Hindawi
Security and Communication Networks
Volume 2022, Article ID 3306098, 12 pages
https://doi.org/10.1155/2022/3306098

Research Article

A Method for Identifying Tor Users Visiting Websites Based on Frequency Domain Fingerprinting of Network Traffic

Yuchen Sun,[1,2] **Xiangyang Luo**,[1,2] **Han Wang**,[1,2] **and Zhaorui Ma**[1,3]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, Henan, China*
[2]*Key Laboratory of Cyberspace Situation Awareness of Henan Province, Zhengzhou 450001, Henan, China*
[3]*Zhengzhou University of Light Industry, Zhengzhou 450001, Henan, China*

Correspondence should be addressed to Xiangyang Luo; xiangyangluo@126.com

Received 16 October 2021; Revised 19 December 2021; Accepted 12 January 2022; Published 31 January 2022

Academic Editor: Weizhi Meng

Although the anonymous communication network Tor can protect the security of users' data and privacy during their visits to the Internet, it also facilitates illegal users to access illegal websites. Website fingerprinting attacks can identify the websites that users are visiting to discern whether they are performing illegal operations. Existing methods tend to manually extract the traffic features of users visiting websites and construct machine learning or deep learning models to classify the features. While these methods can be effective in classifying unknown website traffic, the effect of classification in the use of defensive measures or onion service scenarios is not yet ideal. This paper proposes a method to identify Tor users visiting websites based on frequency domain fingerprinting of network traffic (FDF). We extract the direction and length features of circuit sequences in access traffic and combine and transform them into the frequency domain. The classification of access traffic is accomplished by using a deep learning classification model combining CNN, FC, and Self-Attention. In this paper, the proposed FDF method is experimentally validated in common scenarios of Tor networks. The results show that FDF outperforms the existing methods for classification in different Tor scenarios. It can achieve 98.8% and 94.3% classification accuracy in undefended and WTF-PAD defense scenarios, respectively. In the onion service scenario, the accuracy is improved by 4.7% over the current state-of-the-art Tik-Tok method.

1. Introduction

As people's awareness of protecting personal privacy continues to increase, more and more users are beginning to use anonymous communication systems to interact with the outside world. However, many criminals have also used anonymous communication systems to conduct illegal operations. Some criminals have established illegal trading websites. Users can purchase leaked database information and even network attack services through these websites. Driven by interests, a large number of network intrusions have spawned on the Internet. Tor is currently the most popular anonymous communication system, and it provides privacy to over 200 million users every day [1]. Tor protects the anonymity of user access by creating an encrypted link with three-hop relays. These relays are randomly selected, prevented from being traced through the bridge and pluggable transmission [2], and the links are changed periodically as the client accesses the server. Although it is difficult to directly crack the Tor anonymous communication system, previous studies have shown that network traffic analysis can affect the security of Tor [3–19], especially website fingerprint (WF) attacks. When users visit each website, they will generate different network traffic features, such as different numbers of data packets and different traffic burst patterns. In a WF attack, the enforcer intercepts network traffic and extracts the features of the traffic packets in an encrypted connection between the monitored user and the entry node of Tor. The classifier determines whether the intercepted traffic is associated with the website of interest to the enforcer, and if the traffic matches the classifier, it indicates that the monitored user is visiting the website of interest to the enforcer. The WF attack allows the enforcer to determine whether the monitored user is browsing illegal

websites, especially websites that conduct black transactions, which is of great significance for combating illegal crimes.

The original intention of Tor is to provide users with anonymity during data communication. Tor should try to avoid the occurrence of WF attacks so as not to affect its security. Therefore, defense measures against WF attacks are proposed. But for enforcers, due to a large number of illegal activities in Tor, it is necessary to monitor criminals and illegal websites. Therefore, it is necessary to conduct further research on WF attacks on Tor that uses defensive measures. The proposed defense measures basically reduce the bursts in the original Tor traffic and confuse the traffic, which significantly reduces the efficiency of WF attacks. For measures that may be used by Tor in the future, it is important to improve their recognition accuracy. In addition, the onion service [20] is the most secure service provided by Tor, which contains a large number of illegal transactions. WF attacks on Tor networks using onion service are also of interest for research. To access the website in the onion service, users need to establish more complex links and have a more complete security verification mechanism. This makes the access traffic mixed with a lot of traffic noise generated for the purpose of authentication. Existing methods for fingerprinting Tor traffic using the onion service are not yet very effective. These methods can detect the behavioral patterns of users visiting different websites from different features such as timing and direction of traffic. However, none of them can reduce the influence of traffic noise on fingerprint recognition.

To address the shortcomings in existing studies, this paper adopts a frequency domain transformation method to deal with Tor traffic. Unlike the existing studies, the frequency domain processing method can effectively reduce the impact of noise on fingerprint recognition when users access the server. In particular, for scenarios where defensive measures and onion services are used, the impact of noise on fingerprint recognition is greater due to the increased security mechanisms. We achieve more significant results in these environments than the existing methods.

The contributions of our work are as follows:

(1) We propose FDF, a fingerprint recognition method for websites based on DWT frequency domain processing. We compared several frequency domain processing methods and found that the wavelet transform works best through theoretical as well as experimental analysis. Due to the properties of the frequency domain transform, we combine for the first time the signal element sequence direction as well as length features for the input of a deep learning model.

(2) We have improved the DF [15] model. The Self-Attention module is added to the original model to support intelligent and efficient analysis of website traffic. In the closed-world scenario (we assume that the monitored user only visits the websites we are interested in. The performance of the classifier can be observed more clearly through the closed world), the classification accuracy on Undefended, WTF-PAD [21], and Onion Service [20] datasets are better than the existing models. Especially for the Onion Service [20] dataset, the accuracy of FDF has reached 70.7%, while the accuracy of the current state-of-the-art Tik-Tok [18] method is only 66.0%.

(3) We evaluated FDF in a more realistic open-world scenario (we assume that monitored users can randomly visit different websites. These sites can be sites that we are interested in or sites that we are not interested in. Through the open world, a more realistic environment can be simulated), where we collected a dataset containing 40,000 unmonitored websites and achieved more desirable Precision and Recall in both undefended and WTF-PAD [21] environments, indicating that FDF is effective in real environments.

The remaining sections of this paper are organized as follows. The second part is the background and related work, which describes the existing approaches to the problem of website fingerprinting for the Tor system. The third part is the problem description, describing the process of fingerprint identification of the website. The fourth part introduces the FDF attack process in detail and explains the key steps of the model in principle. The fifth part introduces the experimental dataset, comparative experiments, and experimental results. The sixth part discusses the problems in the experiment of the method proposed. Finally, the seventh part is the conclusion of this paper.

2. Background and Related Work

For website fingerprint attacks, the data processing method of network traffic and the choice of classifier have a significant impact on its attack efficiency. The website fingerprinting methods that have performed well in recent years are shown in Table 1. In the earliest research, researchers used machine learning to classify website traffic [5–13]. The WF attack was first evaluated by Herrmann et al. [5] in 2009. In 2011, Panchenko et al. [6] used the Herrmann et al. dataset and employed an SVM classifier to classify Tor network traffic by various features such as packet traffic and time. The K-NN attack was proposed by Wang et al. [7] in 2014. The method employs a K-Nearest Neighbor (K-NN) classifier that uses a combination of features to evaluate the similarity between different websites through a distance metric. However, this method is not effective in reducing the impact of noise on WF attacks. In 2016, Panchenko et al. [8] proposed a novel WF attack CUMUL against Tor based on the cumulative representation of traces. The method considers the effect of real noise on WF attacks. However, the problem of overfitting occurs in the process of actual classification. In 2016, Hayes et al. [9] proposed a website fingerprinting attack K-FP based on random decision forest. This method uses a random forest to extract fingerprints for each traffic instance, uses Hamming distance to calculate the distance between these fingerprints, and finally classifies them by k-nearest fingerprint technology. This method shortens the classification time and reduces the impact of overfitting on the classification results. However, the noise will have a certain impact on K-FP.

In recent years, with the massive application of deep learning on WF attacks [14–18], the performance of WF attacks has been further improved. Sirinam et al. [15]

Table 1: Well-performing website fingerprinting methods.

Method	Accuracy (%)	Advantage	Disadvantage
K-NN [7]	95.0	Multiple features are used.	The noise is not considered.
CUMUL [8]	97.3	The noise is considered.	Overfitting.
K-FP [9]	95.5	Sorting time is shortened, and overfitting is reduced.	Classification effect is easily affected by the noise.
DF [15]	98.3	Complex CNN model is proposed.	Large datasets are needed.
TF [16]	95.0	Small datasets are needed.	The accuracy needs to be improved.
Tik-Tok [18]	98.4	Multiple time features are considered.	Classification effect is easily affected by circuit congestion.

proposed the Deep Fingerprinting (DF) attack in 2018. The highlight of DF is the design of complex convolutional neural network (CNN) structures. The deep learning model of DF can solve the WF attack problem well, and the framework of the DF model has been basically borrowed in the subsequent studies. However, DF has a long period to complete the WF attack and requires a large number of training sets to achieve better classification results. The data staleness problem also has an impact on the attack during the data collection. Sirinam et al. [16] studied a triplet fingerprint attack TF in 2019. This attack uses a triplet network for N-shot learning [22]. This method can effectively reduce the workload of data collection and training in the implementation of WF attacks, but the accuracy of the attack needs to be improved. Rahman et al. [18] proposed the Tik-Tok attack based on packet timing information in 2020. The method uses a set of new features based on burst level features related to timing. The information contained in each of these features is mutually exclusive and improves the robustness of the classifier. The method fully considers the timing features so that it can obtain effective information and achieve a high accuracy rate of website fingerprinting under the scenario of using defense. However, due to the instability of the Tor link, it is easy to cause circuit congestion [23, 24]. Circuit congestion can have an impact on the timing information, thus reducing classification accuracy.

In order to make the Tor network more secure, researchers have proposed some defensive measures [21, 25–28] to defend against WF attacks. The basic principle is to operate on packet traffic (add, delete, delay packets, etc.). In order to achieve the purpose of confusing flow features, the BuFLO [25] defense method was proposed by Dyer et al. This method achieves the effect of transmitting data packets close to a constant rate by sending data packets of a fixed length in the Tor network at a fixed time. Juarez et al. proposed WTF-PAD [21]. WTF-PAD is a probabilistic connection filling defense based on adaptive filling. It masks the features of traffic bursts by adding short-delay pseudotraffic bursts, thereby reducing the threat of WF attacks. Wang et al. proposed the Walkie-Talkie (W-T) [28] method. W-T modified the browser to communicate in half-duplex mode instead of the usual full-duplex mode. The half-duplex mode converts the cell sequence into a burst sequence, which not only saves additional overhead but also reduces the characteristics of the cell sequence, thereby leaking less information to the enforcer.

3. The Description of Problem

Tor consists of thousands of relays that form a worldwide network of volunteer overlays to direct Internet traffic. During a user's visit to a website, the traffic is encrypted in multiple layers so that an attacker cannot know which websites the user is visiting. Many illegal websites have emerged in the onion service, where users can log in to complete transactions without being tracked. Therefore, obtaining the websites that users are visiting with the knowledge of their identity is a problem worth investigating.

Although Tor can effectively protect the security and privacy of users, it is still possible to reduce the anonymity of users by means of traffic analysis. A series of associated traffic is generated when users visit a website, and the pattern of this traffic is relatively fixed within a certain period of time. That is, users visiting the same website in the same region within a certain time frame can obtain similar packets. Therefore, the user's access traffic can be analyzed to discern which website the user is visiting. As shown in Figure 1, an enforcer is deployed locally to collect the network traffic between the client and the server and identify the website that the user is visiting. This enforcer can be a router, an Internet Service Provider (ISP), an autonomous service, and so on, capable of arbitrarily collecting encrypted traffic between the client and the entry node. The enforcer cannot discard, modify, insert, and delay packets. If the traffic is tampered with during a user's visit to the site, it may result in errors or anomalies on the user's return page. This not only affects users' browsing but also alerts them to the possibility of their privacy being compromised. Especially for illegal users, it will make it more difficult to collect their incriminating evidence.

For sites of interest to the enforcer, we call them monitored sites. For other types of websites, we call them unmonitored websites. In WF attacks, the enforcer's task is to identify the monitored websites. The enforcer needs to set up a classifier, and in addition to that, he has to loop through the Tor network to the monitored website and collect the traffic during the visit. After the collection is complete, the enforcer has to manually extract the traffic features and construct a traffic matrix from all the processed traffic data for the training of the classifier. When the classifier is trained, the enforcer can passively collect the encrypted traffic during the monitored user's access to the server, process the traffic in the same way as the training set, and then use the classifier to classify the traffic to determine

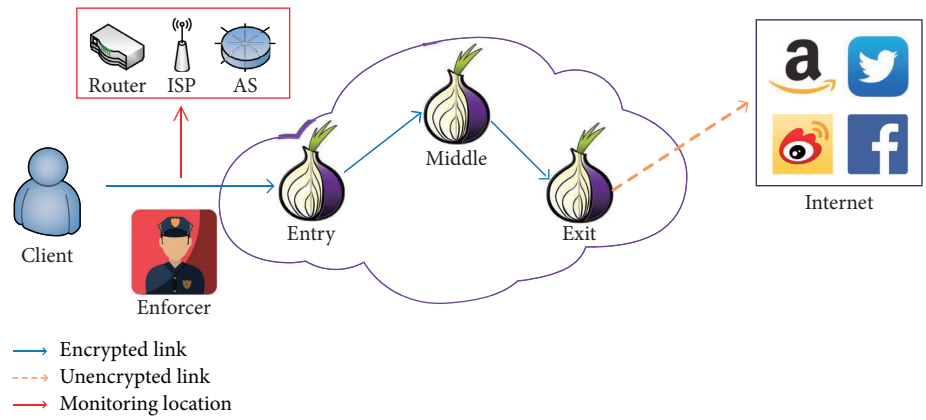

Figure 1: Schematic diagram of website fingerprint recognition process.

whether the website being visited by the monitored user is a monitored website.

In order to improve the readability of the paper, we summarize and explain the notations in our method, as shown in Table 2.

4. The Proposed Method

4.1. The Principle Framework and Main Steps of the Method. In the existing website fingerprint identification methods, the main factor affecting the classification of Tor traffic fingerprints is the noise in the traffic. These noises can effectively confuse the features of the original Tor traffic and reduce its classification accuracy. In response to this problem, we found that frequency domain transformation can reduce the impact of noise on classification and proposed a DWT-based website fingerprint recognition method. The complete fingerprint identification process is shown in Figure 2.

This method is divided into two stages: data preprocessing and classifier classification. Data preprocessing is mainly to extract features from the collected data packets and transform the extracted sequences into the frequency domain to form circuit frequency domain feature sequences. The method of frequency domain transformation can increase the difference of traffic patterns of different websites and can obtain better classification results. Classifier classification focuses on identifying and classifying the preprocessed data using deep learning techniques. Before classifying the unknown traffic, the classifier is trained. This process requires collecting a large number of circuit frequency domain feature sequences and corresponding the sequences to their site labels one by one to generate a training sequence matrix and a training label matrix. After the training is completed, the traffic to be tested is converted into a test sequence matrix for classifier classification. This method uses a deep learning classification model combining CNN, FC, and Self-Attention and uses various regularization techniques in the model to prevent overfitting in the website recognition process.

Step 1. Capture the traffic packets of users visiting the website. Capture the background traffic during the user's visit to the website and generate the raw traffic packets.

Step 2. Extract the feature sequence of the circuit. Extract the direction and length information of the sequence of circuits in the raw network traffic packets, and combine them to form the feature sequence of circuits.

Step 3. Generate the frequency domain feature sequence of the circuit. The feature sequence of the circuit is transformed into the frequency domain feature sequence of the circuit by DWT transformation, and the low-frequency sequence generated after DWT transformation is retained.

Step 4. Store the data into the database. Store the frequency domain feature sequences of the circuits and their corresponding site labels into the database.

Step 5. Generate training set. The frequency domain feature sequences of the circuits and the site labels are extracted from the database according to the model training requirements, and the training sequence matrix and the training label matrix are generated.

Step 6. Construction of the model framework. A suitable neural network framework is selected according to the data type and features of the traffic, and a series of overfitting prevention methods are used to improve the accuracy of the model classification.

Step 7. Model training. The deep learning model is trained using the above matrix. The appropriate hyperparameters are selected through training.

Step 8. Generate the test set. Extract the frequency domain feature sequence of the circuit to be tested from the database and generate the test sequence matrix.

Table 2: List of notations.

Notations	Description
Seq_{dir}	The direction sequence of the circuit.
Seq_{len}	The length sequence of the circuit.
Seq_{mix}	The feature sequence of the circuit.
$x(n)$	The feature sequence of the original circuit.
α	The number of layers of DWT decomposition.
$x_{\alpha,L}(n)$	The low-frequency sequence generated after DWT transformation.
$x_{\alpha,H}(n)$	The high-frequency sequence generated after DWT transformation.
$L(n)$	The low-pass filter.
$H(n)$	The high-pass filter.
Q	The downsampling multiples.
N	The length of the circuit feature sequence.

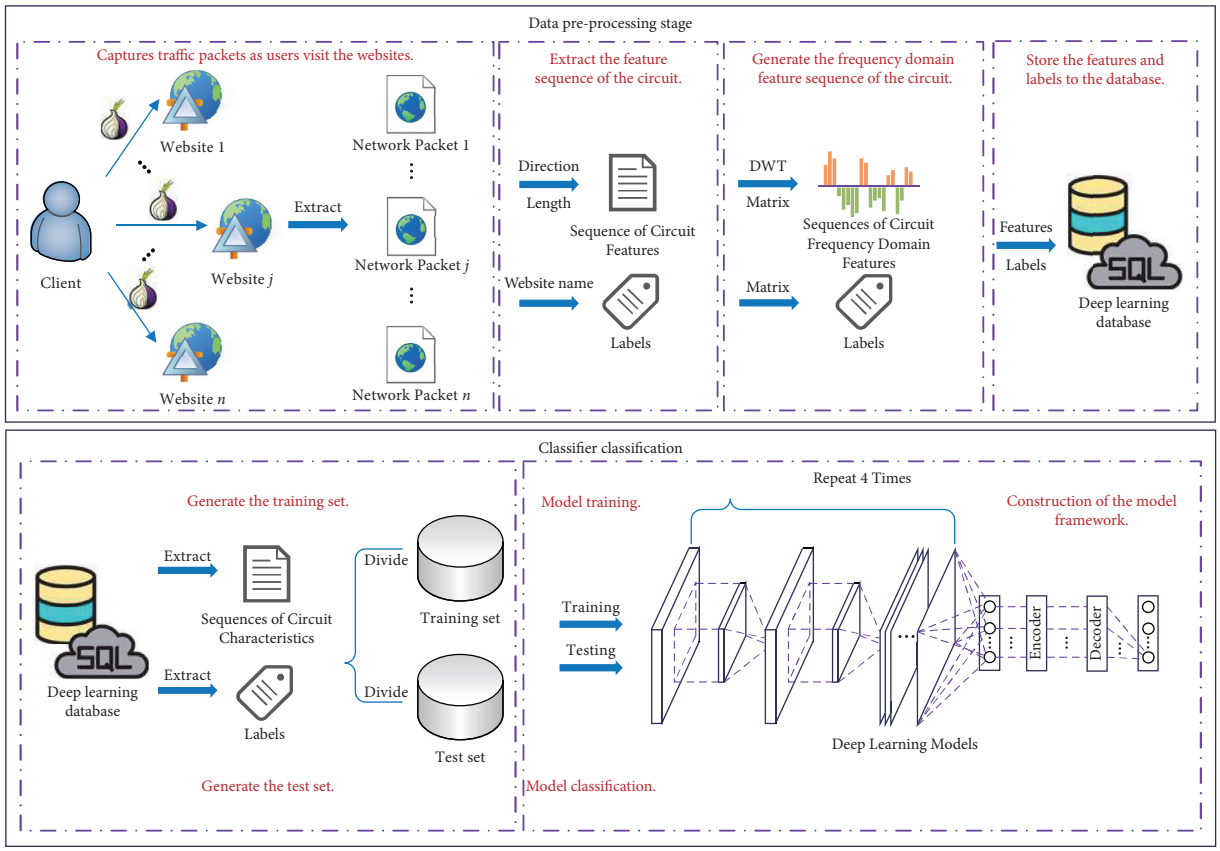

Figure 2: Identification process of Tor users visiting websites based on network traffic frequency domain fingerprinting.

Step 9. Model classification. Use the model to predict the test sequence matrix and obtain the website labels corresponding to the frequency domain feature sequences of the circuit to be tested. Complete the identification of the unknown traffic and correspond the traffic to the website.

4.2. Data Preprocessing

4.2.1. Extract the Feature Sequence of the Circuit. By capturing the packets during a user's visit to a website, we can obtain the circuit sequence of the packets of this website. By analyzing this sequence, we can extract various features, including the direction, length, timing, and burst of the circuit sequence. We select the direction and length of the sequence as the key features to be extracted.

Direction. The sequence of the original circuit is mapped into the value domain of [+1, −1], and the enforcer is usually monitored before the entry relay. Specify the direction of data inflow into the enforcer as "+1" and the direction of data outflow from the enforcer as "−1." By this method, the direction sequence of the circuit Seq_{dir} is formed.

Length. Each packet in the sequence of the circuit is packaged by the protocol before transmission. Clients and servers interact with each other through TCP protocol, so packets that do not contain TCP protocol are to be filtered

out. Next, the length of the TCP protocol layer circuit is extracted to compose the length sequence of the circuit Seq_{len}.

The length sequence and direction sequence of the circuit are combined, and a new feature sequence Seq_{mix} is constructed by multiplying the terms of these two sequences.

$$\text{Seq}_{\text{mix}} = \text{Seq}_{\text{dir}} \times \text{Seq}_{\text{len}}. \tag{1}$$

In a previous study [15], researchers have verified through experimental arguments that the length of the circuit sequence does not significantly improve the accuracy of the attack. A good attack can be achieved by using only the direction of the circuit sequence. However, in our approach, the length of the circuit sequence is necessary. Any time sequence can be seen as an infinite superposition of sine waves of different frequencies and formed. Amplitude is the most basic characteristic of a sine wave. If only the direction of the circuit sequence is used, the full information of the sine wave cannot be reflected. Therefore, we consider combining the length and direction of the circuit sequence to be able to achieve better results in the frequency domain transformation.

4.2.2. Generate the Frequency Domain Feature Sequence of the Circuit. The circuit sequence of a packet based on timing can be understood as the result of the variation of the signal over time. The analysis of the circuit sequence in the frequency domain allows us to obtain more useful information. It is possible to analyze the composition of the sequence frequency, more precisely to decompose the sequence into several subsequences. In this way, the internal connection of each packet in the circuit sequence is reflected, and it is convenient to achieve better results in the subsequent neural network training process.

DWT (Discrete Wavelet Transformation) can discretize the scales and translations of the fundamental wavelets. It can analyze the frequency domain features of local time-domain processes and is more suitable for the analysis of nonsmooth processes. The discrete wavelet transform uses a bandpass filter to decompose the circuit sequence into multiple frequency domain components, which greatly reduces the interference of noise and makes the presentation more intuitive. The architecture of the discrete wavelet transform decomposition process for discrete sequences is shown in Figure 3.

$L(n)$ and $H(n)$ represent the low-pass filter and high-pass filter, and their correspondence is shown in relation (2). $\downarrow Q$ denotes the Q-fold downsampling filter. The sequences decomposed at layer α in the architecture can be represented according to the relations (3) and (4). The high-frequency components are extracted in each layer, while the low-frequency components are deployed to the next layer to continue the decomposition. Since Q-fold downsampling is performed at each layer, if the length of the input circuit sequence is N, then the length of both $x_{\alpha,L}(n)$ and $x_{\alpha,H}(n)$ in the α th layer is N/Q^{α}.

$$L(N-1-n) = (-1)^n H(n), \tag{2}$$

$$x_{\alpha,L}(n) = \sum_{k=0}^{K-1} x_{\alpha-1,L}(Q \cdot n - k) L(k), \tag{3}$$

$$x_{\alpha,L}(n) = \sum_{k=0}^{K-1} x_{\alpha-1,L}(Q \cdot n - k) H(k). \tag{4}$$

In our model, we perform a one-layer architectural decomposition of the circuit sequence and set the multiplier of the downsampling filter to 2. Thus, we are able to obtain the sequence decomposition method as shown in relations (5) and (6).

$$x_{1,L}(n) = \sum_{k=0}^{K-1} x(2n-k) L(k). \tag{5}$$

$$x_{1,L}(n) = \sum_{k=0}^{K-1} x(2n-k) H(k). \tag{6}$$

The circuit sequence is processed in the frequency domain using the relation (5) after the feature processing. The frequency domain processing results in a low-frequency sequence $x_{1,L}(n)$ and a high-frequency sequence $x_{1,H}(n)$. $x_{1,L}(n)$ contains the slowly changing part of the circuit sequence. It is the basic frame of the sequence and belongs to the approximate information of the sequence. $x_{1,H}(n)$ contains the rapidly changing part of the circuit sequence. It belongs to the detailed information of the sequence, which contains the noise. We use the low-frequency part $x_{1,L}(n)$, which can represent the contour features of the sequence, for the training of the model. It can reduce the interference of noise in the sequence for fingerprint recognition.

In our method, both $L(n)$ and $H(n)$ are of constant length, independent of the length N of the circuit sequence. We only need the low-frequency part of the wavelet transform. The convolution of the circuit sequence and the filter requires $O(N)$ time complexity. After each layer of convolution, a branch of length $N/2$ is formed. Therefore, the time complexity required for the entire frequency domain transformation process is $O(N)$.

4.3. Classifier Classification

4.3.1. Construction of the Model Framework. Website fingerprinting on Tor is a supervised classification problem. Starting from DF [15], deep learning techniques have achieved good results on the website fingerprinting problem. We have borrowed from these models and made improvements. In DF, two convolutional layers were used before each Max Pooling. The researchers believe that adding more convolutional layers to each Base Model can obtain a deeper network and extract features more efficiently. In our model, each Max Pooling layer is preceded by only one convolutional layer, which can effectively reduce the complexity of the neural network. After the Base Model, we add a Self-Attention layer. The reason for this is that

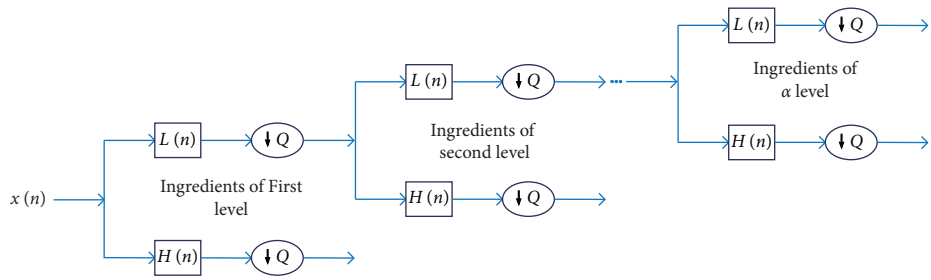

Figure 3: DWT decomposition process.

CNN only considers the information in the receptive field and only acts on a local scale. Self-Attention, on the other hand, considers the information on the entire circuit sequence. It contains a much wider range. Therefore, we consider extracting the local features in the circuit sequence by CNN first and then extracting the global features by Self-Attention, so as to form a complete model. This approach not only reduces the complexity of the neural network but also does not affect the extraction of features. The neural network classification model is shown in Figure 4.

Since neural networks have a fixed input size requirement, for one-dimensional circuit sequences, different lengths of circuit sequences need to be fixed to the same length. After data preprocessing, the circuit sequence length needs to be set to a fixed threshold. Sequences with length less than the threshold are filled with 0, and those with length greater than the threshold are truncated. All circuit sequences are combined to form the input matrix.

To address the selection of hyperparameters in different modules, we empirically assign a range of values to these hyperparameters. For hyperparameters with a small range of values, the hyperparameters are taken iteratively. For hyperparameters with a large range of values, the hyperparameters are taken using the dichotomous method. In the process of model construction, we filtered the hyperparameters module by module and finally obtained the best combination of hyperparameters.

In Tor, many useless data packets are generated when users visit websites due to network congestion, identity verification, and other reasons. This may cause the same user to make multiple visits to the same website in the near time to generate quite different traffic. These noisy data packets can cause overfitting problems during neural network training. For the overfitting problem, we use regularization techniques such as Dropout, Batch Normalization (BN), and Label Smoothing methods. Dropout reduces the interaction between hidden nodes and makes the model more generalizable by making a certain neuron probabilistically stop working. BN normalizes the output results so that the output obeys the standard normal distribution and reduces the internal covariance shift (ICS), which not only helps the network fit faster but also reduces the overfitting problem. Due to the small number of parameters in the convolutional layer, Dropout is rarely used after the convolutional layer, and BN is usually used. In our model, BN is connected immediately after each CNN, and Dropout is used after Max Pooling to prevent overfitting. There are many parameters in the FC and Prediction process, so BN and Dropout can be used together.

In order to approximate the predicted probability distribution to the true distribution during neural network prediction, a common practice is to encode the true labels using the one-hot method. This encoding approach can make the model lack adaptability and be overconfident in its predictions, which can lead to overfitting problems. Label Smoothing smoothes the empirical distribution of the gap between the maximum prediction and the mean of the other categories by adding a smoothing factor. The essence of Label Smoothing is to drive the classification probability results after the activation of the Softmax activation function in the neural network closer to the correct classification, so it is placed in the last part of the model.

4.3.2. Model Training. The selection process of the hyperparameters in the model is shown in Table 3, containing the range of values for each hyperparameter and the value that achieves the best results. We conducted experiments on tuning parameters using the collected Undefended Closed-World dataset and validated them using other datasets, all with good results.

5. Experiments and Results Analysis

To validate the performance of the proposed FDF method, we conducted a series of experiments based on the Undefended, WTF-PAD [21], and Onion Sites [20] datasets.

5.1. Dataset

5.1.1. The Closed-World Dataset. We performed a recursive crawl of the homepages of the top 100 websites ranked by Alexa [31] through the Tor network, with a total of 1000 crawls per site. We deployed the work on LXD containers on ten VPS servers in different countries.

5.1.2. The Open-World Dataset. Since it is not realistic to visit all Internet sites, we selected some of them for simulating the open-world experiment. We visited the top 40,000 websites in Alexa ranking in order. Because these sites are unmonitored sites, they cannot contain the 100 monitored sites collected in the closed-world experiment. We deployed the work in the same ten VPS servers.

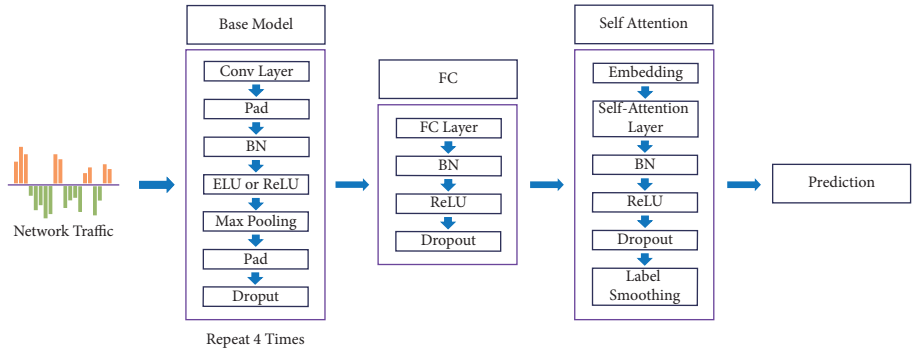

Figure 4: The neural network classification model of WF attack.

Table 3: Selection of hyperparameters for FDF.

Parameters	Search space	Selected value
Input dimension	[500 ... 7000]	5000
Wavelet	Haar, Db, Sym, Coif, Bior, and Rbio	Coif
Base Model	GoogleNet [29], ResNet [30], and DF [15]	DF
Number of FC layers	[1 ... 4]	1
Hidden units (FC)	[256 ... 2048]	512
Hidden dim (Self-Attention)	[128 ... 2048]	256
Optimizer	SGD, adam, adamax, and RMProp	Adamax
Batch size	[32 ... 256]	128
Dropout (Pooling, Self-Attention, and FC)	[0.1 ... 0.8]	[0.1, 0.1, 0.5]

5.1.3. The Onion Service Dataset. A collection of onion domains was conducted by Overdorf et al. [19], and the sites were fingerprinted. They published the dataset used for their experiments to the Internet in the form of tshark logs, which we chose to use for our experiments. Since collecting a large number of onion domains is a difficult task, we chose to use this dataset for experiments.

5.1.4. The Defense Dataset. We performed evaluation tests on the WTF-PAD [15] defense approach. For the WTF-PAD [21] defense, we adapted the raw traffic we collected using a script code posted by the researchers in GitHub. It is used to simulate the traffic generated during access in a real environment according to the defense protocol populated.

A total of five datasets were used in our experiments. In the closed-world scenario, data were collected for undefended, WTF-PAD [21] defense, and onion services, generating the Undefended (CW), WTF-PAD (CW), and Onion Sites (CW) datasets. In the open-world scenario, data collection was performed for both undefended and WTF-PAD [21] defense methods to generate Undefended (OW) and WTF-PAD (OW) datasets. Table 4 shows the website classes and the number of visited website instances in each dataset. We randomly divided each dataset into three parts: training set, validation set, and test set. Due to the large size of the dataset, we divided it according to the ratio of 8 : 1 : 1.

5.2. Website Fingerprinting Experiments on the Closed-World Dataset. The core of the FDF method is to process the circuit sequence in the frequency domain before performing the deep learning fingerprint recognition on the circuit

Table 4: Number of classes and instances in each dataset.

Dataset	Classes	Instances/class	Total
Undefended (CW)	100	1000	100000
Undefended (OW)	40000	10	400000
WTF-PAD (CW)	100	1000	100000
WTF-PAD (OW)	40000	10	400000
Onion Sites (CW)	539	77	41503

sequence. In addition to DWT, Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT) are also mainstream frequency domain processing methods, which have good applications in the direction of image processing. We compared the above three frequency domain processing methods in the closed-world environment. FFT is an efficient and fast algorithm of DFT that can reduce the operation time. Therefore, we use FFT instead of DFT for our experiments.

Table 5 shows the accuracy results of fingerprint recognition on different datasets after processing by three frequency domain processing methods. It can be found that the accuracy rates of the two methods, DCT and FFT, are relatively close to each other. Meanwhile, DWT is significantly better than the other two methods on all three different datasets. This is because circuit sequences are nonstationary signals, and DWT has better results for nonstationary signals, while FFT is more suitable for handling stationary signals.

To show the good attack effect of FDF, we compared it with K-NN [7], K-FP [9], CUMUL [8], DF [15], and Tik-Tok [18] attacks.

Table 5: Comparison of the attack accuracy of three frequency domain processing methods in the closed world.

Method	The accuracy of different dataset		
	Undefended (CW) (%)	WTF-PAD (CW) (%)	Onion Sites (CW) (%)
DCT	98.2	92.6	64.3
FFT	98.3	92.8	65.1
DWT	98.8	94.3	70.7

Table 6 shows the accuracy results of different attack methods in the closed-world scenario for the three environments. It can be found that FDF outperforms the other attacks on the Undefended, WTF-PAD [21], and Onion datasets. Each attack method does not perform very well on the Onion dataset, which we believe is related to the dataset. The Onion dataset has 539 categories with only 77 traffic data per category, which is much less data than other datasets and therefore reduces the accuracy rate.

5.3. Website Fingerprinting Experiments on the Open-World Dataset. To simulate a realistic environment, we conducted experiments in the more realistic open-world scenario. In the open world, the adversary first determines whether the traffic data belongs to monitored or unmonitored sites and second classifies all the traffic belonging to monitored sites according to the limited set of monitored sites.

For open-world scenarios, Precision and Recall were suggested in literature [9, 21] for the evaluation of classifiers. Because the difference between the limited set of monitored sites and the limited set of unmonitored sites may be too large, True Positive Rate (TPR) and False Positive Rate (FPR) can be wrong in the interpretation of the model attack performance.

For the performance evaluation process, we used the standard model proposed in DF Attack [15]. In the open-world scenario, for the dataset of monitored websites, they are trained in the same way as in the closed world. For the dataset of unmonitored websites, it is trained as an additional class. We evaluated undefended and WTF-PAD [21] in the open-world scenario by tuning the attack for Precision and Recall. Precision and Recall are shown in relations (7) and (8).

$$\text{Precision} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}. \quad (7)$$

$$\text{Recall} = \frac{\text{True Positive (TP)}}{\text{True Positive (TP)} + \text{False Positive (FP)}}. \quad (8)$$

Tables 7 and 8 show the results of different methods to tune the attacks for Precision and Recall, respectively, for the two datasets in the open-world scenario. Figure 5 shows the Precision-Recall curves of the attacks in the open world. The above graphs show that DWT shows good results for the Undefended dataset. When attacking tuned for Precision, it achieves 0.99 Precision and 0.94 Recall, while when attacking tuned for Recall, it achieves 0.93 Precision and 0.99 Recall. For the WTF-PAD [21] dataset, the Precision and Recall of all methods decreased due to the added defenses. The best performance was achieved by DWT with a Precision of 0.98 and a Recall of 0.76 when attacking tuned for Precision and a Precision of 0.75 and a Recall of 0.96 when attacking tuned for Recall.

6. Discussion

In this section, we discuss the data preprocessing method before the feature frequency domain processing and the number of DWT decomposition layers for the feature frequency domain processing.

Input for feature frequency domain processing: The most important feature of packets in WF attacks is the direction of the circuit sequence. In DF [15], researchers have also compared packet processing methods and found that the best results can be achieved using only the direction of the circuit sequence. But for frequency domain transformation, not only the direction of the cell sequence but the amplitude of the cell sequence is also very important. If only the direction of the circuit sequence is used and its amplitude is ignored, a lot of information of the original sequence will be leaked during the frequency domain transformation. Thus, the effect of frequency domain processing of the signal cannot be achieved, and the result is degraded. Therefore, we choose to use the direction and length of the signal element sequence as the input for the feature frequency domain processing.

The number of decomposition layers of DWT: in other applications of DWT, multiple layers of wavelet decomposition are often required to achieve better results. Each DWT decomposition results in two components, a high-frequency component, and a low-frequency component. These two components are of the same length. In our method, we use the decomposed low-frequency components every time. In other words, for each layer of DWT decomposition, the length of the circuit sequence is halved. For example, if the input length of the circuit sequence in the FDF model is 5000, a two-layer DWT would require an original circuit sequence length of 20000. We found through statistical analysis that all the original circuit sequences are less than 10,000 in length, so a lot of padding is needed for the circuit sequences. This will have a great impact on the original sequences and lead to a decrease in the accuracy of the classification results. In summary, we choose to perform one layer of DWT decomposition for the feature frequency domain processing. We believe that future additions to the website content and updates to the security mechanisms will result in longer circuit sequences during visits to the website. On this basis, WF attacks using multilayer DWT are expected to achieve a better result.

Table 6: Comparison of attack accuracy between FDF and other methods in the closed world.

Method	The accuracy of different dataset		
	Undefended (CW) (%)	WTF-PAD (CW) (%)	Onion Sites (CW) (%)
K-NN [7]	95.2	16.1	40.9
K-FP [9]	95.6	68.9	45.4
CUMUL [8]	97.5	60.1	47.2
DF [15]	98.3	90.9	53.0
Tik-Tok [18]	98.4	93.5	66.0
Proposed FDF	98.8	94.3	70.7

Table 7: The methods tuned for Precision and tuned for Recall on the Undefended (OW) dataset in the open world.

Method	Tuned for Precision		Tuned for Recall	
	Precision	Recall	Precision	Recall
DF [15]	0.986	0.931	0.929	0.983
Tik-Tok [18]	0.984	0.935	0.918	0.987
Proposed FDF (DCT)	0.973	0.922	0.909	0.981
Proposed FDF (FFT)	0.977	0.931	0.915	0.988
Proposed FDF (DWT)	0.990	0.943	0.931	0.991

Table 8: The methods tuned for Precision and tuned for Recall on the WTF-PAD (OW) dataset in the open world.

Method	Tuned for Precision		Tuned for Recall	
	Precision	Recall	Precision	Recall
DF [15]	0.973	0.736	0.719	0.958
Tik-Tok [18]	0.978	0.751	0.748	0.957
Proposed FDF (DCT)	0.953	0.718	0.692	0.938
Proposed FDF (FFT)	0.961	0.723	0.700	0.947
Proposed FDF (DWT)	0.982	0.756	0.751	0.961

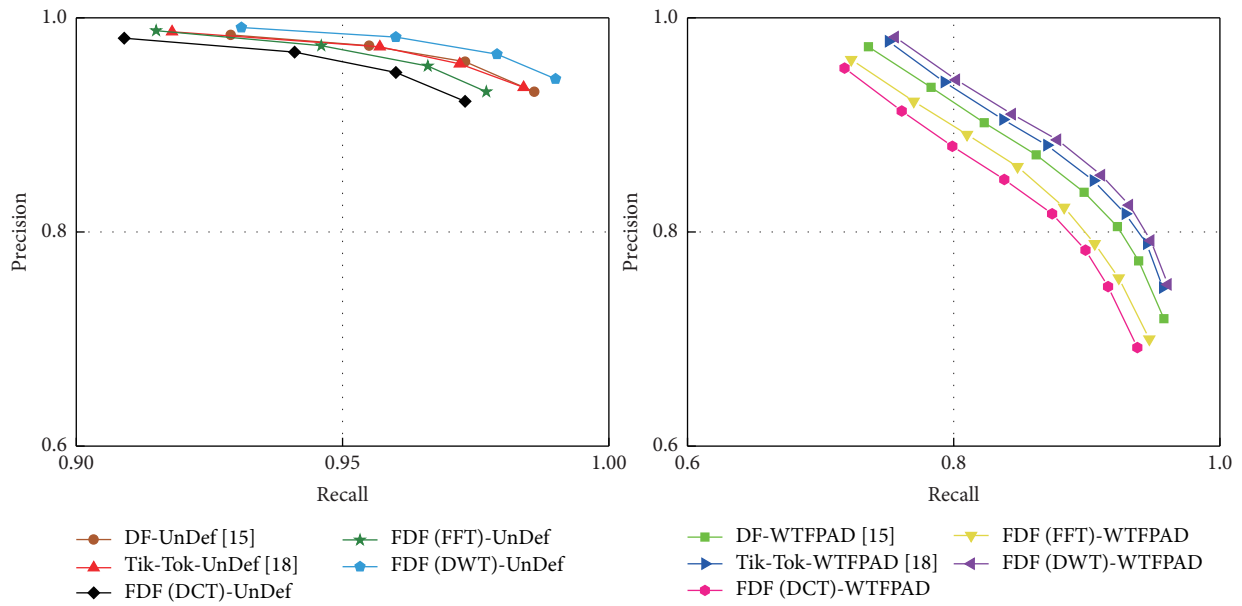

Figure 5: The Precision-Recall curve of attacks in the open world.

7. Conclusion

In this study, we propose an efficient DWT-based WF attack method FDF. We construct key features for traffic analysis by performing DWT on the length and directional features of circuit sequences. After that, we use neural networks to complete the learning and classification of the traffic frequency domain features. Overall, our results show that transforming circuit sequences to the frequency domain for deep learning can achieve good results. However, a large number of training sets are required for data support during the training process. This leads to longer data collection time and increases the difficulty of WF attacks. In the future, we should work to shorten the time to complete the fingerprint identification of the website. One possibility is to learn from the idea of the big data framework [32]. The fingerprint identification process of the website should be layered, especially the data collection process. Collect data through a distributed architecture and reasonably arrange the modules to add new data and delete old data. Ultimately, our methods can effectively respond to urgent tasks [33].

Data Availability

Previously reported Onion Service datasets were used to support this study and are available at DOI:10.1145/3133956.3134005. These prior studies (and datasets) are cited at relevant places within the text as references [12]. The closed-world datasets used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. U1804263, U1736214, and 62172435) and the Zhongyuan Science and Technology Innovation Leading Talent Project (no. 214200510019).

References

[1] "Users—tor metrics," 2010, https://metrics.torproject.org/userstats-relay-country.html.

[2] K. Shahbar and A. N. Zincir-Heywood, "Traffic flow analysis of tor pluggable transports," in *Proceedings of the 11th International Conference on Network and Service Management*, pp. 178–181, IEEE, Barcelona, Spain, November 2015.

[3] A. Montieri, D. Ciuonzo, G. Aceto, and A. Pescape, "Anonymity services tor, i2p, jondonym: classifying in the dark (web)," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 3, pp. 662–675, 2018.

[4] A. Montieri, D. Ciuonzo, G. Bovenzi, V. Persico, and A. Pescap, "A dive into the dark web: hierarchical traffic classification of anonymity tools," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 3, pp. 1043–1054, 2019.

[5] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier," in *Proceedings of the ACM Workshop on Cloud Computing Security*, pp. 31–42, ACM, New York, NY, United States, November 2009.

[6] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pp. 103–114, ACM, New York, NY, United States, October 2011.

[7] T. Wang and I. Goldberg, "Improved website fingerprinting on Tor," in *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pp. 201–212, ACM, Berlin Germany, November 2013.

[8] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *Proceedings of the USENIX Security Symposium*, pp. 143–157, USENIX Association, Sandieago CA, August 2014.

[9] A. Panchenko, F. Lanze, A. Zinnen et al., "Website fingerprinting at internet scale," in *Proceedings of the Network and Distributed System Security Symposium*, Febuary 2016.

[10] J. Hayes and G. Danezis, "k-fingerprinting: a robust scalable website fingerprinting technique," in *Proceedings of the USENIX Security Symposium*, pp. 1187–1203, USENIX Association, Sandieago CA, May 2014.

[11] H. Jahani and S. Jalili, "A novel passive website fingerprinting attack on tor using fast Fourier transform," *Computer Communications*, Elsevier, vol. 96, , pp. 43–51, 2016.

[12] R. Jansen, M. Juarez, R. Galvez, T. Elahi, and C. Diaz, "Inside job: applying traffic analysis to measure tor from within," in *Proceedings of the Network and Distributed System Security Symposium*, Febuary 2018.

[13] A. Shusterman, L. Kang, Y. Haskal et al., "Robust website fingerprinting through the cache occupancy channel," in *Proceedings of the USENIX Security Symposium*, pp. 639–656, USENIX Association, Sandieago CA, November 2014.

[14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, MIT Press, vol. 11, , pp. 3371–3408, 2010.

[15] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1928–1943, ACM, Toronto Canada, October 2018.

[16] P. Sirinam, N. Mathews, M. S. Rahman, and M. 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*, pp. 1131–1148, ACM, London United Kingdom, November 2019.

[17] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: a data-efficient website fingerprinting attack based on deep learning," *Proceedings on Privacy Enhancing Technologies* in *Proceedings of the Privacy Enhancing Technologies*, no. 4, pp. 292–310, Springer, Sandieago CA, Febuary 2019.

[18] M. S. Rahman, P. Sirinam, N. Mathews, K. G. Gangadhara, and M. Wright, "Tik-Tok: the utility of packet timing in website fingerprinting attacks," in *Proceedings on Privacy Enhancing Technologies*vol. 2020, no. 3, , pp. 5–24, Springer, 2020.

[19] R. Overdorf, M. Juarez, G. Acar, R. Greenstadt, and C. Diaz, "How unique is your .onion?: an analysis of the fingerprintability of tor onion services," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2021–2036, ACM, Dallas Texas USA, October 2017.

[20] *Tor: Onion Service Protocol*, https://2019.www.torproject.org/docs/onion-services, 2019.

[21] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *Computer Security - ESORICS 2016*, I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, Eds., vol. 9878, , pp. 27–46, Springer, 2016.

[22] L. Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*vol. 28, no. 4, , pp. 594–611, IEEE, 2006.

[23] R. Dingledine and S. J. Murdoch, *Performance Improvements on Tor or, Why Tor Is Slow and what We're Going to Do about it*, Technical report, The Tor Project, United States, 2009.

[24] P. Dhungel, M. Steiner, I. Rimac, V. Hilt, and K. W. Ross, "Waiting for anonymity: understanding delays in the tor overlay," in *Proceedings of the 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P)*, pp. 1–4, IEEE, Delft, Netherlands, August 2010.

[25] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton, "Peek-a-Boo, I still see you: why efficient traffic analysis countermeasures fail," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 332–346, IEEE, San Francisco, CA, USA, May 2012.

[26] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 227–238, ACM, Scottsdale Arizona USA, November 2014.

[27] X. Cai, R. Nithyanand, and R. Johnson, "CS-BuFLO: a congestion sensitive website fingerprinting defense," in *Proceedings of the Workshop on Privacy in the Electronic Society*, pp. 121–130, ACM, New York, NY, United States, November 2014.

[28] T. Wang and I. Goldberg, "Walkie-talkie: an efficient defense against passive website fingerprinting attacks," in *Proceedings of the 26th USENIX Security Symposium*, pp. 1375–1390, USENIX Association, Vancouver BC, August 2017.

[29] C. Szegedy, W. Liu, Y. Jia et al., "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, IEEE, Boston, MA, June 2015.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, IEEE, Las Vegas, NV, USA, June 2016.

[31] *The Top 500 Sites on the Web*, https://www.alexa.com/topsites, 2016.

[32] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescape, "A big data-enabled hierarchical framework for traffic classification," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2608–2619, 2020.

[33] Wtf-Pad, https://github.com/wtfpad/wtfpad, 2020.