model input, which may cause privacy risks. Besides, a massive encrypted payload causes great storage pressure on traffic classification. In this paper, we propose a reliable encrypted traffic classification framework by only using the flow header called Only Header, which avoids privacy risks and achieves lightweight storage. Firstly, we introduce a twice segmentation mechanism to dilute the interference traffic and increase the weight of effective traffic. Then we use capsule neural networks (CapsNet) to learn spatial and byte features of the flow header. The Only Header's effectiveness is compared with other methods using two public datasets, including ISCX VPN-nonVPN and ISCX Tor-nonTor datasets. The experimental results demonstrate that the Only Header outperforms the state-of-the-art encrypted traffic classification methods.

Keywords encrypted traffic classification · capsule neural networks · twice segmentation mechanism · privacy risk

✉ D. Du
E-mail: dudan@iie.ac.cn

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

1 Introduction

Nowadays, internet traffic classification aims at classifying traffic based on the type of protocol or behavior, which has become a fundamental analytical technique for advanced network management. As a countermeasure to solve the increasingly severe network threats, traffic classification technology can be adopted for identifying the malicious behaviors and then hinder the threats from spreading in time [24]. From another view, with the rapid development of network technology and the gradual rise of novel applications, traffic classification technology can also help to improve the network resource utilization by providing precise traffic type knowledge. Hence, traffic classification is crucial to network management, especially for Network Intrusion Detection (NID) and Quality-of-Service (QoS).

In recent decades, the plain-text network transferring has become a vulnerability with severe consequences, which challenging the regular adoption of the network and users' privacy. Therefore, more and more applications adopt secure protocols such as SSL, VPN, Tor to protect their traffic from being tapped by the Man-in-the-middle attack [10]. Meanwhile, in order to bypass detection by security software such as firewalls, malware software uses encryption techniques to hide communication content. In such a situation, traffic encryption has become a standard practice adopted by benign network applications and malware for different purposes.

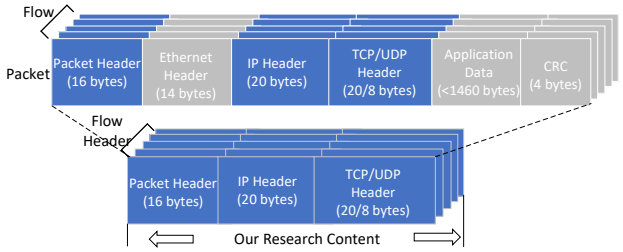


Fig. 1 Definition of the flow header and structure of a packet in the flow. Packet header is the 16-byte Record (Packets) header in the libpcap file format definition, which is a very basic format to save captured network data. Ethernet Header is the 14-byte header of the Ethernet frame. IP Header is the header of the IPv4 packet, consisting 20 bytes. TCP/UDP Header is the header of the TCP/UDP packet, including 20 or 8 bytes. CRC is a 4-byte cyclic redundancy check used to detect any in-transit corruption of data.

Unfortunately, the encrypted traffic brings a challenge to network management, as the payload of the application layer cannot be inspected, making traditional traffic classification approaches don't work [4].

Recently, deep learning performs well for encrypted traffic classification. On the one hand, many studies take the first N (such as 784, 900, 1024, etc.) bytes of encrypted traffic as the model input. They then use convolutional neural network (CNN), stacked autoencoder (SAE), and other models to extract traffic features and achieve service and application classification [15,21,25, 29, 33]. However, the above studies directly touch the application payload, which can easily cause certain privacy troubles [22]. In addition, since a large amount of application payload is used as a feature of the models, such methods put a lot of pressure on data storage. On the other hand, some studies propose to learn the sequence features such as packet sequence of flow and message sequence of flow [21,29]. However, these methods are greatly affected by the environment and user habits [9]. Therefore, they have low robustness.

To tackle the problem mentioned above, we propose a reliable encrypted traffic classification framework without privacy risk, Only Header. It only uses the flow header (shown in Figure 1) as the proposed model's input, which avoids privacy risks and reduces data storage pressure. In more detail, our proposed model first extracts header and splits traffic by a twice segmentation mechanism in preprocessing to dilute the interference traffic and increase the weight of effective traffic. Then, it learns the spatial features and byte features of the flow header using CapsNet that takes the location of fixed strings and the order between packets into consideration. Finally, the traffic is classified by a fully connected softmax layer. To demonstrate the effectiveness of the Only Header, we perform experiments for encrypted traffic identification, regular and VPN traffic classification, regular and Tor traffic classification on the ISCX VPN-nonVPN dataset and ISCX Tor-nonTor dataset. The experimental results demonstrate that our proposed model outperforms the state-of-the-art classification approaches. This paper is a further expansion and deepening of the previous research work [5].

The main contributions of this paper are summarized as follows:

- We propose a reliable encrypted traffic classification framework without privacy risk, Only Header. It only uses the flow header as the proposed model's input, which avoids privacy risks and reduces data storage pressure.
- We propose a novel encrypted traffic classification model using CapsNet. The model is effective as not only the location of fixed strings are taken into consideration, but the order between packets also remains the effective features behind the traffic.
- A twice segmentation mechanism is introduced to increase the effective traffic weight, which shows higher accuracy than traditional traffic representation over packet and flow.
- We evaluate the framework against the state-of-the-art methods on the publicly available ISCX VPN-nonVPN dataset and ISCX Tor-nonTor dataset. Experimental results have demonstrated the proposed model's effectiveness, measured by encrypted traffic identification, regular and VPN traffic classification, and regular and Tor traffic classification accuracy.

2 Related Work

Traffic classification has attracted extensive attention from academic and industrial fields, achieving abundant accomplishments [6, 23]. However, with the widespread application of encrypted traffic, port-based methods [6, 12,16,18] and DPI methods [3,7,13,20,30] are not suitable for encrypted traffic classification. Recently, the methods based on Machine Learning (ML-based) and the methods based on Deep Learning (DL-based) show effective classification results. Since they can identify the encrypted traffic by mining and learning the statistical features. In this section, we outline specific ML-based methods and DL-based methods for encrypted traffic classification.

2.1 ML-based Methods for Encrypted Classification

ML-based methods extract statistical features such as packet size and duration from the traffic samples. They

then use the appropriate ML algorithms to learn the statistical traffic features for encrypted traffic classification. These methods mainly include two parts: feature extraction and model selection. In feature extraction, Moore et al. [17] propose almost 250 flow or packet features for encrypted classification. Okada et al. [19] analyze 49 flow features of encrypted traffic and non-encrypted traffic and obtain strong correlation features such as mean packet size, inter-arrival time (IAT), and transfer time. In general, although time-related features have outstanding classification capability, they show the worse robustness [23]. Therefore, if the traffic classifier is not designed for a specific network, time-related features will easily make the performance of the classifier unstable.

Machine learning models used in encrypted traffic classification mainly include supervised learning models and semi-supervised learning models. Okada et al. [19] propose an encrypted classification method based on the estimation of traffic features called EFM, and then they combine several supervised learning models (SVM, Naive Bayes, C4.5) to achieve application classification of encrypted traffic. Arndt et al. [1] compare C4.5, k-means, and Multi-Objective Genetic Algorithm (MOGA) in encrypted classification. C4.5 shows the best robustness, while MOGA shows the lowest false positive rate. Bar-Yanai et al. [2] propose a real-time classification model of encrypted traffic by combining k-means and KNN algorithms, which takes into account the light complexity of k-means and the accuracy of KNN. Zhang et al. [32] propose an improvement to the k-means algorithm, using the harmonic mean to reduce the impact of random initial clustering scores. This method can increase the accuracy of the k-means algorithm used for encrypted traffic classification.

Given the ML-based methods mentioned above, almost without exception, they have a common disadvantage that they show an over-reliance on feature selection. This process requires a comprehensive prior knowledge of the field so that we may lose essential features. Meanwhile, these methods are challenging to transfer when encountering a new scene.

2.2 DL-based Methods for Encrypted Classification

Deep learning is an effective way to solve the problem of feature design. It can automatically select features from the raw traffic during training instead of extracting features manually [11]. In previous studies, DL-based methods usually take the raw traffic data as input, which includes the underlying protocol layer and the upper application data. Specifically, Wang et al. [27] extract the first 1000 bytes of TCP flow and use a stacked autoencoder (SAE) to achieve encrypted protocol classification. Wang et al. [25] propose to select the first 784 bytes of the raw traffic and then use one-dimensional convolution neural networks (1dCNN) to learn the spatial features for encrypted service classification. Lotfollahi et al. [15] use the IP header and the first 1480 bytes of the IP packet payload as the input of CNN and SAE models to achieve encrypted service and application classification. Zou et al. [33] combine CNN and Long Short Term Memory (LSTM), using CNN to learn in-packet features of the first 784 bytes in a single packet, and using LSTM to learn inter-packet features of any three consecutive packets. Besides, other similar studies also get the same excellent classification accuracy [5, 28, 31]. It can be seen that these methods are based on extracting the first N bytes data of encrypted traffic and then learn the spatial features, sequence features, and byte features of the traffic through suitable deep learning models.

Other methods are to learn encrypted traffic features for the time sequence of traffic (such as packet length sequence, message sequence, etc.), and then use Markov chain, LSTM, etc., to learn the sequence features of encrypted traffic. Yao et al. [29] regard encrypted network flow as a time sequence and build an attention-based LSTM model to learn the flow's sequence features. Shapira et al. [21] create images based on the sequence features of the packet size and the arrival time and use CNN to learn the image's spatial features.

However, the methods based on raw traffic bytes have disadvantages of privacy problems and massive storage pressure. These methods all utilize the encrypted traffic application payload as one of the features of the model. Due to the encryption algorithm, the application payload is irregular ciphertext and does not contain useful features. Moreover, taking the application payload as one of the features increases the pressure of data storage. Secondly, Taylor et al. [22] show that although most applications currently use encryption protocols to protect user data, 80% of applications have both encrypted and unencrypted connections. Developers usually consider the importance and cost of data. Information such as passwords and locations are transmitted in encrypted mode, while other information is still transmitted in plain text. Therefore, 78% of applications have privacy issues.

In addition, the methods based on the time sequence of traffic are unstable and poor in robustness. Although these methods do not involve traffic application payload, the flow's sequence features are easily affected by network performance and user habits, resulting in large differences [9]. Therefore, such methods are less robust.

In order to design an encrypted traffic classification method that is robust and does not involve user privacy, we focus on mining the difference of the flow header in each category of traffic by deep learning models. On the one hand, privacy problems can be avoided. On the other hand, the flow header can reduce the data storage pressure because of its lightweight.

3 The Proposed Model

In this section, we propose a reliable encrypted traffic classification framework without privacy risk, Only Header. It uses CapsNet to learn the spatial features and byte features of the flow header. The details of our proposed model are shown in Figure 2, which consists of extracting flow header, training CapsNet for encrypted classification. Finally, the fine-tuned model is applied to traffic identification, regular and VPN traffic classification, regular and Tor traffic classification.

3.1 Extracting Flow Header

In this section, we design extracting flow header as the first part of the Only Header by the following steps: extracting header, twice segmentation, flow padding. We advocate only extracting flow header from the raw traffic, which can avoid privacy risk and reduce data storage pressure. Besides, we introduce a twice segmentation mechanism to dilute the interference traffic and increase effective traffic weight. Hence, extracting flow header can achieve extracting header bytes, traffic segmentation, traffic cleaning, and traffic standardization.

3.1.1 Extracting Header

The application payload is easy to cause privacy risks and enormous pressure on data storage. So we only extract the header of every packet in the flow, including packet header, IP header, and TCP/UDP header. In addition, we delete the IP address in the packet to avoid model overfitting [15, 25].

3.1.2 Twice Segmentation

1. Flow Segmentation

DL-based methods need to divide the continuous traffic into discrete units plurality according to a particular granularity [26]. Raw traffic P is a set containing the different size of packets, denoted as:

$$P = \{p^1, \ldots, p^i \ldots, p^{|P|}\} \tag{1}$$

where $|P|$ is the number of packets in P, p^i is the i-th packet in P, which is defined as:

$$p^i = (x^i, b^i, t^i) \tag{2}$$

where $i = 1, 2, \ldots, |P|$, $b^i \in (0, \infty)$, $t^i \in [0, \infty)$, x^i is the five-tuple (source IP, source port, destination IP, destination port, transport layer protocol) of the i-th packet, b^i is the byte length of the i-th packet and t_i is the start time of the i-th packet.

Raw traffic is first segmented by flow because it is frequently used in current traffic classification studies [6]. A flow F is a group of packets in P that have the same five-tuple, which is defined as:

$$F = \{p^1 = (x^1, b^1, t^1), \ldots, p^n = (x^n, b^n, t^n)\} \tag{3}$$

where $n \leq |P|$, it is the packet number of F.

2. Packet Segmentation

Actual network traffic usually exists massive smaller-size flow that is unrelated to the class of traffic such as SNMP, DNS, and ARP, affecting the effective classification of traffic. Owing to those larger-size flow are the main activities in the communication process that have less unrelated traffic, we propose a packet segmentation to dilute unrelated flows and increase the weight of valid flow. It splits flow continuously by setting the maximum number of packets in the flow F. G_i denotes the i-th traffic in F that is defined as:

$$G_i = \{p^1 = (x^1, b^1, t^1), \ldots, p^m = (x^m, b^m, t^m)\}$$
$$m = \begin{cases} |F| - \sum_{k=1}^{i-1} |G_k| & if\ |F| - C \cdot (i-1) < C \\ C & otherwise \end{cases} \tag{4}$$

where m is the number of the packet in G_i, and C is the maximum number of packets that is defined as:

$$C = \frac{L_{sample}}{max(L_{header})} \tag{5}$$

where L_{sample} denotes the byte length of a sample, and L_{header} denotes the byte length of the sum of packet header, IP header (deleting the 4-byte source IP address and the 4-byte destination IP address), TCP/UDP header. Noteworthy, Since the TCP header is 20 bytes and the UDP header is 8 bytes, we uniformly select the first 20 bytes of the TCP/UDP packet in order to preserve the maximum header information. Thus, the maximum byte length of the sum of header is 48. The reason for this setting is that we hope to make full use of G_i to predict the whole flow accurately. In our view, the more packets G_i has, the more representative it is. Thus, we make C reach the maximum.

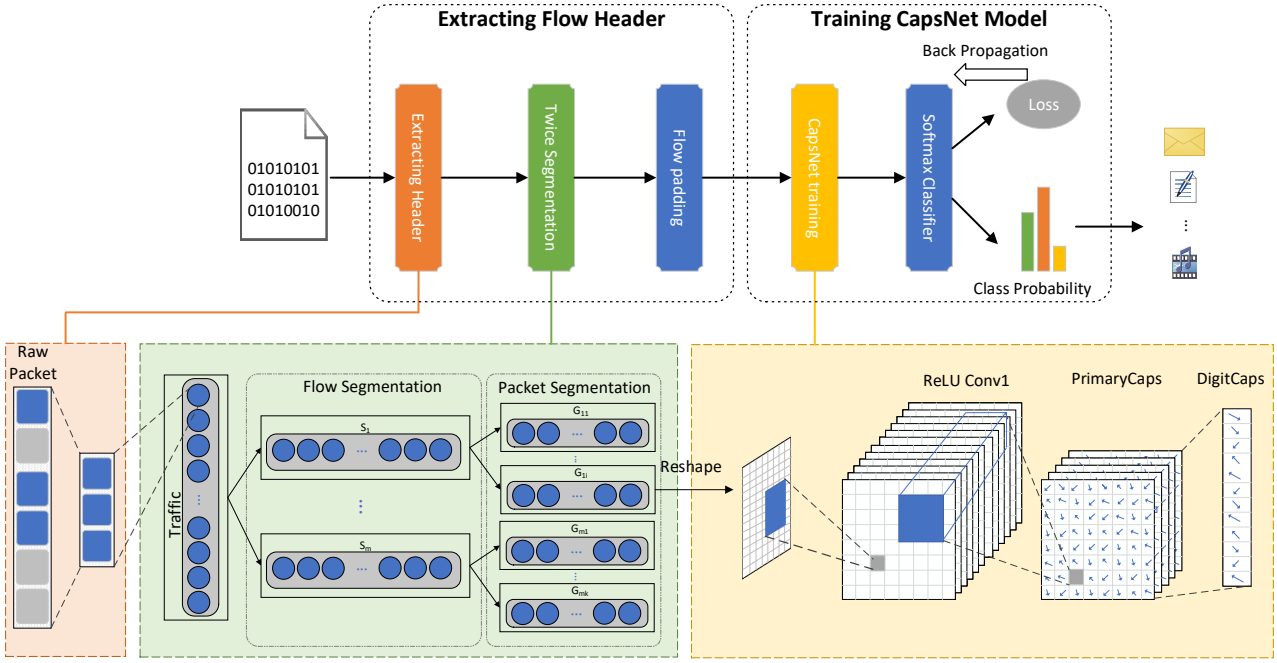


Fig. 2 Framework of the Only Header for encrypted traffic classification.

3.1.3 Flow Padding

Using neural networks to train data requires a fixed amount of input, so we have a uniform size of 784 bytes for the traffic of the above steps. When G_i is larger than 784 bytes, only the first 784 bytes are retained. Otherwise, the 0x00 is added in the end to complement it to 784 bytes. In addition, to make the traffic as a normative input to the following model, we reshape 784 bytes to 28*28 matrix.

3.2 Training CapsNet for Encrypted Classification

In this section, we design the training CapsNet model as another part of the Only Header. We use the CapsNet to classify the traffic matrices with the size of 28*28, which consists of convolution operation and dynamic routing. The model structure is shown in Figure 3. The input and output of CapsNet use vectors instead of scalars of traditional neural networks. The length of vectors indicates the probability of the encrypted traffic, and the direction indicates the attributes of the features such as size and position. In addition, compared to CNN, CapsNet no longer adopts pooling operations. It is well known that the pooling operation also discards some necessary information, including accurate location information, while reducing connection parameters and refining features.

3.2.1 Convolution Operation

CapsNet model reads traffic matrices via preprocessing mentioned above with the size of 28*28*1 that ranging from 0-255, so we first normalize traffic matrices to limit the value range to [0,1].

In the ReLU Conv1 layer, a convolution operation of stride 1 is performed on a traffic matrix using 256 convolution kernels with the size of 9*9 to generate 256 feature matrices of traffic with the size of 20*20.

Subsequently, the second convolutional layer (PrimaryCaps) is used as the input of the capsule to construct the tensor structure. Specifically, we perform 8 different weighted Con2d operations on 256 feature matrices of traffic and execute 32 convolution kernels with the size of 9*9 and a stride of 2 in each Con2d to finish convolution operation. Finally, 6*6*32 vectors with a dimension of 8 are generated. Each vector is a new capsule unit formed by 8 common convolution units. The length of the capsule indicates the probability of a class that traffic belongs to. The direction of the capsule indicates the attributes of traffic (location of fixed strings, the order between packets).

3.2.2 Dynamic Routing

The third layer of DigitCaps propagates and updates the input capsule. The capsule processing is divided into two steps: linear combination and routing. For the linear combination, the capsule output activity vector

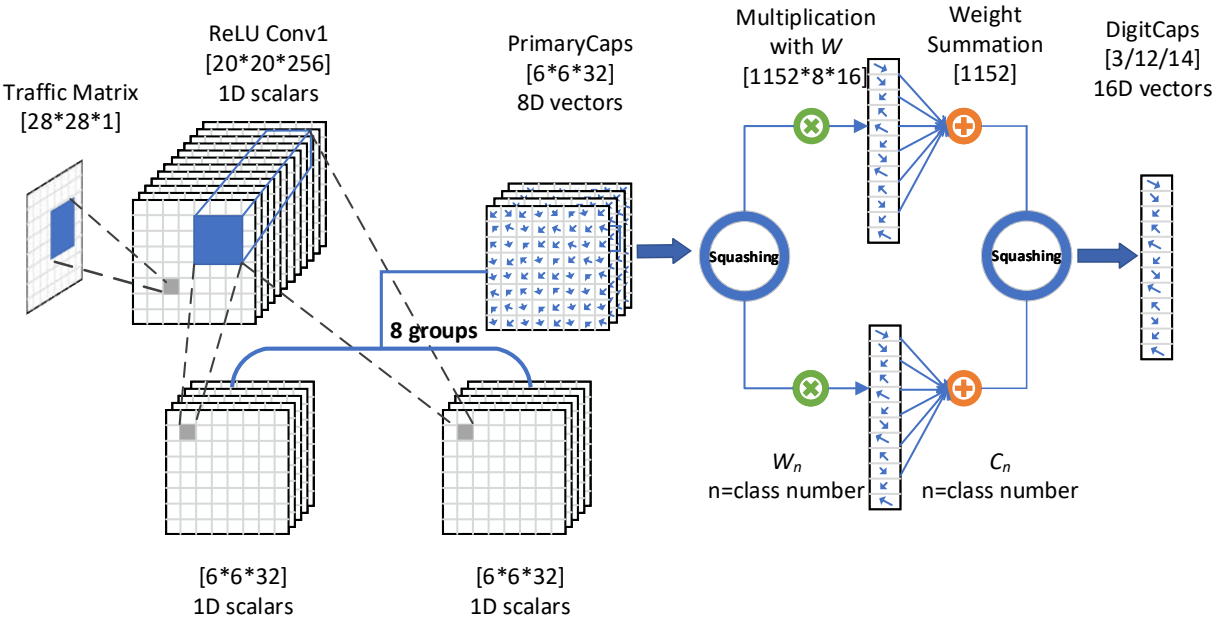


Fig. 3 Schematic diagram of traffic classification model based on Capsule.

of the lower layer u_i is multiplied by a weight matrix W_{ij} to obtain a prediction vector $\hat{u}_{j|i}$, and all inputs of the higher layer capsule s_j are weighted summations of the predicted vectors, given by

$$\begin{aligned} \hat{u} &= W_{ij} u_i \\ s_j &= \sum_i c_{ij} \hat{u}_{j|i} \end{aligned} \tag{6}$$

where c_{ij} is a coupling coefficient determined by iterative dynamic path.

For the dynamic routing mechanism, to find the most suitable path between the capsule's output and the next layer's input, c_{ij} in (6) is updated by

$$c_{ij} = \frac{exp\{b_{ij}\}}{\sum_k \exp\{b_{ik}\}} \tag{7}$$

where b_{ij} is the logarithmic prior probability of capsule i coupled to capsule j.

The length of one capsule output vector is between [0,1], indicating the probability of a certain class. Thus, a squashing function is used to compress vectors that is defined as follows:

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||} \tag{8}$$

where v_j is the output vector of capsule j and s_j is its total inputs.

W_{ij} and other convolution parameters of the entire network are updated by the loss function. Therefore, we use the margin loss commonly used in SVM as the loss function, defined as:

$$\begin{aligned} L_c = & T_c \max(0, m^+ - ||v_c||)^2 + \\ & \lambda(1 - T_c) \max(0, ||v_c|| - m^-)^2 \end{aligned} \tag{9}$$

where c is predicted class and T_c is an indication function that if c is correct, T_c equals 1, otherwise 0. m^+ is upper boundary of v_c while m^- is lower boundary, λ is regularization strength. We adopt reconstruction loss to avoid overfitting. Table 1 describes the main parameters of each layer in our model.

Table 1 The main parameters of CapsNet model.

Name	Operation	Input	Filter	Stride	Output
ReLU Conv1	Convolution	28*28	9*9	1	20*20*256
PrimaryCaps	Convolution	20*20*256	9*9	2	1152*8*1
DigitCaps	Multiplication	1152*8*1	—	—	1152*16*1
	Summation	1152*16*1	—	—	5/12*16*1
FC	Softmax	5/12*16*1	—	—	5/12*1*1
FC	Softmax	5/12*1*1	—	—	5/12

4 Experiment

4.1 Dataset

The most critical condition for training deep learning models is that there are a large number of represen-

Table 2 Number of samples in each class for encrypted traffic classification.

Regular Traffic		VPN Traffic		Tor Traffic	
Class	Total	Class	Total	Class	Total
Chat	7815	VPNChat	8820	TorChat	1295
Email	7940	VPNEmail	1580	TorEmail	10000
File	10000	VPNFile	10000	TorFile	10000
P2P	7565	VPNP2P	10000	TorP2P	10000
VoIP	10000	VPNVoIP	10000	TorVoIP	10000
Streaming	10000	VPNStreaming	10000	TorVideo	10000
				TorAudeo	10000
				TorBrowsing	10000

tative datasets. However, the lack of available datasets is an essential factor hindering traffic classification [25]. To demonstrate the effectiveness of the proposed method, we use the ISCX VPN-nonVPN dataset [8] and ISCX Tor-nonTor [14] to evaluate the Only Header. ISCX VPN-nonVPN dataset provides 150 raw traffic files, including 7 kinds of conventional encrypted pcap files (chat, streaming, etc.) and 7 kinds of VPN pcap files (VPNchat, VPNstreaming, etc.). ISCX Tor-nonTor dataset provides 85 raw traffic files, including 8 kinds of conventional encrypted pcap files (browsing, email, etc.) and 8 kinds of Tor pcap files (Torbrowsing, Toremail, etc.).

On the ISCX VPN-nonVPN dataset, the author labels 150 traffic files according to specific applications instead of marking them according to service, making some traffic files ambiguous. Particularly, browsing is HTTPS traffic generated when browsing or executing a task that contains a browser [8]. We are not sure some certain traffic files like hangoutVoIP belonging to browsing or belonging to Voip. Therefore, we decide to delete browsing and VPNbrowsing labels, changing 14 classes to 12 classes. In addition, because the non-Tor traffic in the ISCX Tor-nonTor dataset is derived from the ISCX VPN-nonVPN dataset, we only use Tor traffic. Finally, we get three encrypted traffic, including regular encrypted traffic, VPN traffic, and Tor traffic.

According to (5), the maximum number of packets in packet segmentation is set to 16. Application and the total number of samples are shown in Table 2.

4.2 Experimental Setup

In this paper, we use Python3, TensorFlow as software frameworks, which run on Ubuntu 16.04 64bit OS. The server is a DELL R720 with 16CPU cores and 128GB of memory. An Nvidia Corporation GM204GL GPU is used as the accelerator.

Table 3 Task description to evaluate our preprocessing method.

Exp	Description	Classification
1	Identification of regular, VPN and Tor	3 classification
2	Regular and VPN traffic classification	12 classification
3	Regular and Tor traffic classification	14 classification

4.3 Evaluation Metric

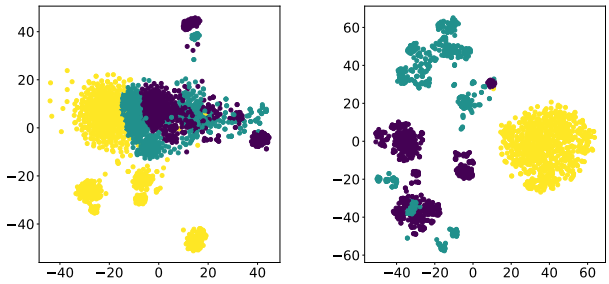
We use accuracy (*Acc*), precision (*Pre*), recall (*Rec*), and F-measure (F_1) metrics to evaluate our proposed methods, reflecting the ability of the method to identify network traffic. Accuracy is used to evaluate the overall effect of the method. Precision and recall reflect the recognition efficiency of the identification method in each class. F-measure is the evaluation index obtained by comprehensive precision and recall.

4.4 Traffic Preprocessing Evaluation

In extracting flow header, we observe that application payload can cause privacy risk and cannot be regarded as effective features in encrypted traffic classification. Therefore, we only extract the flow header to avoid privacy trouble and reduce data storage pressure. Moreover, we introduce the twice segmentation mechanism to split traffic. The segmentation mechanism performs flow segmentation and packet segmentation on the traffic. It can achieve the purposes of diluting the proportion of unrelated traffic and increasing effective traffic weight. In order to evaluate the above traffic preprocessing methods, we perform encrypted traffic identification, regular and VPN traffic classification, regular and Tor traffic classification on the ISCX VPN-nonVPN and ISCX Tor-nonTor datasets. The detailed task description is shown in Table 3.

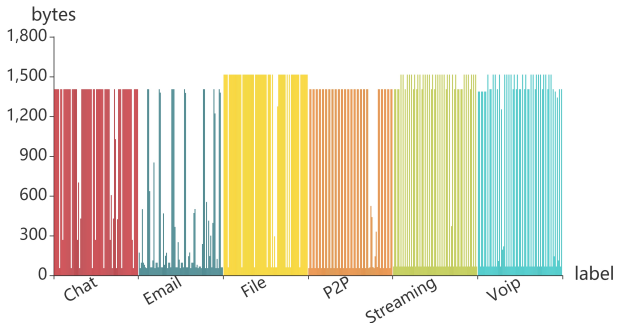
4.4.1 Analysis of the Flow Header

On the flow header analysisr, we perform T-SNE dimensionality reduction visualization on regular encrypted traffic, VPN traffic, and Tor traffic, and the results are shown in Figure 4. It can be seen that TSNE's dimensionality reduction visualization performs well, regardless of whether it contains application payload. However, in the visualization of all data, some sample points are not easy to separate in regular encrypted traffic and VPN traffic, while the visualization of flow header performs better. From an intuitive point of view, the flow header is more conducive to traffic classification.

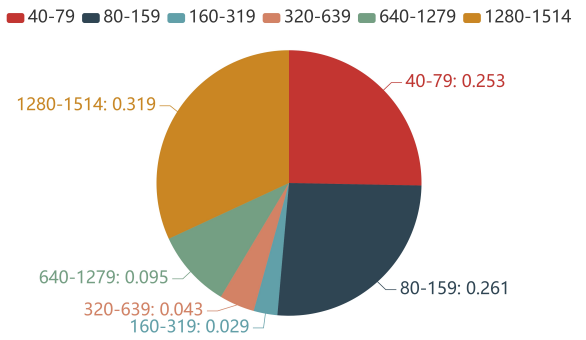


(a) T-SNE embedding of all data. (b) T-SNE embedding of the flow header.

Fig. 4 T-SNE embedding of the encrypted traffic in identification task. Purple denotes regular encrypted traffic, green denotes VPN traffic, and yellow denotes Tor traffic.



(a) Packet size distribution of each encrypted traffic class.



(b) Percentage of packet size in each interval.

Fig. 5 Packet size distribution of the three types of encrypted traffic (regular encrypted traffic, VPN traffic and Tor traffic).

Next, we analyze the impact of only the flow header on data storage. First, we count the packet size distribution of regular encrypted traffic, VPN traffic, and Tor traffic shown in Figure 5. Figure 5(a) depicts the packet size distribution for each class, with each class containing 100 randomly selected packets from the entire transmission conversation. Figure 5(b) depicts the proportion of packet sizes at different intervals for all transmission conversations in the six classes. It can be

Table 4 Size change of encrypted traffic on whether to remain application data (GB).

Class	All data	Flow header	Proportion
Regular	1.94	0.13	14.9:1
VPN	1.96	0.17	11.5:1
Tor	11.32	0.74	15.3:1

Table 5 Result on whether to remain application payload for encrypted traffic (%).

Exp	All data		Flow header	
	Acc	F_1	*Acc*	F_1
1	99.9	99.9	99.9	99.9
2	99.1	99.3	99.2	99.2
3	98.1	98.0	99.3	99.2

observed that except for the Email traffic, others are usually transmitted in large size packets (around 1500 bytes). In addition, the packets with 1280-1514 size account for the highest percentage of packets, taking up 31.9%, and the packets with more than 79 bytes occupy 74.7%. This indicates that most packets contain application payload. More intuitively, we count the size changes on whether to remain application payload of the three types of encrypted traffic, as shown in Table 4. It can be seen that all data is 11-15 times larger than the flow header. Therefore, if the application payload cannot generate effective features, extracting flow header will significantly reduce data storage pressure during the encrypted traffic classification process.

In order to verify the performance of the flow header, we perform 3 group experiments that the detail of tasks are shown in Table 3. According to Table 5, the flow header doesn't reduce the classification effect of encrypted traffic in the three tasks. On the contrary, in all tasks, the classification results of the Only Header are as well as the classification results of all data, even better in regular and Tor traffic classification (Exp 3). A large amount of randomized encrypted payload cannot be used as effective features for classification when the application payload is retained. Therefore, when the application payload is removed, the flow header has a regular field distribution, which is more conducive to mining spatial features and byte features between different classes. Therefore, This process can not only keep the accuracy of classification but also avoid privacy risk and greatly reduce data storage pressure.

4.4.2 Analysis of Twice Segmentation Mechanism

We propose the twice segmentation to increase the weight of effective traffic. In order to evaluate the twice seg-

Table 6 Result on whether to perform the twice segmentation for encrypted traffic (%).

Exp	Raw Flow		Twice Segmentation	
	Acc	F_1	*Acc*	F_1
1	99.9	99.9	99.9	99.9
2	97.2	95.4	99.2	99.2
3	98.4	97.7	99.3	99.2

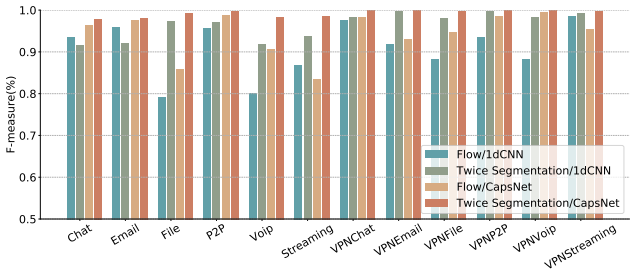


Fig. 6 F-measure about whether to conduct twice segmentation using 1dCNN and CapsNet.

mentation mechanism, we perform 3 group tasks mentioned in Table 3. Moreover, to verify that CapsNet is more suitable than 1dCNN in traffic classification tasks, we use both models for comparison in regular and VPN traffic classification (Exp 2). The results are shown in Table 6 and Figure 6. As Table 6 shows, in encrypted traffic identification (Exp 1), regardless of whether to perform the twice segmentation mechanism, both the accuracy and F-measure can reach 99.9%. In regular and VPN traffic classification (Exp 2), the twice segmentation mechanism improves 2.0% of the accuracy and 3.8% of the F-measure. In regular and Tor traffic classification (Exp 3), the twice segmentation mechanism increases the accuracy by 0.9% and the F-measure by 1.5% over raw flow.

Figure 6 describes the performance results of CapsNet and 1dCNN on regular and VPN traffic classification. In the 1dCNN model, we observe that 10 kinds of traffic (except Chat and Email classes) can achieve better results using twice segmentation. Besides, the F-measure of two classes, File and Voip, are less than 80% using flow. In contrast, twice segmentation improves the F-measure, making each class reaches more than 90%. In the CapsNet model, all kinds of traffic using twice segmentation are better than the traditional flow, and each of them is above 97%. Moreover, in the comparison of 1dCNN and CapsNet, most classes of traffic show CapsNet performs better than 1dCNN no matter whether to conduct twice segmentation. Compared to other combinations, the F-measure for each class achieves the best value with twice segmentation and CapsNet model.

In summary, our proposed twice segmentation mechanism has shown better experimental results in encrypted traffic classification. In addition, no matter whether we execute packet segmentation or not, CapsNet shows higher accuracy and F-measure than 1dCNN.

4.5 Baseline Experiments Comparison

In this subsection, we perform three experiments mentioned in Table 3 to evaluate the Only Header and compare the results with baseline methods on ISCX VPN-nonVPN and ISCX Tor-nonTor datasets. Owing to the accuracy and F-measure for encrypted traffic identification reach 99.9%, and the baseline methods also can achieve 99% accuracy, we no longer analyze and compare this task in detail. The results of other tasks are as follows, including regular and VPN traffic classification, regular and Tor traffic classification.

4.5.1 Comparison on Regular and VPN Traffic Classification

We compare the Only Header to the following baseline methods for regular and VPN traffic classification.

- **1dCNN [25]** utilizes the first 784 bytes of raw flow as the input of the 1dCNN model. 1dCNN performs better than traditional CNN on encrypted traffic classification.
- **Deep Packet [15]** converts IP packets, including IP header and IP payload, into a 1500-byte vector and then uses 1dCNN and SAE to learn the deep features of the packet vector to realize encrypted traffic classification.
- **Attention based LSTM and HAN [29]** regards the flow (the first ten packets and the first 1500 bytes of each packet) as a time sequence and constructs the attention based LSTM and hierarchical attention network (HAN) to learn flow time features.
- **CNN-LSTM [33]** combines CNN and LSTM. It employs CNN to learn the spatial features of the single packet and utilizes LSTM to learn the time sequence features of three consecutive packets in the flow.

In order to evaluate and compare the effectiveness of the Only Header for regular and VPN traffic classification, we adopt the above preprocessing to experiment on raw traffic collected in the ISCX VPN-nonVPN dataset. The experiment shows that the precision and recall of each class are as high as 97% as shown in Table 7. The F-measure of 9 classes (except Chat, Email,

Table 7 Classification result of the Only Header for regular and VPN traffic (%).

Class	*Pre*	*Rec*	F_1
Chat	97.3	98.3	97.8
Email	97.6	98.9	98.1
File	99.5	99.6	99.6
P2P	98.8	99.7	99.2
Voip	99.3	97.0	98.1
Streaming	99.2	98.9	99.1
VPNChat	99.2	99.8	99.5
VPNEmail	99.7	99.7	99.7
VPNFile	99.2	99.3	99.3
VPNP2P	99.8	99.9	99.9
VPNVoip	99.8	99.7	99.7
VPNStreaming	99.9	99.5	99.7
Accuracy		**99.2**	

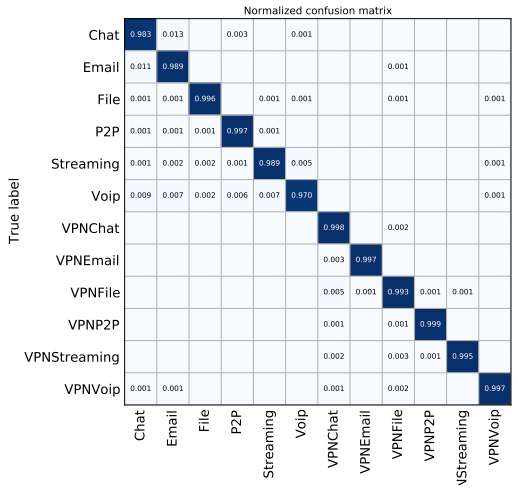


Fig. 7 Confusion matrix of the Only Header for regular and VPN traffic.

Table 8 Result of the Only Header compared with baseline methods for regular and VPN traffic classification (%).

Method	Model	Input	*Acc*
Only Header	CapsNet	flow header	99.2
1dCNN [25]	1dCNN	flow	86.6
Deep Packet [15]	SAE	packet	88.2
	1dCNN	packet	89.8
Attention based LSTM and HAN [29]	Attention based LSTM	flow	91.2
	HAN	flow	89.8
CNN-LSTM [33]	CNN and LSTM	flow	92

and Voip) reach 99%. In addition, the F-measure of VPN traffic classes are better than regular encrypted traffic, indicating that Only Header performs especially outstanding in VPN traffic classification. The confusion matrix of the Only Header with rows normalized for regular and VPN traffic classification is shown in Fig. 7. As the figure shows, all of the traffic classes on the diagonal show the deeper blue color, indicating the effective classification ability of the Only Header for regular and VPN traffic classification.

Compared with the baseline methods for encrypted traffic service classification on the ISCX VPN-nonVPN dataset that use deep learning as well, Table 8 reports that accuracy is higher 7.2% than CNN-LSTM in [33] that is the best method to our best knowledge. In a word, the Only Header performs better and achieves the standard of practical application.

Table 9 Classification result of the Only Header for regular and Tor traffic (%).

Class	*Pre*	*Rec*	F_1
Chat	96.8	97.2	97.0
Email	97.4	98.1	97.7
File	99.6	99.6	99.6
P2P	98.4	99.9	99.1
Voip	99.4	96.9	98.1
Streaming	98.9	99.3	99.1
TorChat	99.2	100.0	99.6
TorEmail	100.0	100.0	100.0
TorFile	99.9	99.8	99.9
TorP2P	99.7	99.8	99.7
TorVoIP	100.0	100.0	100.0
TorVideo	99.9	100.0	100.0
TorAudio	99.8	99.6	99.7
TorBrowsing	99.6	99.7	99.6
Accuracy		**99.3**	

4.5.2 Comparison on regular and Tor Traffic Classification

We compare Only Header to the following baseline methods for regular and Tor traffic classification.

- **C4.5 [14]** presents a time analysis on Tor traffic flows and proposes a series of features according to the time sequence of flow, and then it utilizes C4.5 to learn the flow features on Tor traffic classification.
- **FlowPic [21]** creates the image according to the packet sizes and packet arrival times of flow and then uses CNN to learn the spatial features of the image to build an encrypted traffic classification model.

Tor traffic only supports the encrypted links and TCP flow over the Internet. It is complicated to trace and analyze its traffic [14]. We use regular encrypted

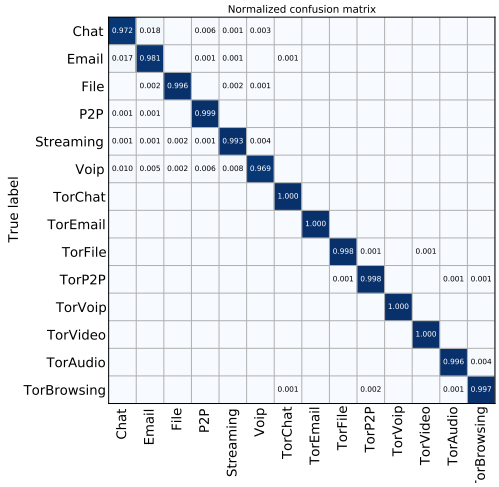


Fig. 8 Confusion matrix of the Only Header for regular and Tor traffic.

Table 10 Result of the Only Header compared with baseline methods for regular and Tor traffic classification (%).

Method	Model	Input	*Acc*
Only Header	CapsNet	flow header	99.3
C4.5 [14]	C4.5	flow features	84
FlowPic [21]	FlowPic	flow time sequence	85.7

traffic in the ISCX VPN-nonVPN dataset and Tor traffic in the ISCX Tor-nonTor dataset to implement regular and Tor traffic classification. As shown in Table 9, we observe that all kinds of traffic can reach more than 99% except Chat and Email. The accuracy of classification is 99.3%. Besides, the precision and recall of both TorEmail and TorVoIP are 100%. The confusion matrix of the Only Header with rows normalized for regular and Tor traffic classification is shown in Fig. 8. As the figure shows, all of the traffic classes on the diagonal show the deeper blue color, indicating the effective classification ability of the Only Header for regular and Tor traffic classification.

On the comparison of other approaches, it is shown in Table 10 that the accuracy of the Only Header is higher 15.3% and 13.6% than baseline methods. Therefore, the Only Header makes a great improvement for Tor traffic classification.

5 Conclusion

Based on the analysis of the current research on encrypted traffic classification, this paper proposes a reliable encrypted traffic classification framework without privacy risk. It utilizes CapsNet model to learn the spatial and byte features of the flow header, which avoids privacy troubles and reduces data storage pressure. Besides, the Only Header is more suitable for encrypted traffic classification tasks than others for the reason that it takes into account the location of fixed strings and the order between packets. Meanwhile, the Only Header increases the effective traffic weight by a twice segmentation mechanism, which exhibits higher accuracy than traditional traffic representation such as packet and flow. The experimental results show our study yields significant improvements against the state-of-the-art methods on ISCX VPN-nonVPN and ISCX Tor-nonTor traffic dataset.

Acknowledgements

This research is supported by National Key Research and Development Program of China (No.2019QY1300), and CCF-NSFOCUS Kun-Peng Scientific Research Foundation (No.2020010), and the Strategic Priority Research Program of Chinese Academy of Sciences (No.XD C02040100). This work is also supported by the Program of Key Laboratory of Network Assessment Technology, the Chinese Academy of Sciences, Program of Beijing Key Laboratory of Network Security and Protection Technology.

Declarations

5.1 Compliance with Ethical Standards statements

5.1.1 Ethical approval

1. Disclosure of potential conflicts of interest
   - The authors have no relevant financial or non-financial interests to disclose.
   - The authors have no conflicts of interest to declare that are relevant to the content of this article.
   - All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.
   - The authors have no financial or proprietary interests in any material discussed in this article.
2. Research involving Human Participants and/or Animals

   Not applicable
3. Informed consent

   Not applicable

5.1.2 Funding

This research is supported by National Key Research and Development Program of China (No.2019QY1300), and CCF-NSFOCUS Kun-Peng Scientific Research Foundation (No.2020010), and the Strategic Priority Research Program of Chinese Academy of Sciences (No.XD C02040100). This work is also supported by the Program of Key Laboratory of Network Assessment Technology, the Chinese Academy of Sciences, Program of Beijing Key Laboratory of Network Security and Protection Technology.

5.1.3 Conflicts of interest

- The authors have no relevant financial or non-financial interests to disclose.
- The authors have no conflicts of interest to declare that are relevant to the content of this article.
- All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.
- The authors have no financial or proprietary interests in any material discussed in this article.

5.1.4 Availability of data and material

Data of this study are from public datasets which publish in [8, 14] and are available on `https://www.unb.ca/cic/datasets/index.html`.

5.1.5 Code availability

The code cannot be shared at this time as the code also forms part of an ongoing study.

5.2 Authors' contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Susu Cui, Jian Liu and Cong Dong. The first draft of the manuscript was written by Susu Cui and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

1. Arndt, D.J., Zincir-Heywood, A.N.: A comparison of three machine learning techniques for encrypted network traffic analysis. In: 2011 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2011, Paris, France, April 15, 2011, pp. 107–114. IEEE (2011)
2. Bar-Yanai, R., Langberg, M., Peleg, D., Roditty, L.: Realtime classification for encrypted traffic. In: P. Festa (ed.) Experimental Algorithms, 9th International Symposium, SEA 2010, Ischia Island, Naples, Italy, May 20-22, 2010. Proceedings, *Lecture Notes in Computer Science*, vol. 6049, pp. 373–385. Springer (2010)
3. Bonfiglio, D., Mellia, M., Meo, M., Rossi, D., Tofanelli, P.: Revealing skype traffic: when randomness plays with you. In: J. Murai, K. Cho (eds.) Proceedings of the ACM SIGCOMM 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Kyoto, Japan, August 27-31, 2007, pp. 37–48. ACM (2007)
4. Cao, Z., Xiong, G., Zhao, Y., Li, Z., Guo, L.: A survey on encrypted traffic classification. In: L. Batten, G. Li, W. Niu, M. Warren (eds.) Applications and Techniques in Information Security, pp. 73–81. Springer Berlin Heidelberg, Berlin, Heidelberg (2014)
5. Cui, S., Jiang, B., Cai, Z., Lu, Z., Liu, S., Liu, J.: A session-packets-based encrypted traffic classification using capsule neural networks. In: Z. Xiao, L.T. Yang, P. Balaji, T. Li, K. Li, A.Y. Zomaya (eds.) 21st IEEE International Conference on High Performance Computing and Communications; 17th IEEE International Conference on Smart City; 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019, Zhangjiajie, China, August 10-12, 2019, pp. 429–436. IEEE (2019)
6. Dainotti, A., Pescapè, A., Claffy, K.C.: Issues and future directions in traffic classification. IEEE Network **26**(1), 35–40 (2012)
7. Dewang Chen, Shixin Li, Lijun Pei: A classification algorithm on traffic state of expressway link based on ensemble fuzzy classifier. In: 2010 8th World Congress on Intelligent Control and Automation, pp. 330–334 (2010)
8. Draper-Gil, G., Lashkari, A.H., Mamun, M.S.I., Ghorbani, A.A.: Characterization of encrypted and VPN traffic using time-related features. In: O. Camp, S. Furnell, P. Mori (eds.) Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISSP 2016, Rome, Italy, February 19-21, 2016, pp. 407–414. SciTePress (2016)
9. Fu, Y., Xiong, H., Lu, X., Yang, J., Chen, C.: Service usage classification with encrypted internet traffic in mobile

messaging apps. IEEE Transactions on Mobile Computing **15**(11), 2851–2864 (2016)
10. Gai, K., Qiu, M., Zhao, H.: Privacy-preserving data encryption strategy for big data in mobile cloud computing. IEEE Transactions on Big Data pp. 1–1 (2017)
11. Goodfellow, I.J., Bengio, Y., Courville, A.C.: Deep Learning. Adaptive computation and machine learning. MIT Press (2016)
12. Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.C.: Transport layer identification of P2P traffic. In: A. Lombardo, J.F. Kurose (eds.) Proceedings of the 4th ACM SIGCOMM Internet Measurement Conference, IMC 2004, Taormina, Sicily, Italy, October 25-27, 2004, pp. 121–134. ACM (2004)
13. Korczynski, M., Duda, A.: Markov chain fingerprinting to classify encrypted traffic. In: 2014 IEEE Conference on Computer Communications, INFOCOM 2014, Toronto, Canada, April 27 - May 2, 2014, pp. 781–789. IEEE (2014)
14. Lashkari, A.H., Draper-Gil, G., Mamun, M.S.I., Ghorbani, A.A.: Characterization of tor traffic using time based features. In: P. Mori, S. Furnell, O. Camp (eds.) Proceedings of the 3rd International Conference on Information Systems Security and Privacy, ICISSP 2017, Porto, Portugal, February 19-21, 2017, pp. 253–262. SciTePress (2017)
15. Lotfollahi, M., Siavoshani, M.J., Zade, R.S.H., Saberian, M.: Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput. **24**(3), 1999–2012 (2020)
16. Madhukar, A., Williamson, C.L.: A longitudinal study of P2P traffic classification. In: 14th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 2006), 11-14 September 2006, Monterey, California, USA, pp. 179–188. IEEE Computer Society (2006)
17. Moore, A., Zuev, D., Crogan, M.: Discriminators for use in flow-based classification. Tech. rep. (2013)
18. Moore, A.W., Papagiannaki, K.: Toward the accurate identification of network applications. In: C. Dovrolis (ed.) Passive and Active Network Measurement, 6th International Workshop, PAM 2005, Boston, MA, USA, March 31 - April 1, 2005, Proceedings, *Lecture Notes in Computer Science*, vol. 3431, pp. 41–54. Springer (2005)
19. Okada, Y., Ata, S., Nakamura, N., Nakahira, Y., Oka, I.: Application identification from encrypted traffic based on characteristic changes by encryption. In: 2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), pp. 1–6 (2011)
20. Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. In: S.I. Feldman, M. Uretsky, M. Najork, C.E. Wills (eds.) Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17-20, 2004, pp. 512–521. ACM (2004)
21. Shapira, T., Shavitt, Y.: Flowpic: Encrypted internet traffic classification is as easy as image recognition. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2019, Paris, France, April 29 - May 2, 2019, pp. 680–687. IEEE (2019)
22. Taylor, V., Nurse, J.R.C., Hodges, D.: Android apps and privacy risks : what attackers can learn by sniffing mobile device traffic (2014)
23. Velan, P., Cermák, M., Celeda, P., Drasar, M.: A survey of methods for encrypted traffic classification and analysis. Int. Journal of Network Management **25**(5), 355–374 (2015)
24. Viegas, E., Santin, A.O., Neves, N.F., Bessani, A., Abreu, V.: A resilient stream learning intrusion detection mechanism for real-time analysis of network traffic. In: 2017 IEEE Global Communications Conference, GLOBECOM 2017, Singapore, December 4-8, 2017, pp. 1–6. IEEE (2017)
25. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics, ISI 2017, Beijing, China, July 22-24, 2017, pp. 43–48. IEEE (2017)
26. Wang, W., Zhu, M., Zeng, X., Ye, X., Sheng, Y.: Malware traffic classification using convolutional neural network for representation learning. In: 2017 International Conference on Information Networking, ICOIN 2017, Da Nang, Vietnam, January 11-13, 2017, pp. 712–717. IEEE (2017)
27. Wang, Z.: The applications of deep learning on traffic identification. BlackHat USA **24**(11), 1–10 (2015)
28. Yao, H., Gao, P., Wang, J., Zhang, P., Jiang, C., Han, Z.: Capsule network assisted iot traffic classification mechanism for smart cities. IEEE Internet Things J. **6**(5), 7515–7525 (2019)
29. Yao, H., Liu, C., Zhang, P., Wu, S., Jiang, C., Yu, S.: Identification of encrypted traffic through attention mechanism based long short term memory. IEEE Transactions on Big Data pp. 1–1 (2019)
30. Yeganeh, S.H., Eftekhar, M., Ganjali, Y., Keralapura, R., Nucci, A.: CUTE: traffic classification using terms. In: 21st International Conference on Computer Communications and Networks, ICCCN 2012, Munich, Germany, July 30 - August 2, 2012, pp. 1–9. IEEE (2012)
31. Zeng, Y., Gu, H., Wei, W., Guo, Y.: $deep-full-range$: A deep learning based network encrypted traffic classifi-

cation and intrusion detection framework. IEEE Access **7**, 45182–45190 (2019)
32. Zhang, M., Zhang, H., Zhang, B., Lu, G.: Encrypted traffic classification based on an improved clustering algorithm. In: Y. Yuan, X. Wu, Y. Lu (eds.) Trustworthy Computing and Services - International Conference, ISCTCS 2012, Beijing, China, May 28 - June 2, 2012, Revised Selected Papers, *Communications in Computer and Information Science*, vol. 320, pp. 124–131. Springer (2012)
33. Zou, Z., Ge, J., Zheng, H., Wu, Y., Han, C., Yao, Z.: Encrypted traffic classification with a convolutional long short-term memory neural network. In: 20th IEEE International Conference on High Performance Computing and Communications; 16th IEEE International Conference on Smart City; 4th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018, Exeter, United Kingdom, June 28-30, 2018, pp. 329–334. IEEE (2018)

Susu Cui received the B.S. degree from Nanchang University in 2019. She is currently pursuing the Ph.D. degree with the Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. Her research interests include encrypted traffic analysis and network security.

Jian Liu received the Ph.D. degree in Chinese Academy of Sciences in 2005. He is an associate professor at the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include software and network security, system security.

Cong Dong received the B.S. degree from Tianjin University in 2017. He is currently pursuing the Ph.D. degree with the Institute of Information Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include machine learning and network security.

Zhigang Lu received the Ph.D. degree in Chinese Academy of Sciences in 2010. He is an professor at the Institute of Information Engineering. His research interests include network security and network situational awareness.

Dan Du received the Master degree in Chinese Academy of Sciences in 2016. She is an engineer at the Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include network security, network situational awareness and mobile security.