Only Header: a reliable encrypted traffic classification framework without privacy risk

Susu Cui¹,² · Jian Liu¹,² · Cong Dong¹,² · Zhigang Lu¹,² · Dan Du¹

Accepted: 14 July 2022 / Published online: 3 October 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

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 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

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 (Viegas et al. 2017). 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 (Shen et al. 2021).

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 (Gai et al. 2017). 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. 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 do not work (Cao et al. 2014).

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 (Wang et al. 2017; Lotfollahi et al. 2020; Yao et al. 2022; Shapira and Shavitt 2019; Zou et al. 2018). However, the above studies directly touch the application payload, which can easily cause certain privacy troubles (Taylor et al. 2014). 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 (Shen et al. 2017; Yao et al. 2022; Shapira and Shavitt 2019; Shen et al. 2021). However, these methods are greatly affected by the environment and user habits (Fu et al. 2016). 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 Fig. 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 (Cui et al. 2019).

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 VPNnonVPN 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 (Dainotti et al. 2012; Velan et al. 2015). However, with the widespread application of encrypted traffic, port-based methods (Dainotti et al. 2012; Karagiannis et al. 2004; Moore and Papagiannaki 2005; Madhukar and Williamson 2006) and deep packet inspection (DPI) methods (Chen et al. 2010; Yeganeh et al. 2012; Sen et al. 2004; Bonfiglio et al. 2007; Korczynski and Duda 2014) 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. (2013) propose almost 250 flow or packet features for encrypted classification.
Okada et al. (2011) 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 (Velan et al. 2015). 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. (2011) 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 and Zincir-Heywood (2011) 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-Yanaei et al. (2010) 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. (2012) 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 (Xu et al. 2021). It can automatically select features from the raw traffic during training instead of extracting features manually (Goodfellow et al. 2016). 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 (2015) extract the first 1000 bytes of TCP flow and use a stacked autoencoder (SAE) to achieve encrypted protocol classification. Wang et al. (2017) 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. (2020) 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. (2018) 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 (Yao et al. 2019; Zeng et al. 2019; Cui et al. 2019). It can be seen that these methods are based on extracting the first $N$ bytes 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. (2022) regard encrypted network flow as a time sequence and build an attention-based LSTM model to learn the flow’s sequence features. Shapira and Shavitt (2019) 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. (2014) 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 is 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 (Fu et al. 2016). 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 Fig. 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 (Wang et al. 2017; Lotfollahi et al. 2020). This is due to the fact that when datasets are collected in a limited environment with a limited number of hosts collecting traffic, it is highly likely that IP address is an important feature to distinguish between different classes. However, in a real environment, the IP address is a numerical label that is connected to a computer network that uses the internet protocol for communication. Thus, it cannot be used as a feature to distinguish traffic classes.

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 (Wang et al. 2017). Raw traffic $P$ is a set containing the different size of packets, denoted as:

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

(1)

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

$$p^i = (x^i, b^i, t^i)$$

(2)

where $i = 1, 2, \ldots, |P|$, $b^i \in (0, \infty)$, $t^i \in [0, \infty)$, $x^i$ is the five-tuple traffic (source IP, source port, destination IP, destination port, protocol).
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 (Dainotti et al. 2012). A flow $F$ is a group of packets in $P$ that have the same five-tuple. The flow in this paper refers specifically to bi-directional flow, that is, the source IP address and source port can be interchanged with the destination IP and destination port. $F$ is defined as:

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

where $n \leq |P|$, and 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:

$$G_i = \{ p^1 = (x^1, b^1, t^1), \ldots, p^m = (x^m, b^m, t^m) \}$$

$$m = \begin{cases} |F| - \sum_{k=1}^{i-1} |G_k| & \text{if } |F| - C \cdot (i-1) < C \\ C & \text{otherwise} \end{cases}$$

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

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

where $L_{\text{sample}}$ denotes the byte length of a sample, and $L_{\text{header}}$ 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.

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. As in Wang et al. (2017); Lotfollahi et al. (2020); Yao et al. (2022), we use 0x00 as the padding value. Using 0x00 value does not change the parameters of the neural network. Therefore, it does not introduce bias to the classification results. When $G_i$ is larger than 784 bytes, only the first 784 bytes are retained. Otherwise, the $0 \times 00$ 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 Fig. 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 to 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 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

$$\hat{u} = W_{ij}u_i$$
$$s_j = \sum_i c_{ij}\hat{u}_{j|i}$$  \hspace{1cm} (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

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{jk})}$$ \hspace{1cm} (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:

$$v_j = \frac{||s_j||^2}{1 + ||s_j||^2} \frac{s_j}{||s_j||}$$ \hspace{1cm} (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:

$$L_c = T_c \max(0, m^+ - ||v_c||)^2 + \lambda (1 - T_c) \max(0, ||v_c|| - m^-)^2$$ \hspace{1cm} (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, $\lambda$ is regularization strength. We adopt reconstruction loss to avoid overfitting.

Hyperparameter tuning is essential to achieving the best classification performance. We train our model using Adam algorithm with a batch size of 128 examples. The learning rate is set to 0.001, and the number of learning epochs is set to 100 for the model training. Moreover, Table 1 describes the main parameters of each layer in our model. We also apply threefold cross-validation on the datasets to validate the results.

4 Experiment

4.1 Dataset

The most critical condition for training deep learning models is that there are a large number of representative datasets. However, the lack of available datasets is an essential factor hindering traffic classification (Wang et al. 2017). To demonstrate the effectiveness of the proposed method, we use the ISCX VPN-nonVPN dataset (Draper-Gil et al. 2016) and ISCX Tor-nonTor (Lashkari et al. 2017) 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 conven-
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    | Summation     | 1152*16*1   |        |        | 5/12*16*1         |
| FC           | Softmax       | 5/12*16*1   |        |        | 5/12*16*1         |
| FC           | Softmax       | 5/12*1*1    |        |        | 5/12              |

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

| Class        | Total | Class        | Total | Class        | Total |
|--------------|-------|--------------|-------|--------------|-------|
| Chat         | 7815  | VPNChat      | 8820  | TorChat      | 1295  |
| Email        | 7940  | VPNEmail     | 1580  | TorEmail     | 10,000|
| File         | 10,000| VPNFile      | 10,000| TorFile      | 10,000|
| P2P          | 7565  | VPNP2P       | 10,000| TorP2P       | 10,000|
| VoIP         | 10,000| VPNVoIP      | 10,000| TorVoIP      | 10,000|
| Streaming    | 10,000| VPNStreaming | 10,000| TorVideo     | 10,000|
|              |       |              |       | TorAudeo     | 10,000|
|              |       |              |       | TorBrowsing  | 10,000|

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.
### Table 3: Task description to evaluate our preprocessing method

| Exp | Description                     | Classification |
|-----|---------------------------------|----------------|
| 1   | Identification of regular, VPN and Tor traffic | 3 classification |
| 2   | Regular and VPN traffic classification | 12 classification |
| 3   | Regular and Tor traffic classification | 14 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) T-SNE embedding of all data, (b) T-SNE embedding of the flow header.

#### 4.4.1 Analysis of the flow header

On the flow header analysis, we perform T-SNE dimensionality reduction visualization on regular encrypted traffic, VPN traffic, and Tor traffic, and the results are shown in Fig. 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. Due to the encryption algorithm, application payload is encrypted as randomized ciphertexts. When all data of flow are visualized by TSNE embedding, there is no difference in the payloads between the three categories. Thus, it will be difficult to aggregate into three clusters. When only the flow header is used, it can be easily aggregated into three clusters. Because the flow header is not encrypted and its value is strongly correlated with the network environment and application type.

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 Fig. 5. Figure 5a depicts the packet size distribution for each class, with each class containing 100 randomly selected packets from the entire transmission conversation. Figure 5b depicts the proportion of packet sizes at different intervals for all transmission conversations in the six classes. It can be 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 are 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 does not 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 segmentation 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 Fig. 6. In Table 6 and Fig. 6, flow represents the whole flow as the identification object for traffic classification, which means only perform the flow segmentation in the twice segmentation. And the twice segmentation means the traffic after flow segmentation and packet segmentation as the identification object for classification. 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
Table 7  Classification result of the Only Header for regular and VPN traffic (%)

| Class          | Pre | Rec | F1  |
|----------------|-----|-----|-----|
| 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| 97.7| 99.7|
| VPNFile        | 99.2| 99.3| 99.3|
| VPNP2P         | 99.8| 99.9| 99.9|
| VPNV0ip        | 99.8| 99.7| 99.7|
| VPNStreaming   | 99.9| 99.5| 99.7|
| Accuracy       | 99.2|

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** Wang et al. (2017) 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** Lotfollahi et al. (2020) 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-LSTM and HAN** Yao et al. (2022) 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** Zou et al. (2018) 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, and Voip) reaches 99%. In addition, the F-measure of VPN traffic classes is 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 and F-measure are higher 7.2% and 6.2% than CNN-LSTM in Zou et al. (2018) that is the best method to our best knowledge. In a word, the Only Header performs better and achieves the standard of practical application.

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** Lashkari et al. (2017) 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** Shapira and Shavitt (2019) creates the image according to the packet sizes and packet arrival times of
Table 8 Result of the Only Header compared with baseline methods for regular and VPN traffic classification (%)

| Method                      | Model       | Input      | Acc  | F1   |
|-----------------------------|-------------|------------|------|------|
| Only Header                 | CapsNet     | Flow header| 99.2 | 99.2 |
| 1dCNN Wang et al. (2017)    | 1dCNN       | Flow       | 86.6 | 89.7 |
| Deep Packet Lotfollahi et al. (2020) | SAE         | Packet     | 88.2 | 92   |
|                             | 1dCNN       | packet     | 89.8 | 93   |
| Attention-LSTM and HAN Yao et al. (2022) | Attention-LSTM | Flow     | 91.2 | 88.6 |
| CNN-LSTM Zou et al. (2018)  | CNN and LSTM| Flow       | 89.8 | 84.8 |

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

| Class       | Pre  | Rec  | F1   |
|-------------|------|------|------|
| 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 |      |      |

Tor traffic only supports the encrypted links and TCP flow over the Internet. It is complicated to trace and analyze its traffic (Lashkari et al. 2017). We use regular encrypted 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 and the F-measure is higher 15.2% and 34.27% 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.

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by SC, JL and CD. The first draft of the manuscript was written by SC, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This research is supported by National Key Research and Development Program of China (No. 2019QY1300), CCF-NSFOCUS Kun-Peng Scientific Research Foundation (No. 2020010), and the Strategic Priority Research Program of Chinese Academy of Sciences (No. XDC02040100). This work is partially supported by NSFC (No. 61902376). 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.

Availability of data and materials Data of this study are from public datasets published in Draper-Gil et al. (2016); Lashkari et al. (2017) and are available on https://www.unb.ca/cic/datasets/index.html.

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

Declarations

Conflict of interests 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.

Research involving human participants and/or animals Not applicable.

Informed consent Not applicable.

References

Arndt DJ, Zincir-Heywood AN (2011) 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
Bar-Yanai R, Langberg M, Peleg D, Roditty L (2010) Realtime classification for encrypted traffic. In: Festa P (ed) Experimental algorithms, 9th international symposium, SEA 2010, Ischia Island, Naples, Italy, May 20–22, 2010. Proceedings, lecture notes in computer science. Springer, Berlin, vol 6049, pp 373–385
Bonfiglio D, Mellia M, Meo M, Rossi D, Tofanelli P (2007) Revealing skype traffic: when randomness plays with you. In: Murai J, Cho K (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
Cao Z, Xiong G, Zhao Y, Li Z, Guo L (2014) A survey on encrypted traffic classification. In: Batten L, Li G, Niu W, Warren M (eds) Applications and techniques in information security. Springer, Berlin, pp 73–81
Chen D, Li S, Pei L (2010) 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
Cui S, Jiang B, Cai Z, Lu Z, Liu S, Liu J (2019) A session-packets-based encrypted traffic classification using capsule neural networks. In: Xiao Z, Yang LT, Balaji P, Li T, Li K, Zomaya AY (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
Dainotti A, Pescape A, Claffy KC (2012) Issues and future directions in traffic classification. IEEE Netw 26(1):35–40
Draper-Gil G, Lashkari AH, Mamun M, Ghorbani AA (2016) Characterization of encrypted and VPN traffic using time-related features. In: Camp O, Furnell S, Mori P (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
Fu Y, Xiong H, Lu X, Yang J, Chen C (2016) Service usage classification with encrypted internet traffic in mobile messaging apps. IEEE Trans Mobile Comput 15(1):2851–2864
Gai K, Qiu M, Zhao H (2017) Privacy-preserving data encryption strategy for big data in mobile cloud computing. IEEE Trans Big Data 7(4):678–688
Goodfellow IJ, Bengio Y, Courville AC (2016) Deep Learning. MIT Press, Adaptive computation and machine learning
Karagiannis T, Broido A, Faloutsos M, Claffy KC (2004) Transport layer identification of P2P traffic. In: Lombardo A, Kurose JF (eds) Proceedings of the 4th ACM SIGCOMM internet measurement conference, IMC 2004, Taormina, Sicily, Italy, October 25–27, 2004. ACM, pp 121–134
Korcynski M, Duda A (2014) Markov chain fingerprinting to classify encrypted traffic. In: 2014 IEEE conference on computer communications, INFOCOM 2014, Toronto, Canada, April 27–May 2, 2014. IEEE, pp 781–789
Lashkari AH, Draper-Gil G, Mamun M, Ghorbani AA (2017) Characterization of tor traffic using time based features. In: Mori P, Furnell S, Camp O (eds) Proceedings of the 3rd international conference on information systems security and privacy, ICISSP 2017, Porto, Portugal, February 19–21, 2017. SciTePress, pp 253–262
Lotfollahi M, Siavoshani MJ, Zade RSH, Saberian M (2020) Deep packet: a novel approach for encrypted traffic classification using deep learning. Soft Comput. 24(3):1999–2012
Madhukar A, Williamson CL (2006) 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 Sept 2006, Monterey, California, USA. IEEE Computer Society, pp 179–188

Moore AW, Papagiannaki K (2005) Toward the accurate identification of network applications. In: Dovrolis C (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. Springer, Berlin, vol 3431, pp 41–54

Moore A, Zuev D, Crogan M (2013) Discriminators for use in flow-based classification. Technical report

Okada Y, Ata S, Nakamura N, Nakahira Y, Oka I (2011) 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

Sen S, Spatscheck O, Wang D (2004) Accurate, scalable in-network identification of p2p traffic using application signatures. In: Feldman SI, Uretsky M, Najork M, Wills CE (Eds) Proceedings of the 13th international conference on World Wide Web, WWW 2004, New York, NY, USA, May 17–20, 2004. ACM, p 512–521

Shapira T, Shavit Y (2019) Flowpici: 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. IEEE, pp 680–687

Shen M, Wei M, Zhu L, Wang M (2017) Classification of encrypted traffic with second-order Markov chains and application attribute bigrams. IEEE Trans Inf Forensics Secur 12(8):1830–1843. https://doi.org/10.1109/TIFS.2017.2692682

Shen M, Liu Y, Zhu L, Du X, Hu J (2021) Fine-grained webpage fingerprinting using only packet length information of encrypted traffic. IEEE Trans Inf Forensics Secur 16:2046–2059. https://doi.org/10.1109/TIFS.2020.3046876

Shen M, Zhang J, Zhu L, Xu K, Du X (2021) Accurate decentralized application identification via encrypted traffic analysis using graph neural networks. IEEE Trans Inf Forensics Secur 16:2367–2380. https://doi.org/10.1109/TIFS.2021.3050608

Taylor V, Nurse JRC, Hodges D (2014) Android apps and privacy risks: what attackers can learn by sniffing mobile device traffic. Centre for Doctoral Training in Cyber Security

Velan P, Cermák M, Celeda P, Drasar M (2015) A survey of methods for encrypted traffic classification and analysis. Int J Netw Manag 25(5):355–374

Viegas E, Santin AO, Neves NF, Bessani A, Abreu V (2017) 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. IEEE, pp 1–6

Wang Z (2015) The applications of deep learning on traffic identification. BlackHat USA 24(11):1–10

Wang W, Zhu M, Wang J, Zeng X, Yang Z (2017) 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. IEEE, pp 43–48

Wang W, Zhu M, Zeng X, Ye X, Sheng Y (2017) Malware traffic classification using convolutional neural network for representation learning. In: 2017 international conference on information networking, ICOIN 2017, Da Nang, Vietnam, Jan 11–13, 2017. IEEE, pp 712–717

Xu Y, Liu X, Cao X, Huang C, Liu E, Qian S, Liu X, Wu Y, Dong F, Qiu CW et al (2021) Artificial intelligence: a powerful paradigm for scientific research. Innovation 2(4):100179

Yao H, Gao P, Wang J, Zhang P, Jiang C, Han Z (2019) Capsule network assisted IoT traffic classification mechanism for smart cities. IEEE Intern Things J 6(5):7515–7525

Yao H, Liu C, Zhang P, Wu S, Jiang C, Yu S (2022) Identification of encrypted traffic through attention mechanism based long short term memory. In: IEEE transactions on big data, vol 8, no. 1. pp 241–252. https://doi.org/10.1109/TBDA.2019.2940675

Yeganeh SH, Eftekhar M, Ganjali Y, Keralapura R, Nucci A (2012) CUTE: traffic classification using terms. In: 21st international conference on computer communications and networks, ICCCN 2012, Munich, Germany, July 30–August 2, 2012. IEEE, pp 1–9

Zeng Y, Gu H, Wei W, Guo Y (2019) deep full range: a deep learning based network encrypted traffic classification and intrusion detection framework. IEEE Access 7:45182–45190

Zhang M, Zhang H, Zhang B, Lu G (2012) Encrypted traffic classification based on an improved clustering algorithm. In: Yuan Y, Wu X, Lu Y (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. Springer, Berlin, pp 124–131

Zou Z, Ge J, Zheng H, Wu Y, Han C, Yao Z (2018) 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. IEEE, pp 329–334

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.