TPOTI: A Triplet-Network-based Obfuscated Tor Traffic Identification

Menglei Li*, Chao Wu*, Wentian Zhao*, Tian Song†
*School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing, China
†School of Education, Beijing Institute of Technology, Beijing, China

***Abstract*—The Onion Router (Tor) have brought significant challenges to network traffic analysis. The dynamic and often fragmented nature of network communication makes session continuity difficult to maintain, while the high dimensional features extracted from all packets within a complete session lead to high computational overhead. Furthermore, the inherent imbalance in network traffic data can hinder accurate classification and limit practical applicability in real-world environments. To reduce the computation overhead and improve the classification accuracy, we propose a TriPlet-network based Obfuscated Tor traffic Identification model (TPOTI). Specifically, to preprocess the network traffic data, we propose a Time-aware Session Slicing strategy to slice the traffic session by using the static and dynamic time thresholds to identify the connection actions. Such strategy effectively captures the traffic information as comprehensively as possible while reducing the feature overhead. We then construct a triplet network for feature extraction and traffic classification, which combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to capture spatial and timing features simultaneously. By reducing the distance between intra-class samples and increasing the gap between inter-class samples, the triplet network effectively differentiates normal traffic from Tor traffic and mitigates the data imbalance problem. Experimental results indicate TPOTI provides a low-cost viable solution for detecting Tor traffic and its obfuscated variants, achieving a classification accuracy of 99.45%, which outperforms the existing methods.**

***Index Terms*—Tor anonymous network, Obfuscated traffic identification, Triplet network, Flow interaction, Imbalanced data.**

I. Introduction

Anonymous networks are communication technologies designed to hide important characteristics of traffic in order to enhance user anonymity, with the most common anonymous network being the Tor network. Tor is a low-latency anonymous communication protocol that transmits data over a distributed network to conceal users' identity and location information [1]. Compared to traditional HTTPS encryption techniques, Tor uses its onion routing protocol [2] to build multi-hop encrypted transmission, Which not only ensures that users' communication data is not intercepted by third parties, but more importantly, it conceals the identity of visitors through a multi-layer encryption and stepwise decryption mechanism, thereby ensuring the anonymity and security of the data [3].

Due to the anonymity of the Tor network, it is not only used to ensure communication privacy but also to conceal certain illegal purposes and criminal activities [4]. For instance, the illicit website "Silk Road", which employs the Tor protocol for communication, is rife with various drug and illegal arms transactions. Additionally, attackers exploit the Tor network to deploy botnets, among other malicious activities. Therefore, to ensure the safety and health of cyberspace, identifying the traffic of the Tor network is particularly important. Researchers have proposed and employed various traffic analysis techniques for simple Tor traffic identification (targeting unobfuscated Tor traffic). However, these methods primarily focus on traditional traffic analysis techniques, such as deep packet inspection (DPI) and simple machine learning models, which are unable to adapt to the increasingly sophisticated Tor network protocols.

In recent years, with the updates and improvements to the Tor protocol, a series of traffic obfuscating techniques have been implemented to enhance Tor's privacy protection mechanisms, among which the most common include Obfs4 obfuscation proxy technology (Obfs4) [5], meek covert channel construction technology (Meek) [6], and Format-Transforming Encryption (FTE) [7] traffic transformation technology. The Obfs4 technique adds a camouflage layer to Tor traffic and fills it with random lengths to obfuscate the initial flow signatur [8], making it appear as random data, thereby resisting deep packet inspection and traffic analysis. Meek utilizes cloud services (such as Google or Amazon) [9] as intermediaries to disguise Tor traffic as regular Hypertext Transfer Protocol Secure (HTTPS) traffic, allowing it to bypass censorship and blocking. FTE transforms Tor traffic into encrypted traffic of a specific format, making it appear as other types of traffic during transmission, thereby concealing its true content. The introduction of these obfuscated mechanisms has rendered traditional analysis methods, such as port detection and DPI, ineffective for identifying Tor traffic. Consequently, machine learning and deep learning methods have gradually become the mainstream approaches for Tor traffic identification [10]. However, the current identification of Tor and obfuscated Tor traffic mainly focuses on fine-grained feature extraction of each packet in the complete session flow, or analyzing the features of the entire data flow. These methods lack the exploration of traffic session slicing strategies [11], and the high computational overhead methods that rely on traffic continuity are not suitable

ISBN 978-3-903176-72-0 © 2025 IFIP

for actual network environments. How to identify Tor traffic in long-term normal traffic interactions while reducing the feature cost of identification is still an area that needs further research.

In this paper, we tackle these problems by proposing TriPlet-network based Obfuscated Tor traffic Identification model (TPOTI) to identify and classify Tor traffic, obfuscated Tor traffic, and normal traffic. This method uses the Time-aware Session Slicing strategy to approximate the establishment process of flow interaction connections by setting static and dynamic time thresholds and then slice sessions. By employing a triplet network to learn the intra-class similarity and inter-class differences among samples, the method effectively extracts interaction patterns and features from the sliced traffic, thereby ensuring effective identification of Tor traffic and obfuscated Tor traffic with low feature overhead. Furthermore, the TPOTI method effectively solves the problem of unbalanced datasets by prioritizing the perception of minority class samples, thereby improving the generalization ability of the model and the accuracy of traffic classification.

The major contributions of the proposed paper are summarized as follows:

1) **Setting a Time-aware Session Slicing strategy**: In cases where Tor traffic cannot identify the connection establishment actions through handshake fields, we innovatively use time thresholds as indicators for establishing connections between the two parties. This method slices the complete session traffic based on the static and dynamic time thresholds to reduce reliance on full session traffic characteristics, and enhance the analytical capabilities of the Tor traffic identification training model.
2) **Utilizing Flow-Level interaction features for traffic identification**: We analyze the flow-level interaction information contained in the first 128 bytes of the first 8 packets. This approach allows for an approximate coverage of the entire traffic interaction process during feature extraction, facilitating a comprehensive capture of the differences and variations in the interaction processes of Tor traffic, obfuscated Tor traffic, and normal traffic, thereby improving the accuracy of Tor traffic identification and classification.
3) **Employing a Class-aware Triplet Network for feature extraction**: We propose a triplet network to extract features from Tor traffic and obfuscated Tor traffic. This method prioritizes the selection of imbalanced class samples for combination, allowing for a more comprehensive capture of minority class features when comparing intra-class and inter-class sample similarities, thereby enhancing the ability to identify minority classes and effectively extracting valid features even in imbalanced situations.

The remainder of the paper is organized as follows. In Section II, we summarize the current research in the field of Tor traffic and obfuscated Tor traffic identification, and comparing it with our proposed technique. In Section III, we provide a detailed introduction to the framework of our proposed TPOTI method and its complete technical modules. In Section IV, we present all the detailed information regarding the experimental setup, experimental results, and comparisons of this technology, and analyze the feature overhead compression situation. Finally, we present the conclusions of this technology and outline future work in Section V.

II. Related Work

In this section, we discuss the advanced methodologies in current Tor traffic analysis and obfuscated Tor traffic analysis, comparing these with the TPOTI technology developed in this paper.

Regarding of traffic identification, extensive research has been conducted both domestically and internationally, employing various methods including port number and protocol matching, deep packet inspection based on payload content, behavior-based network behavior analysis, and machine learning and deep learning techniques based on traffic features. However, due to the encryption and anonymity of Tor traffic, the first three identification techniques are no longer applicable. Therefore, machine learning and deep learning methods based on traffic features have become key technologies for addressing the problem of Tor traffic identification.

When analyzing and identifying unobfuscated Tor traffic, the publishers of the ISCXTor2016 dataset implemented a series of machine learning algorithms for detecting and identifying Tor traffic while making the dataset open source [12]. They focused on time-related features, using algorithms like Random Forest to identify Tor traffic and its corresponding applications. In 2020, the same researchers expanded upon the original ISCXTor2016 dataset by collecting and releasing the Darknet2020 dataset [13], while also proposing the DeepImage method. This method utilizes feature selection to extract the most important features from headers and payloads, generating grayscale images that are fed into a two-dimensional convolutional neural network (2D-CNN) for detecting and characterizing darknet traffic. Marim [14] trained on the Darknet2020 dataset, introduced address information as a classification feature in conventional statistical features, and used recursive feature elimination method to select the most representative features. In addition, others [15], [16] also choose to extract the statistical characteristics of traffic to identify Tor traffic.

In addition to using statistical features for traffic analysis, some researchers [17] focused on the first 54 bytes of the data packet, and Dodia [18] used the first three active Tor connections in each Packet Capture file (PCAP) of Tor traffic to extract browser fingerprints and construct host features, achieving recognition of Tor traffic with an accuracy of up to 98%.

With the application of traffic obfuscated techniques, although most research still tends to focus on detecting unobfuscated Tor traffic, there has also been a series of studies targeting obfuscated Tor traffic. In the research focused on the identification of single obfuscated Tor traffic, Yao [19] proposed a Gaussian mixture-based Markov model targeting the Meek obfuscation protocol. They modeled the density distribution of the time intervals and sizes of Meek-Tor traffic messages using a Gaussian mixture function, employing two-dimensional observations to identify Meek-Tor obfuscated traffic, ultimately achieving an accuracy of 99%. And He [20], Wang [21] performed traffic detection based on the special properties of Obfs4 obfuscated traffic and improved the recognition accuracy and speed. However, the above methods are limited to a certain type of Tor obfuscated traffic.

Additionally, when detecting multiple types of obfuscated Tor traffic simultaneously, some researchers [22] identified traffic generated by Tor's pluggable obfuscated transport methods, including Obfs3, Obfs4, and ScrambleSuit. However, both Obfs3 and ScrambleSuit are no longer used in Tor. Another study [23] collected three currently popular types of obfuscated Tor traffic—Meek, FTE, and Obfs4—and made the dataset open source. They extracted 12 statistical features from the flow using a sliding window and utilized a multi-class machine learning model to identify the obfuscated Tor traffic and its types, achieving an accuracy of 99%. Lv [24] constructs an incremental learning framework for the feature set of obfuscated Tor traffic technology and uses edge sample enhancement method for recognition. However, the features selected in the study relied on the continuity of the traffic and only extracted low-order statistical features, failing to fully exploit the potential feature space.

In the aforementioned traffic identification methods, whether identifying Tor traffic or obfuscated Tor traffic, the traffic features used primarily rely on the continuity of the traffic, such as the need to count the number of packets within a unit of time. In the context of large-scale traffic, calculating the statistical features of the traffic requires significant amounts of time and memory. Additionally, since Tor traffic constitutes a very small proportion of daily traffic, the resulting data imbalance also poses a challenge. Therefore, how to eliminate the dependence of Tor traffic identification on continuous streams and balanced data volumes, as well as how to reduce the time and memory costs of traffic identification, has become a key issue that this project needs to address.

III. Our Approach

In this section, we present a comprehensive overview of the proposed framework designed for the identification of Tor traffic. Furthermore, we analyze the possibility behind the model's effectiveness in differentiating between various traffic volumes.

A. Framework Overview

Figure 1 shows the main components of the framework: dataset, Time-aware Session Slicing, feature extraction using triplet network and traffic identification.

The Tor traffic and obfuscated Tor traffic classification model is based on facts:

1) The detection of interaction traffic over extended periods is constrained by limited time and memory resources.
2) The distinct characteristics of various traffic types are primarily manifested in the dynamic behaviors associated with the interaction process during connection establishment [25]. Consequently, when resources are limited, the identification and classification of traffic types should prioritize the interaction actions occurring in the connection establishment phase, while relatively downplaying the data transmission status that follows.
3) Tor traffic and obfuscated Tor traffic typically lack a clear handshake process [26], which hinders the identification of the interaction connection process based on Transmission Control Protocol (TCP) handshake fields.

Guided by the aforementioned facts, we utilized commonly used open-source datasets to categorize the collected original PCAP packets into five distinct categories for the entire experiment. These categories include normal traffic, Tor traffic and obfuscated Tor traffic types such as Meek-based, Obfs4-based, and FTE-based. The network data for each PCAP packet encompasses all interactive traffic data packets exchanged between two hosts over a specified period, which includes multiple connection processes and data transmission sessions.

In the data preprocessing module, traffic sessions are segmented using two methods: (1) static time thresholds and (2) a dynamic threshold determined by the inflection point of the cumulative distribution function (CDF). The resulting segments are stored as sliced PCAP files, each corresponding to a single session connection. Features are extracted from these slices and converted into 8×128 grayscale images and 8×128 byte sequences, serving as inputs for subsequent feature extraction models.

In the feature extraction module, this paper employs a triplet network model to extract feature vectors from five types of traffic data. The feedforward subnetwork within the triplet network comprises both CNN and LSTM [27]. These two feature sets are then combined to form a spatiotemporal feature vector for the traffic data packets. In the traffic classification module, a Multi-Layer Perceptron network (MLP) [28] is utilized to classify the feature vectors. This approach effectively distinguishes between normal traffic and Tor traffic, as well as differentiates among various obfuscated categories of Tor traffic.

B. Threat Model

In the threat model for Tor traffic identification, attackers infer whether user's traffic is Tor traffic by analyzing network traffic characteristics and patterns. In this paper, the attacker is typically positioned at

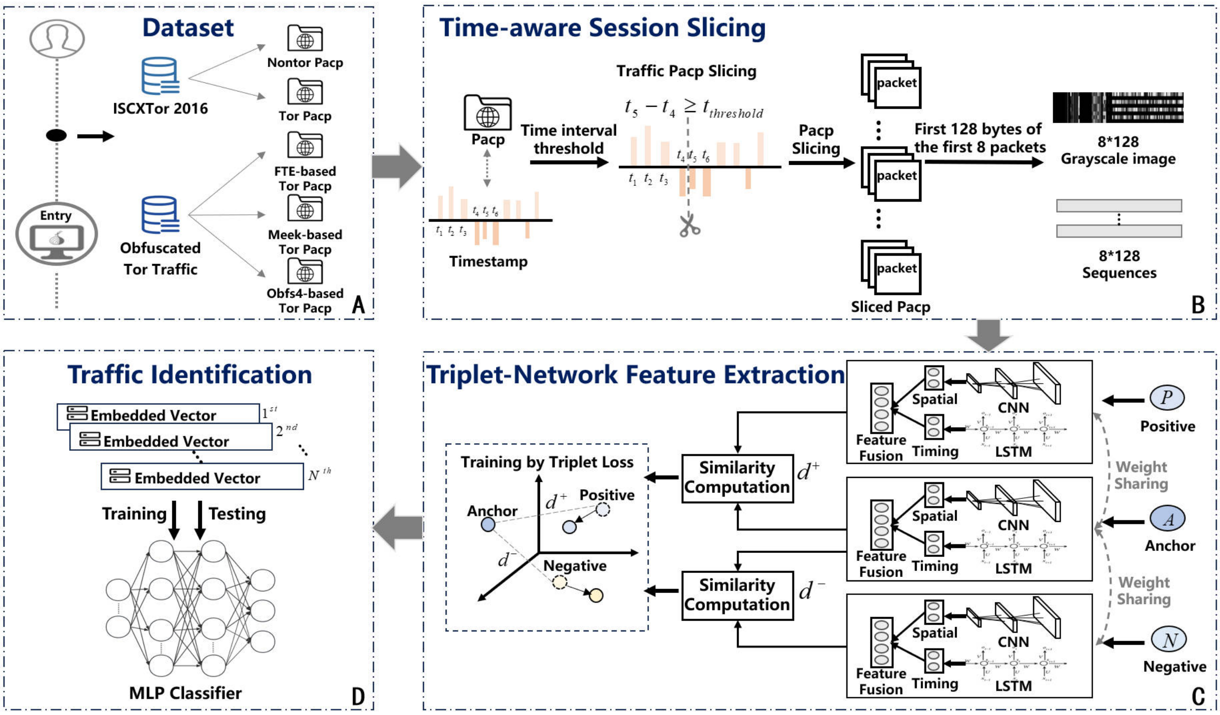

Fig. 1. Framework of our proposed method, consisting of:(A) Dataset: Combination of two open source datasets.(B) Time-aware Session Slicing: Slicing the traffic session by using a time threshold to identify the connection actions.(C) Triplet-Network-Based Feature Extraction: Using triplet network which combines CNN and LSTM to capture features of Tor traffic, and deal with imbalanced datasets.(D) Traffic Identification: Tor traffic classification using MLP.

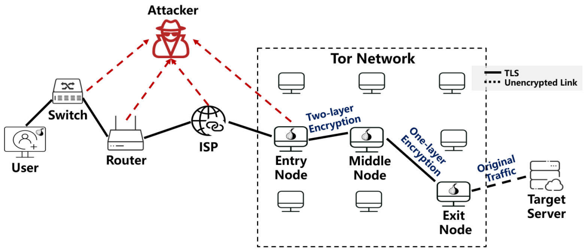

Fig. 2. Threat model for Tor traffic classification.

the switch, router, Internet Service Provider (ISP), or entry node, and may include eavesdroppers on the user's local network, local system administrators, ISPs, or operators of the entrance node. Attackers can only observe and record the traces of network traffic that pass through their vantage points; they do not possess the capability to discard, delay, or modify actual packets in the traffic, nor can they decrypt them. The Tor traffic identification threat model is illustrated in Figure 2.

C. Data Preprocessing

The data preprocessing module includes the conversion of network stream data into a format suitable for feature selection algorithms and deep learning models. This module consists of three key components: network flow slicing, traffic anonymization, and traffic extraction and conversion.

Time-aware Session Slicing: Network session slicing involves analyzing the traffic interaction information collected between two hosts over a specified duration, which includes multiple connection establishment processes and data transmission events. Given the constraints of limited resources such as memory and time, traffic identification and classification typically prioritize the interaction actions occurring during the connection establishment phase, while the subsequent data transmission process is often overlooked. Therefore, accurately identifying the connection establishment actions for normal traffic, Tor traffic, and obfuscated Tor traffic is crucial.

However, the inherent anonymity of Tor traffic and obfuscated Tor traffic results in significant differences in the structure and transmission methods of their data packets compared to traditional TCP traffic. The design of these protocols complicates traffic analysis, as it is challenging to rely solely on TCP handshake fields for effective feature recognition. In this context, it becomes particularly reasonable to approximate the identification of the connection process based on the time intervals between data packets.

In a complete network traffic session (stored in PCAP format), multiple interactive connection processes and data transmission processes are typically included. By analyzing the inter-packet time intervals in the original PCAP packets of five types of traffic, it can be observed that the time intervals between packets are mostly concentrated below 0.4 seconds, often in millisecond-scale decimals. The short intervals usually correspond to stable data transmission processes. However, when the time interval between packets significantly increases and exceeds a specific threshold, it often indicates the initiation of a new interactive connection action. Therefore, slicing the traffic when the time interval reaches a predefined threshold can effectively preserve the dynamic characteristics of traffic interaction connections. Based on the analysis of the CDF, the time threshold is set between 0.4 seconds and 1.5 seconds. This range is chosen to retain a sufficient number of original packets to capture as many features of the traffic session as possible, while significantly reducing the computational cost of Tor traffic detection and minimizing data redundancy.

The time thresholds for the data transmission and connection establishment processes can also be dynamically determined. This is achieved by locating the extrema of the second derivative of the CDF. The method identifies robust inflection points on the CDF curve. These points correspond to locations where the probability density changes most significantly. Taking FTE_Tor traffic as an example, Figure 3 shows that the inflection point occurs at 1.91 seconds. Based on this approach, different traffic types can adaptively determine distinct time thresholds. The differentiated time thresholds for five types of traffic are presented in Table I.

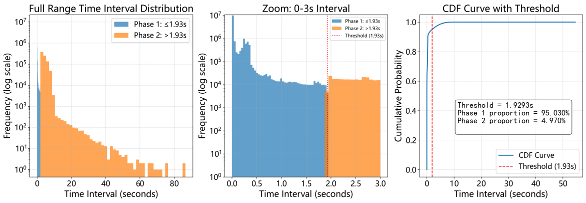

Fig. 3. Visualization of time threshold determination of FTE_Tor traffic based on CDF.

TABLE I
DYNAMIC DETERMINATION OF TIME THRESHOLD BASED ON INFLECTION POINT OF CDF.

Type	nonTor	Tor	FTE	Meek	Obfs4
Time Threshold	0.04	0.03	1.93	0.99	2.24

Traffic anonymization: After slicing the stream data packets to obtain sub-stream data packets that approximate the process of traffic interaction and connection, the first 128 bytes of each first 8 TCP data packets from each sub-stream are extracted. This approach captures the traffic characteristics and patterns as comprehensively as possible while utilizing lower computing resources. Simultaneously, the addresses within the network traffic data are anonymized to mitigate the risk of the model overly relying on addresses for traffic category classification and judgment. The address information, which includes source and destination Media Access Control addresses (MAC), Internet Protocol addresses (IP), and port numbers, is replaced with zeros of equal length and standardized to the same address.

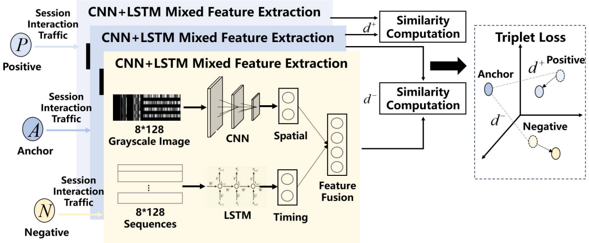

Fig. 4. Detailed structure of Class-aware Triplet Network model.

Traffic extraction and conversion: The anonymized traffic data is converted into 8 × 128 grayscale images and 8 × 128 byte sequences, which serve as inputs for subsequent feature extraction. This transformation facilitates the effective analysis of traffic characteristics and patterns, enabling the model to leverage both visual and sequential data representations for improved classification and identification.

D. Triplet-Network-Based Feature Extraction

In this paper, the feature extraction module employs a triplet network model, which consists of three subnetworks with shared weights and hyperparameters, as illustrated in Figure 4. This model effectively extracts features from traffic samples by learning the similarities (within-class) and differences (between-class) between samples. As an extension of Siamese Network, triplet networks are particularly well-suited for scenarios involving multiple or uncertain sample types, especially when the training dataset is limited or imbalanced.

Specifically, the triplet network generates an array containing three distinct input instances from the training data to form a triplet structure, defined as anchor input, positive input, and negative input. These inputs are then fed into the corresponding subnetworks for weight sharing. To address the issue of sample imbalance in the dataset, the constructed triplet prioritizes traffic data with the fewest sample types. The definitions of anchor input, positive input, and negative input are as follows:

- **Anchor (A)**: A sample randomly selected from the traffic training dataset to serve as a reference, referred to as the Anchor (e.g., Meek-based traffic).
- **Positive sample (P)**: A sample randomly selected from the same class as the Anchor (e.g., Meek-based traffic).
- **Negative sample (N)**: A sample randomly selected from a different class than the Anchor (e.g., Normal traffic).

The triplet structure inputs are fed separately into the subnetworks, which comprise CNN and LSTM network. The CNN, with its advantages of local connections, weight sharing, and pooling layers, effectively extracts local features from images while reducing the number of parameters, making it well-suited for extracting spatial features from traffic grayscale images. In contrast, the LSTM captures long-term dependencies in sequential data through its gating mechanisms and cell states. Its adaptive memory selection and forgetting functions enable the effective extraction of timing features from a series of flow sequences. By combining the features extracted from both networks, the model can effectively obtain the spatiotemporal characteristics of traffic flow interactions. Consequently, CNN and LSTM models were selected as the foundational subnetworks, with weights shared among the three subnetworks.

The fundamental principle of the triplet network is to simultaneously input anchor points, positive samples, and negative samples. And by utilizing the feature embedding space learned by the subnetworks, the model aims to minimize the distance between the anchor points and positive samples while maximizing the distance between the anchor points and negative samples. This approach facilitates effective similarity learning and classification, leading to the definition of the triplet loss function as follows:

$$\mathcal{L} = \max(d(A,P) - d(A,N) + \alpha, 0), \qquad (1)$$

where d represents the distance between two samples. In this paper, the Euclidean distance is used to measure the straight-line distance between two samples. $d(A,P)$ is the distance between the anchor point and the positive sample, $d(A,N)$ is the distance between the anchor point and the negative sample, and α is a hyperparameter used to achieve the goal of bringing the positive sample closer to the anchor point and keeping the negative sample away from the anchor point, ensuring that the distance between the anchor point and the negative sample is greater than the distance between the anchor point and the positive sample. During the entire training process, the triplet network continuously adjusts the parameters of its subnetworks to minimize the loss function. Through backpropagation, the network learns to map similar samples to nearby positions in the feature space, while mapping dissimilar samples to more distant positions. The process ultimately enables effective feature extraction from the input traffic samples.

E. Traffic Identification

In this paper, on the basis of implementing feature extraction, this experiment inputs the extracted feature vectors into a MLP for training. The MLP, a type of feedforward neural network, consists of multiple layers of neurons, with each neuron connected to the neurons in the previous layer through an activation function, thereby forming a complex nonlinear mapping. Upon receiving the extracted feature vectors, the MLP learns the relationships and interactions among the features. Through the neurons in the hidden layers, the model can automatically identify the most significant feature combinations for final classification, thereby enhancing its performance. The training process of the MLP employs the backpropagation algorithm in conjunction with gradient descent to update the weights, ultimately achieving five classifications of traffic types.

IV. Experimental Evaluation and Discussing

In this section, we provide a detailed description of the experimental design of the TPOTI method and evaluate its accuracy in identifying Tor traffic and obfuscated Tor traffic, while also comparing it with other existing techniques.

A. Dataset and Experimental environment

The approach is tested on open-source datasets. It combines two open-source datasets: the ISCXTor2016 (Tor-nonTor) dataset [12] and the Obfuscated-Traffic dataset [23]. Both datasets contain raw PCAP session traffic packets and have been utilized by numerous researchers for experimental studies, making them representative.

ISCXTor2016: The ISCXTor2016 dataset is a labeled open-source dataset released by the University of New Brunswick (UNB), which includes two types of traffic: normal traffic and Tor traffic, with the latter being unobfuscated. This dataset is an actual traffic dataset generated from a defined set of tasks.

Obfuscated-Traffic: The Obfuscated-Traffic dataset was published by Xu et al. and includes Tor traffic data for three types of obfuscation. They deployed Tor clients based on Meek, Obfs4, and FTE on six cloud servers and developed scripts to access the internet, thereby automatically generating network traffic.

All experiments were executed on a physical workstation equipped with an Intel® Core™ i7-10510U CPU @ 2.30GHz and 32GB DDR4 SDRAM, running a native Windows 10 operating system.

B. Experimental Results and Comparison

Before the experiment, the raw dataset (session traffic PCAP packets) is preprocessed. The sliced dataset is further divided into training and testing sets with a ratio of 8:2. After preprocessing the sliced packets and converting them into 8×128 grayscale images and byte sequences, the dataset is input into the TPOTI model for training.

1) ***Comparative Experiment***

In the comparative experiments, we evaluate the TPOTI model with the algorithms proposed by Lashkari [12] and Xu [23] on two integrated datasets, and the evaluation results presented in Table II. Among them, the Lashkari's method extracts the time-related features of the packet interval (such as statistical features of forward, backward, and bidirectional flows), and classification is performed using Random Forest algorithms. However, due to the lack of consideration for class imbalance issues and the use of a larger flow timeout value, the classification performance is suboptimal, achieving an F1-score of only 72.13%.

In the Xu's method, time-based statistical features are extracted using a sliding window approach, and classification is conducted using algorithms such as Random Forest. Although these advanced machine learning models are employed, the optimal results are still significantly inferior to those of the TPOTI model, primarily because class imbalance is not addressed. In contrast, the TPOTI model not only effectively mitigates sample imbalance issues but also captures both spatial and temporal features of the data packets. As a result, it achieves significant advantages in accuracy, precision, recall, and F1-score, demonstrating its efficiency and robustness in the task of Tor traffic classification. And smaller time thresholds (0.45s) can improve classification accuracy, likely because finer-grained time segmentation preserves more temporal sequence information. Notably, the use of dynamically time thresholds for session segmentation further enhances classification performance, achieving an accuracy of up to 99.45%.

TABLE II
EVALUATION SCORES OF COMPARATIVE CLASSIFICATION ALGORITHMS.

Dataset	**ISCXTor2016 && Obfuscated-Traffic**			
Method	**Accuracy**	**Precision**	**Recall**	**F1_Score**
Lashkari et al. [12]	87.68	83.19	64.73	72.13
Xu et al. [23]	89.09	85.08	74.26	79.30
TPOTI(t=1.5s)	98.76	98.77	98.76	98.76
TPOTI(t=1.3s)	99.02	99.01	99.02	99.01
TPOTI(t=0.68s)	99.12	99.13	99.12	99.12
TPOTI(t=0.45s)	99.23	99.25	99.23	99.24
TPOTI(Dynamic)	**99.45**	**99.45**	**99.45**	**99.45**

2) *Ablation Study*

To validate the effectiveness of the TPOTI model, we design an ablation study. In this experiment, by gradually removing specific components of the model, the contribution of each module to the system performance is evaluated, and the overall effectiveness of the system is verified. The classification evaluation results under four different scenarios are shown in Table III.

Specifically, when only the CNN module is removed (using triplet network + LSTM), the model's accuracy decreases by approximately 1%. But the model's performance reaches overfitting. This result indicates that the CNN module plays a certain role in extracting spatial features and preventing overfitting. When only the LSTM module is removed (using triplet network + CNN), the model's accuracy, precision, recall, and F1-score all significantly decline, with accuracy dropping by 4.1% -9.45%. This result shows that the LSTM module plays an irreplaceable role in capturing temporal sequence features, such as temporal dependent relationships between packets. When only the triplet network is removed (using the CNN + LSTM), the model's accuracy, precision, recall, and F1-score experience a substantial decline, with accuracy decreasing by over 17%. Additionally, the model again exhibits overfitting. This highlights the significant advantage of the triplet network in addressing class imbalance issues, thereby effectively enhancing the model's generalization capability. The experimental results demonstrate that all three modules contribute significantly to the performance of the TPOTI model. This is because the original data set is unbalanced in quantity after being sliced according to the time thresholds. For example, the number of three types of obfuscated Tor traffic is significantly higher than the number of normal traffic. But the triplet network can effectively deal with the imbalanced dataset, leading to improvement.

C. *Overhead Discussion*

In order to verify the rationality of the Flow Interaction Time-aware Session Slicing, we analyze from two dimensions: feature compression efficiency and model performance balance. The experimental results are shown in Table IV. "Dataset Size" stands for the size of packets that are analyzed to classify the Tor traffic data in the combination of two open source datasets. "FLOPS/sample" denotes the number of floating point operations used to classify each individual sample by models, and "Total FLOPS" represents the total floating point operations used to classify all samples in the dataset.

Different session segmentation methods result in significant variations in dataset size and computational cost (FLOPS). Traditional approaches based on Random Forests (RF), such as those proposed by Lashkari and Xu, operate on relatively large datasets (60GB and 40GB, respectively) and maintain low per-sample FLOPS. However, their total computational overhead remains high, reaching 22 and 14.7 trillion FLOPS, respectively, indicating inefficiency when handling large-scale data.

In contrast, the TPOTI method leverages time-aware session slicing to substantially reduce data volume while maintaining high detection performance. For instance, with a time threshold of 0.45s, the processed dataset is reduced to approximately 3% of the original size, with total FLOPS dropping to 6.61 trillion, yet the detection accuracy remains as high as 99.23%. As the time threshold increases, both dataset size and FLOPS further decrease, while the accuracy experiences only a marginal drop of 0.48 percentage points, demonstrating strong robustness and computational efficiency.

Notably, the TPOTI (Dynamic) variant achieves the lowest total computational cost with a minimal dataset size of 4GB and maintains the same model complexity as other TPOTI variants. And the accuracy of the model has been improved. Compared to traditional RF-based methods, TPOTI (Dynamic) offers superior efficiency and classification accuracy, making it particularly suitable for resource-constrained or real-time applications.

TABLE III
EVALUATION SCORES OF ABLATION STUDY. SUBTABLES (A), (B), (C) AND (D) DEMONSTRATE THE RESULTS AT TIME THRESHOLDS OF 1.5S, 1.3S, 0.68S AND 0.45S, RESPECTIVELY.

method	**A: time threshold = 1.3s**				**method**	**B: time threshold = 0.68s**			
	accuracy	precision	recall	F1		accuracy	precision	recall	F1
CNN+LSTM	84.47	84.45	84.47	84.46	CNN+LSTM	88.47	88.45	88.47	88.4
Triplet+LSTM	98.71	98.72	98.71	98.71	Triplet+LSTM	98.84	98.85	98.84	98.84
Triplet+CNN	89.67	89.95	89.67	89.71	Triplet+CNN	92.38	92.01	92.36	92.15
TPOTI	**99.02**	**99.01**	**99.02**	**99.01**	**TPOTI**	**99.12**	**99.13**	**99.12**	**99.12**
method	**C: time threshold = 0.45s**				**method**	**D: time threshold = Dynamic**			
	accuracy	precision	recall	F1		accuracy	precision	recall	F1
CNN+LSTM	87.93	88.01	87.98	87.99	CNN+LSTM	89.83	87.28	88.78	88.02
Triplet+LSTM	98.70	98.72	98.70	98.71	Triplet+LSTM	98.83	98.61	98.65	98.62
Triplet+CNN	95.13	94.92	95.13	94.97	Triplet+CNN	95.82	95.14	94.93	94.97
TPOTI	**99.23**	**99.25**	**99.23**	**99.24**	**TPOTI**	**99.42**	**99.45**	**99.45**	**99.45**

TABLE IV
SUMMARY OF TOTAL PROCESSED PACKETS AND COMPUTATIONAL OVERHEAD OF DIFFERENT METHODS.

Model	**Dataset Size**	**FLOPS/ sample (M)**	**Total FLOPS (T)**
Original Data	668G	-	-
Lashkari et al.(RF)	60G	1.57	22
Xu et al. (RF)	40G	1.57	14.70
TPOTI (t=0.45s)	20.8G	1.96	6.61
TPOTI (t=0.68s)	17.6G	1.96	5.57
TPOTI (t=1.3s)	10.4G	1.96	3.30
TPOTI (t=1.5s)	9G	1.96	2.84
TPOTI (Dynamic)	**4G**	**1.96**	**1.25**

V. CONCLUSION

In this paper, we focus on the characteristics of the interaction connection establishment phase at the session level. Aiming at the absence of explicit handshake processes in Tor traffic and obfuscated Tor traffic (i.e., it is impossible to make interactive connection judgments based on the Transmission Control Protocol handshake field), we propose a method based on threshold slicing of packet time intervals between inner-packets. Specifically, the static and dynamic time thresholds are set for session traffic using the Cumulative Distribution Function (CDF) to approximate the extraction of the key connection establishment phase within the session. To fully capture the spatiotemporal features of the interaction connection establishment process, we employ Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) for feature extraction, combined with a triplet network to address the class imbalance issue in the dataset, thereby maximizing the discriminative capability of features across five types of traffic. The effectiveness of the model is evaluated based on metrics such as accuracy, precision, recall, and F1-score, and its feasibility and validity are further verified through ablation experiments and comparisons with state-of-the-art methods.

We combine two widely used open source datasets (ISCXTor2016 and Obfuscated-Traffic) for experimental verification and conducts comparative analysis with existing technologies. The experimental results demonstrate that our model exhibits significant advantages in metrics such as accuracy, precision, recall, and F1-score. Method based on dynamic time threshold has achieved an outstanding performance of 99.45% in the accuracy metric. This success is primarily attributed to the fact that our proposed method can effectively deal with the category imbalance problem in the dataset, and through a reasonable traffic session slicing strategy, it can more fully extract the key features of the five types of traffic in the process of establishing interactive connections. The subsequent ablation experiments further corroborate these findings. These results not only provide a novel solution for the identification of Tor traffic and its obfuscated variants but also offer valuable insights and reference for research in related fields.

REFERENCES

[1] Roger Dingledine, Nick Mathewson, Paul F Syverson, et al. Tor: The second-generation onion router. In *USENIX security symposium*, volume 4, pages 303–320, 2004.

[2] Kyle Swan. Onion routing and tor. *Geo. L. Tech. Rev.*, 1:110, 2016.

[3] Ya Liu, Xiao Wang, Bo Qu, and Fengyu Zhao. Atvitsc: A novel encrypted traffic classification method based on deep learning. *IEEE Transactions on Information Forensics and Security*, 2024.

[4] Liang Ke, Xinyu Chen, and Haizhou Wang. An unsupervised detection framework for chinese jargons in the darknet. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pages 458–466, 2022.

[5] Felix Günther, Douglas Stebila, and Shannon Veitch. Obfuscated key exchange. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2385–2399, 2024.

[6] Yibo Xie, Gaopeng Gou, Gang Xiong, Zhen Li, and Wei Xia. Domeye: Detecting network covert channel of domain fronting with throughput fluctuation. *Computers & Security*, 144:103976, 2024.

[7] Jonathan Oakley, Lu Yu, Xingsi Zhong, Ganesh Kumar Venayagamoorthy, and Richard Brooks. Protocol proxy: An fte-based covert channel. *Computers & Security*, 92:101777, 2020.

[8] Xuebin Wang, Zeyu Li, Wentao Huang, Meiqi Wang, Jinqiao Shi, and Yanyan Yang. Towards comprehensive analysis of tor hidden service access behavior identification under obfs4 scenario. In *Proceedings of the 2021 ACM International Conference on Intelligent Computing and its Emerging Applications*, pages 205–210, 2021.

[9] Xuebin Wang, Zhipeng Chen, Zeyu Li, Wentao Huang, Meiqi Wang, Shengli Pan, and Jinqiao Shi. Identification of meek-based tor hidden service access using the key packet sequence. In *International Conference on Computational Science*, pages 554–568. Springer, 2022.

[10] Nhien Rust-Nguyen, Shruti Sharma, and Mark Stamp. Darknet traffic classification and adversarial attacks using machine learning. *Computers & Security*, 127:103098, 2023.

[11] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 1131–1148, 2019.

[12] Arash Habibi Lashkari, Gerard Draper Gil, Mohammad Saiful Islam Mamun, and Ali A Ghorbani. Characterization of tor traffic using time based features. In *International Conference on Information Systems Security and Privacy*, volume 2, pages 253–262. SciTePress, 2017.

[13] Arash Habibi Lashkari, Gurdip Kaur, and Abir Rahali. Didarknet: A contemporary approach to detect and characterize the darknet traffic using deep image learning. In *Proceedings of the 2020 10th International Conference on Communication and Network Security*, pages 1–13, 2020.

[14] Mateus Coutinho Marim, Paulo Vitor Barbosa Ramos, Alex B Vieira, Antonino Galletta, Massimo Villari, Roberto M de Oliveira, and Edelberto Franco Silva. Darknet traffic detection and characterization with models based on decision trees and neural networks. *Intelligent Systems with Applications*, 18:200199, 2023.

[15] Raju Gudla, Satyanarayana Vollala, Ruhul Amin, et al. A novel approach for classification of tor and non-tor traffic using efficient feature selection methods. *Expert Systems with Applications*, 249:123544, 2024.

[16] Sanaa Mohsin, Baraa Wasfi Salim, and Awaz Naaman Saleem. Darknet traffic recognition using meta-learning. In *2024 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, pages 95–99. IEEE, 2024.

[17] Pitpimon Choorod, Tobias J Bauer, and Andreas Aßmuth. Distinguishing tor from other encrypted network traffic through character analysis. *arXiv preprint arXiv:2405.09412*, 2024.

[18] Priyanka Dodia, Mashael AlSabah, Omar Alrawi, and Tao Wang. Exposing the rat in the tunnel: Using traffic analysis for tor-based malware detection. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 875–889, 2022.

[19] Zhongjiang Yao, Jingguo Ge, Yulei Wu, Xiaodan Zhang, Qiang Li, Lei Zhang, and Zhuang Zou. Meek-based tor traffic identification with hidden markov model. In *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 335–340. IEEE, 2018.

[20] Yongzhong He, Liping Hu, and Rui Gao. Detection of tor traffic hiding under obfs4 protocol based on two-level filtering. In *2019 2nd International Conference on Data Intelligence and Security (ICDIS)*, pages 195–200. IEEE, 2019.

[21] Han Wang, Xiangyang Luo, and Yuchen Sun. An obfs-based tor anonymous communication anline identification method. In *2020 6th International Conference on Big Data and Information Analytics (BigDIA)*, pages 361–366. IEEE, 2020.

[22] Mohammad Hassan Mojtahed Soleimani, Muharram Mansoorizadeh, and Mohammad Nassiri. Real-time identification of three tor pluggable transports using machine learning techniques. *The Journal of Supercomputing*, 74(10):4910–4927, 2018.

[23] Wenliang Xu and Futai Zou. Obfuscated tor traffic identification based on sliding window. *Security and Communication Networks*, 2021(1):5587837, 2021.

[24] Sicai Lv, Zibo Wang, Yunxiao Sun, Chao Wang, and Bailing Wang. Edge exemplars enhanced incremental learning model for tor-obfuscated traffic identification. *Electronics*, 14(8):1589, 2025.

[25] Yao Li, Xingshu Chen, Wenyi Tang, Yi Zhu, Zhenhui Han, and Yawei Yue. Interaction matters: Encrypted traffic classification via status-based interactive behavior graph. *Applied Soft Computing*, 155:111423, 2024.

[26] Likun Liu, Haining Yu, Shilin Yu, and Xiangzhan Yu. Network traffic obfuscation against traffic classification. *Security and Communication Networks*, 2022(1):3104392, 2022.

[27] Zhen Zhao, Ze Li, Fuxin Li, and Yang Liu. Cnn-lstm based traffic prediction using spatial-temporal features. In *Journal of Physics: Conference Series*, volume 2037, page 012065. IOP Publishing, 2021.

[28] Shahbaz Rezaei and Xin Liu. Deep learning for encrypted traffic classification: An overview. *IEEE communications magazine*, 57(5):76–81, 2019.