Bidirectional Flow-Based Image Representation Method for Detecting Network Traffic Service Categories

Ziyu Jiang

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China

jiagzy@bupt.edu.cn

Abstract. Network traffic identification is crucial for network resource management and improving service quality. Traditional methods, such as port-based and deep packet inspection approaches, face challenges due to the increasing complexity of network environments, privacy concerns, and the emergence of encrypted traffic. This paper aims to address the issues of low accuracy and slow operation speed in encrypted traffic classification while ensuring the protection of user privacy. We propose a data processing method that transforms network traffic into images representing bidirectional flow packet arrival timestamps and packet sizes. By employing this data processing approach and utilizing deep recognition algorithms, the study conducts service analysis on network traffic. Experimental results demonstrate that the bidirectional flow-based image representation method achieves a 90.9% accuracy rate for the traffic analysis task on a TOR-encrypted imbalanced dataset, surpassing the accuracy of the unidirectional flow image method. Furthermore, the method also shows improvements in operation speed, enabling online network traffic detection.

Keywords: Bidirectional Flow; Image Representation; Network Traffic; Detection.

1. Introduction

Network resource management's key component of network traffic identification is the improvement of quality of service (QoS). In recent years, it has risen to the top of the list of hot research topics. Network traffic identification categorizes traffic into service type (instant messaging, voice traffic, etc.). Different traffic classes have distinct QoS requirements. By identifying network traffic, priorities can be set in advance to allocate different service traffic according to demand, achieving better QoS control, avoiding network congestion, improving resource utilization, and ensuring overall user acceptance of the service.

As people's network demands become increasingly diverse, the network environment grows more complex, and issues related to network data security and privacy protection become more severe. In this context, technologies such as encrypted traffic and TOR have rapidly developed and have been widely applied to various software applications. However, encrypted traffic, while providing protection for users, also poses challenges for network traffic service identification analysis.

Deep packet inspection techniques and port-based techniques are examples of conventional service analysis techniques. In order to identify the type of traffic, the former depends on the properties of the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) protocols, which employ port numbers provided by the Internet provided Numbers Authority (IANA). However, with the emergence of non-standard and dynamic ports, this approach is no longer appropriate for the present network environment. The latter raises user privacy concerns, has a large computational overhead, and is unable to handle the majority of encrypted network traffic. It not only detects packet header information but also uses the content of the packets to establish traffic categories. Methods based on statistics and machine learning have become more prevalent with the emergence of machine learning. These techniques often use intricate patterns or supervised learning algorithms as classifiers to extract pertinent characteristics from packets manually or automatically.

In these studies, Chen et al. proposed a method to convert flow sequences into images to display the static and dynamic behavior of different applications, avoiding the use of manual features that may lead to information loss [1]. Shapira et al. introduced the FlowPic method, which converts

extracted network traffic packet size and timestamp information into images for each unidirectional flow and feeds the images into a KNN method, achieving great success with an accuracy of 86.9% on an imbalanced dataset with TOR encryption [2]. They also analyzed the experimental results, finding that the identification of Blowsing and Chat traffic categories was particularly challenging.

Based on the FlowPic method, this paper proposes utilizing the conversion of network flows into images for bidirectional flows, which may facilitate the differentiation of easily confused network traffic categories. To be more specific, this paper utilizes timestamp and package's size instead of using manually extracted feature information handling short time windows of bidirectional flows rather than the entire bidirectional conversation. The experimental results demonstrate that the method based on bidirectional flows achieves 90.9% accuracy on an imbalanced dataset with TOR encryption, surpassing the accuracy of the unidirectional flow method.

2. Data preprocessing

In this chapter, the core content of the article will be elaborated - the methods of data preprocessing, specifically, how to convert network traffic packets (pcap files) into two-dimensional histograms representing bidirectional flows.

2.1. Bidirectional Flow Extraction

The dataset comprises of pcap files obtained using sniffer tools, each of which is associated with a particular application, traffic type, and encryption method. All of these captures, however, also include sessions from many traffic categories since, when running an application action, sessions from numerous other tasks run concurrently. To address this issue in the dataset, after separating the flows, data with fewer packets was removed, both to eliminate packets from other sessions and because packets with too few data could not be used to generate images as part of the dataset. Subsequently, for each pcap file labeled with application, encryption technique, and traffic category, the data was divided into network flows defined by a 5-tuple: {source IP, source port, destination IP, destination port, protocol definition}, with timestamps and packet sizes recorded.

To generate bidirectional flows and to ensure a fixed direction, where the upper part of the graph represents upstream traffic and the lower part represents downstream traffic, an open-source network traffic analysis tool called NFStream in Python was used to analyze and process large amounts of network traffic data [3]. Among its functions, it can quickly process pcap files into bidirectional flows and automatically determine the direction of flows. Then, based on the direction of the packets, the packet size of upstream traffic is recorded as positive and the packet size of downstream traffic is recorded as negative, and the timestamp and packet size of each packet in each bidirectional flow are saved. Specifically, all packets greater than 1500 bytes (less than 5% of the total number of packets) were logged as having a packet size of 1500, because the vast majority of packet sizes are no larger than the Ethernet MTU value of 1500 bytes.

2.2. Data Augmentation

To facilitate better learning, this work fixed a specific time length and required that the traffic for each data point be generated within a fixed time interval. Based on the experiments in the Flowpic paper [2], the author chose 60 seconds as the time interval and divided each bidirectional flow into equally sized blocks. In addition, to increase the number of training set samples and reduce overfitting, following the method described in the paper, each session can be divided into 60-second blocks with 45-second overlap (15-second difference between the start times of adjacent blocks). This data augmentation method is inspired by the usage of object boundary box masks, commonly referred to as sliding windows in object detection tasks [4]. Noting that there was no overlap between the training and testing blocks within a single session, the data augmentation procedure was carried out after all sessions were divided into training and testing sets.

2.3. Image Conversion: Transforming Data into Visual Representation

During the preprocessing stage, records are extracted from each flow, with each record in the data packets consisting of a pair of packet size and arrival timestamp. Then, all record lists belonging to the same traffic category and using the TOR encryption method are collected into a single set.

This study's objective is to create a representation of a flow-based, two-dimensional histogram that may be seen as a payload size distribution plot [5]. All record pairs are plotted in the first stage using the X-axis to represent data packet arrival time and the Y-axis to represent data packet size, with opposite data packets set to negative values. After the processing brfore, data packets size are no more than 1500 bytes. As a result, the Y-axis representing packet size is restricted to the range from -1500 to 1500. By deducting the arrival time of the first data packet in the flow, all arrival time values for the X-axis are first normalized. For simplicity, the 2D histogram is set as a square image in this work. Consequently, all arrival time values and data packet sizes are normalized to be between 0 and 256 (i.e., 60 seconds maps to 256, and -1500 bytes to 1500 bytes maps to 256). Then, each cell in the two-dimensional histogram contains the number of data packets that arrived in each corresponding time interval and were that size. This is done after all normalized record pairs have been added.

This procedure produces a 256x256 histogram from the bidirectional flow arrival times and data packet sizes. The total amount of data packets in the initial time window is equal to the sum of the values in the histogram. Each histogram is stored in an image matrix. These images are then used as inputs for the model.

2.4. Discoveries after data preprocessing

The result obtained in this chapter is the use of the ISCXTor non-Tor pcap file dataset, which is detailed in Chapter 3.2. After processing the data from this dataset as mentioned above, the result in Figure 1 is obtained.

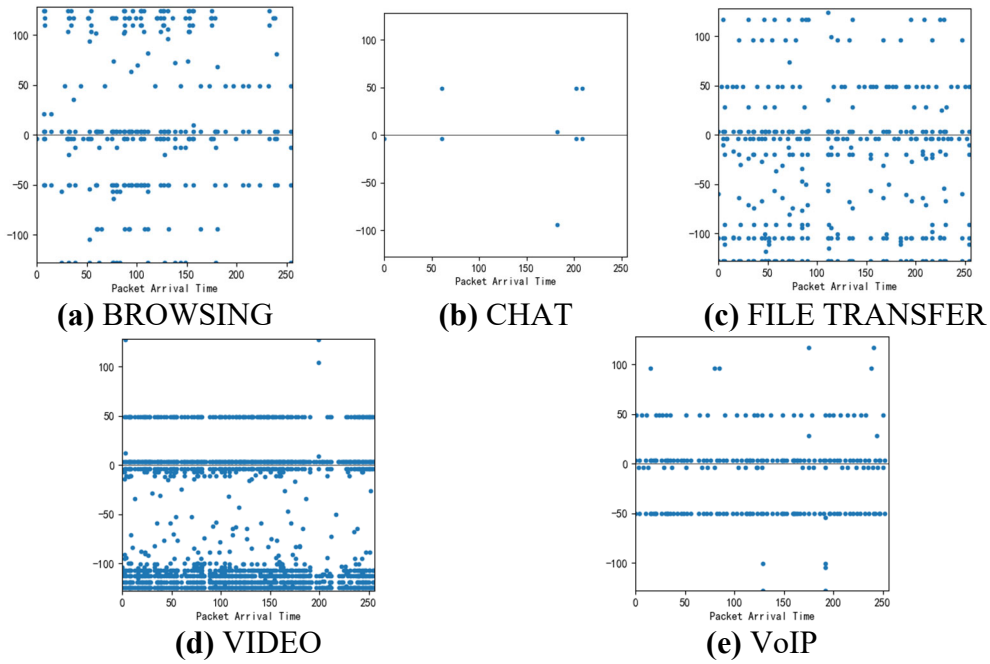

Fig. 1. Converting Various Types of Bidirectional Flows into Images (Original)

Figure 1 displays the results of converting five different types of traffic flows encrypted by TOR into bidirectional flow images. On one hand, this work observes that some services exhibit distinct characteristics, such as VIDEO, where the majority of data packet sizes are clustered around 50, 0, and -100 (after compression). VoIP also stands out from other applications, and even by visual inspection, it can be quickly distinguished. However, some challenges are also identified, such as the very few bidirectional flow data packets for CHAT, which may become an obstacle for feature recognition. Additionally, it is uncertain whether there are still oversights in filtering data packets, as

some bidirectional flows with only a small number of data packets and no particularly obvious features are found in different applications.

3. Results and Analysis

3.1. Hardware Setup

The experiments were conducted on a system running Windows 11, equipped with a 12th Generation Intel Core i7-12700H processor, with a base clock speed of 2.30 GHz. The system had 16.0 GB of RAM to ensure efficient execution of the experiments.

In addition to the CPU and RAM, the experiments primarily utilized Python programming, leveraging the TensorFlow framework and various Python libraries for implementing the experimental tasks. This software setup enabled the efficient execution of the experiments and facilitated the use of cutting-edge machine learning techniques and algorithms.

3.2. Dataset

This work used a labeled pcap file dataset, ISCXTor-nonTor, provided by the Canadian Institute for Cybersecurity [6]. ISCXTor contains seven types of captured traffic for both regular (non-TOR) sessions and Tor sessions. Due to the large size of the packets, running the dataset required significant time and memory, and ultimately, only TOR encrypted traffic was trained and tested. The specific information for the TOR dataset is presented in Table 1.

Table 1. Dataset description

	VoIP	Video	File Transfer	Chat	Browsing
Tor	Facebook, Skype and Hangouts voice calls	Vimeo and YouTube	Skype, SFTP and FTPS	ICQ, AIM, Skype, Facebook and Hangouts	Firefox and Chrome

To examine the effects of converting images into bidirectional and unidirectional flows on accuracy and training speed, we transformed the dataset into two-dimensional histograms corresponding to both unidirectional flows (with the x-axis mapping the 60-second packet arrival time to 256 values, and the y-axis mapping the packet size of 1500 bytes to 256 values) and bidirectional flows. Table 2 presents the size of the dataset when converted into two-dimensional histograms for each label.

Table 2. The number of samples for each category in the dataset

	VoIP	Video	File Transfer	Chat	Browsing
bidirectional flows	3780	1481	1636	359	2305
unidirectional flows	7501	2701	3214	562	3838

3.3. Experimental Implementation Details

A Convolutional Neural Network (CNN), a key component of deep learning, is used in the experiment [7]. To demonstrate the superiority of bidirectional flow features over unidirectional flow features in traffic classification, the network architecture of FlowPic has not been altered in this work [2]. Specifically, the experiments in this paper maintain a LeNet-5-style architecture and apply the same optimization techniques, consisting of seven layers (excluding the input layer) [8].

In order to avoid complex co-adaptation on the training data, dropout approach is used in this work to minimize overfitting. Although all neurons are used, the "dropout" chance is doubled by their outputs [9]. In this study's CNN (Figure 2), the CONV2 layer uses a dropout probability of 0.25 and the fully connected layer a dropout probability of 0.5. During the optimization process, the gradient-based Adam optimizer is employed. Default hyperparameters ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) as provided by Kingma et al. are utilized, and the batch size is set to 128 [10].

To reliably compare the results of all sub-problems, the networks are trained for 40 epochs, with each epoch running 40 batches, as demonstrated in the literature. The highest level of accuracy obtained during training is saved, which is also known as "early stopping"). As shown in Figure 2, the network converges after 10-15 epochs when applied to the multi-class traffic classification problem for TOR. More importantly, the test accuracy curve saturates, ensuring that overfitting is not reached and the results are stable, aside from the small variance between the learning and test curves.

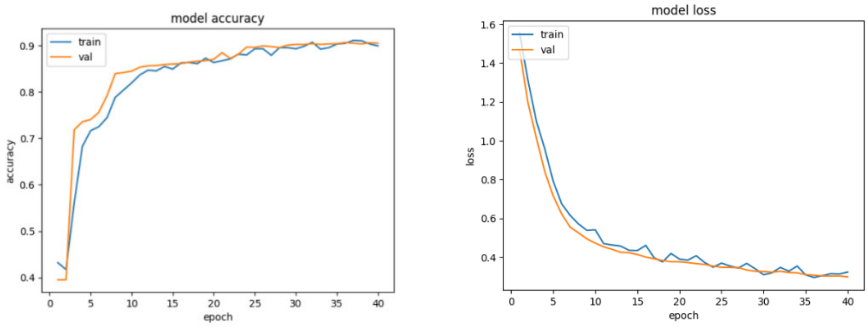

Fig. 2. Training and testing accuracy and loss rate curves of the CNN when applied to the multi-class traffic classification problem for TOR-encrypted flows. (Original)

3.4. Experimental result

3.4.1 Accuracy

Compared to the data processing method of converting unidirectional flows into two-dimensional histograms, this paper achieves a certain improvement. The paper achieves a 90.7% accuracy rate on the TOR encrypted dataset, surpassing the 86.9% accuracy rate reported in the FlowPic paper (Table 3). At the same time, this article also attempts to implement the data processing method of converting unidirectional flows into two-dimensional histograms, achieving an accuracy of 85.3%. Therefore, it can be demonstrated that the additional information provided by bidirectional flows contributes to an improvement in traffic classification. Furthermore, the two-dimensional histograms generated in this experiment have been compressed, using a size of 256x256 (regardless of whether they represent unidirectional or bidirectional flow histograms). This means that the actual sample size to be processed has not increased, but rather, the same size of individual samples now contains more information, leading to improved accuracy.

Table 3. Experimental results

	Accuracy	Speed
Bidirectional flow-based method	90.7%	2min43s
Unidirectional flow-based method	85.3%	5min19s

3.4.2 Speed

In addition to the pursuit of accuracy, the speed of training is also crucial. After converting to bidirectional flows, the size of the dataset generated from the same pcap file is reduced, resulting in a significant increase in training speed. Specifically, the training time for processing the same dataset with unidirectional flow two-dimensional histograms is 5 minutes and 19 seconds, while the training time for bidirectional flows is 2 minutes and 43 seconds. Moreover, this method supports online detection for traffic analysis. The extracted data requires a continuous 60-second window, and since each sample is compressed into a 256x256 image, the testing speed is extremely fast and can be considered negligible. Therefore, with an approximate monitoring delay of 1 minute, online detection for traffic analysis can be achieved.

3.4.3 Analysis of experimental results

This work generates a confusion matrix corresponding to the bidirectional flow two-dimensional histogram method which shows in Figure 3 and conducts further analysis of the experimental results based on this matrix.

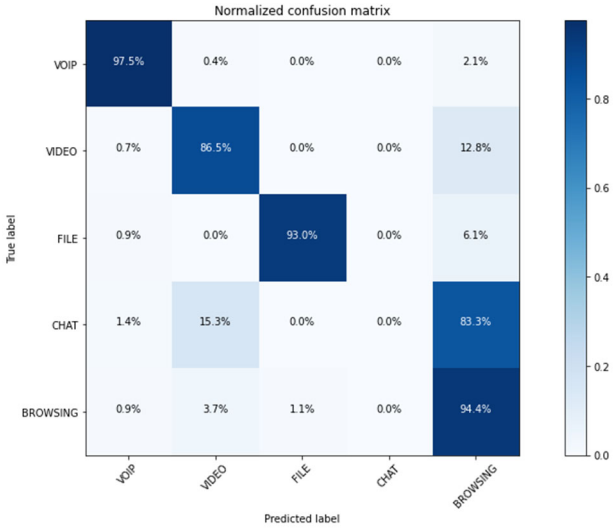

Fig. 3. confusion matrix corresponding to the bidirectional flow two-dimensional histogram method (Original)

From Figure 3, it can be observed that the overall experimental results are satisfactory. However, there is still room for improvement in the classification of the CHAT category. This work attempts to analyze the reasons for this. First, the data volume for the CHAT category is too small, which may not be sufficient for the model to learn the features of CHAT category network traffic. Second, in combination with Figure 1, we can find that each sample in the CHAT category has a small amount of traffic data, making it difficult for the classifier to identify the features of this category from such a limited amount of data, thus misclassifying it as the BROWSING category. At the same time, some samples from each category are identified as BROWSING type, which could be due to insufficient filtering during the preprocessing of the dataset. In fact, while observing the generated two-dimensional histograms, we noticed that some datasets contain only a small amount of traffic and exhibit significant differences compared to other histograms of the same category. These datasets might be the result of packets from other service categories generated during user service usage (e.g., packets sent by network inspection agencies). This could potentially be the cause for a small number of packets being misclassified in each category. Apart from this, other service types have high recognition accuracy, demonstrating the feasibility of using bidirectional flow two-dimensional histograms for service identification.

4. Conclusion

This paper proposes a data preprocessing method that converts network traffic data into bidirectional flow two-dimensional histograms based on timestamps and packet size. This model relies only on a short time window of bidirectional flows and uses flow-based records, resulting in low memory resource, storage, and execution time costs, making it suitable for practical deployment. Compared to previous methods, this approach achieves a certain degree of improvement in both accuracy and execution speed for the service identification analysis of encrypted network traffic. The main contribution of this paper is to enhance service identification analysis for the more challenging TOR encrypted network traffic, and due to its fast execution efficiency, it can enable online service identification analysis with a 1-minute delay. Future research directions are as follows:

A balanced dataset should be used to address the issue of insufficient CHAT category data, making it difficult for the classifier to learn corresponding features.

At the same time, update the data preprocessing method and further consider how to differentiate between CHAT and BROWSING categories.

One can attempt to perform more fine-grained classification on the dataset, such as classifying different software for the same service, which may achieve higher accuracy. This is because the

bidirectional flow two-dimensional histogram features generated by different software for the same service may vary due to differences in protocol design.

References

[1] Chen, Zhitang, et al. "Seq2Img: A sequence-to-image based approach towards IP traffic classification using convolutional neural networks." 2017 IEEE International Conference on Big Data (Big Data) 2017.

[2] Shapira, Tal, and Y. Shavitt. "FlowPic: A Generic Representation for Encrypted Traffic Classification and Applications Identification." IEEE Transactions on Network and Service Management. 99, (2021):1-10.

[3] Le Cun, Yann, et al. "Handwritten zip code recognition with multilayer networks." 10th International Conference on Pattern Recognition. 2, (1990).

[4] Lashkari, Arash Habibi, et al. "Characterization of Tor Traffic using Time based Features." International Conference on Information Systems Security & Privacy 3, (2017).

[5] Qin, Tao , et al. "Robust application identification methods for P2P and VoIP traffic classification in backbone networks." Knowledge-Based Systems 82.C, (2015):152-162.

[6] Szegedy, C. , A. Toshev , and D. Erhan . "Deep Neural Networks for object detection." Advances in Neural Information Processing Systems 26, (2013).

[7] Aouini, Zied, and Adrian Pekar. "NFStream: A flexible network data analysis framework." Computer Networks, vol. 204, Elsevier, (2022), 108719.

[8] Lecun, Yann , et al. "Gradient-Based Learning Applied to Document Recognition." Proceedings of the IEEE 86(11), (1998): 2278-2324

[9] Hinton, Geoffrey E. , et al. "Improving neural networks by preventing co-adaptation of feature detectors.", http://dx.doi.org/10.9774/GLEAF.978-1-909493-38-4_2. 2012.

[10] Kingma, Diederik , and J. Ba . "Adam: A Method for Stochastic Optimization." Computer Science (2014).