Research Article

Deep Nearest Neighbor Website Fingerprinting Attack Technology

Maohua Guo , Jinlong Fei , and Yitong Meng

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450002, China

Correspondence should be addressed to Jinlong Fei; feijinlong_2021@163.com

Received 27 April 2021; Revised 24 June 2021; Accepted 5 July 2021; Published 15 July 2021

Academic Editor: Weizhi Meng

Copyright © 2021 Maohua Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

By website fingerprinting (WF) technologies, local listeners are enabled to track the specific website visited by users through an investigation of the encrypted traffic between the users and the Tor network entry node. The current triplet fingerprinting (TF) technique proved the possibility of small sample WF attacks. Previous research methods only concentrate on extracting the overall features of website traffic while ignoring the importance of website local fingerprinting characteristics for small sample WF attacks. Thus, in the present paper, a deep nearest neighbor website fingerprinting (DNNF) attack technology is proposed. The deep local fingerprinting features of websites are extracted via the convolutional neural network (CNN), and then the k-nearest neighbor (k-NN) classifier is utilized to classify the prediction. When the website provides only 20 samples, the accuracy can reach 96.2%. We also found that the DNNF method acts well compared to the traditional methods in coping with transfer learning and concept drift problems. In comparison to the TF method, the classification accuracy of the proposed method is improved by 2%–5% and it is only dropped by 3% when classifying the data collected from the same website after two months. These experiments revealed that the DNNF is a more flexible, efficient, and robust website fingerprinting attack technology, and the local fingerprinting features of websites are particularly important for small sample WF attacks.

1. Introduction

As a popular instrument for anonymously browsing the Internet, Tor presents privacy protection services to more than 8 million users each day [1]. Although Tor encrypts the information of routing and communication, it is still not able to hide the unique traffic features of each website. Therefore, some side-channel attack methods are enabling the attackers to infer specific network behaviors by analysis of the anonymous traffic of Tor. The website fingerprinting (WF) attack is the most representative attack technology, in this regard. The idea is that the contents of various websites are different (such as the pictures, web code, scripts, and style sheets). Hence, the anonymous traffic metadata are generated by the different browser loads webpages although the contents of the communication in the Tor network are encrypted. In the related research [2–10], WF attack is considered a classification problem. First, a unique fingerprinting model is made by attackers for each website and a suitable classifier is built by training based on these fingerprinting characteristics. Then, they can collect users'

access traffic, utilize the classifier to categorize the traffic, and locate the specific websites visited by users.

Regarding machine learning, the effectiveness of WF attacks is mainly based on the algorithm of classifiers and hand-made feature sets. Previous work indicated that using the feature sets composed of traffic bursts and packet lengths, the accuracy of Tor website identification can reach more than 90% through support vector machines (SVMs) [11], k-nearest neighbors (k-NNs) [12], random forest [13], and other classifiers. Using deep learning in the field of traffic recognition, researchers found that the classification accuracy can be improved significantly by applying deep learning models to WF attacks [2–5, 7, 8, 10, 13]. For instance, Rimmer et al. proposed the AWF model in 2017 [10] reached up to 94% accuracy when evaluated in a closed world of 900 websites and the DF model proposed by Sirinam et al. in 2018 [8] achieved an accuracy of more than 98% in a closed world of 95 websites. Furthermore, the models achieved a precision rate and a recall rate of more than 90%. However, using the model proposed in the previous study faced two serious challenges. First, the

traditional deep model needs to collect hundreds of training samples for each monitored website. It will face overfitting problem when the training samples are small after hundreds of thousands of training rounds to make the neural network obtain better fingerprint feature extraction ability. Secondly, the model performs poorly in dealing with the concept drift problem and cannot handle the effect of time on accuracy. The classifier's accuracy drops sharply when classifying the attack on the traffic traces captured by the monitored websites a few days after training. To maintain the performance of the classifier, it is essential to update it over time. Moreover, recollecting, maintaining, and updating the dataset and retraining the model are very time-consuming and memory-consuming. This is unacceptable for attackers.

To address the problems faced by deep learning models, Attarian et al. [7] proposed a website fingerprinting attack method AdaWFPA based on the adaptive stream mining algorithm HoeffdingTree (2019). The method effectively improves the ability of the model to cope with the concept drift problem; however, the initial training still requires proving numerous samples. Sirinam et al. [5] proposed a TF model, which requires only 20 examples per website in a closed world to achieve a classification accuracy of almost 95%. Although the TF model achieved a breakthrough in the small sample WF attacks, it mainly concentrates on designing the neural network structure such as traditional deep learning methods. Such approaches abstract the original fingerprinting features of websites into a high-dimensional website-level hybrid feature through neural network extraction and classification with machine learning classifiers. These methods focused on the overall features of website traffic while ignoring the importance of local features in classification. Moreover, when the TF model classifies the websites not pretrained, "secondary training" on the model is required to maintain the accuracy of classification at a higher level. In this paper, we reconsider the learning goals of neural networks, focused on the local features of website fingerprinting and presented a model, which is more appropriate for small sample WF attacks. The main contributions of our work are as follows:

- (i) A new attack model, deep nearest neighbor fingerprinting (DNNF), is proposed. In this attack method, the deep local fingerprinting of websites abstracted by neural networks is used for classification, and the accuracy rate can reach over 96% under only 20 examples available for each website.
- (ii) A comparative experiment is conducted at the same time. Compared to the TF model, the DNNF model is more appropriate for transfer learning. The accuracy rate is increased by 2%–5% without "secondary training" when faced with a brand-new website fingerprinting classification task.
- (iii) Furthermore, in this study, it is concluded that the DNNF model is more effective to cope with the concept drift caused by the highly dynamic web content. When this model categorizes the fingerprinting data from the same batch of websites after two months, the accuracy rate only drops by 3%.

In summary, in this paper, a novel idea of achieving higher accuracy with small sample data is developed. Compared to the existing methods, the proposed model performs well in coping with model migration and probability drift, which further demonstrates the seriousness of Tor damage resultant from the small sample WF attacks. Ultimately, these results provide new ideas on research directions for WF attacks and how to counter them.

The remainder of this paper is as follows. In Section 2, previous WF attack methods are summarized and reviewed. Section 3 introduces the threat model of WF attacks and the design of the DNNF model. In Section 4, the used dataset and the data representation method are summarized. Section 5 presents the experimental results and analysis of this work. Section 6 discusses the limitations in our work and the future research direction. In Section 7, our main findings are summarized.

2. Background and Related Works

Fingerprinting attacks are widely used in the fields of encrypted traffic identification [14], intrusion detection [15, 16], website fingerprinting, etc. In the WF attack method, passive traffic analysis technology is used [17–19]. An attacker configures a network environment similar to the monitored person and utilizes the same encryption proxy technology to access each target site. They extract, analyze, and compare the created communication traffic features to recognize the real address of the communication peer of the monitored person. The effectiveness of the website fingerprinting attack method was proved in the previous work [11, 20–24].

Herrmann et al. first utilized the WF attack method to evaluate Tor, and then researchers have performed a huge deal of work on Tor's WF attack. Panchenko et al. represented a new feature of unique burstiness of traffic and utilized support vector machine (SVM) for website fingerprinting classification, and the accuracy rate reached 55% [11]. To model website fingerprinting, Wang et al. extracted feature vectors with more than 3000 dimensions and used a weight-based distance metric and a k-nearest neighbor (k-NN) classifier to measure the similarity of website fingerprinting [12]. The CUMUL technology was proposed by Panchenko et al. based on the new feature of cumulative packet size [25]. Hayes and Danezis selected 150 important samples from the feature vector up to 4,000 dimensions to describe website fingerprinting and utilized random forest classifiers to perform website fingerprinting attacks (k-FP) [13]. These techniques prove that an accuracy rate of over 90% can be achieved by making feature sets manually to represent a site and select proper machine learning algorithms for classification and training.

With the development of deep learning technology in the fields of image, voice, and video, researchers started to combine deep learning with WF attacks [2, 3, 5, 6, 8, 10, 26–28] to obtain better experimental results. This paper concentrates on the three most effective methods.

2.1. AWF. Rimmer et al. [10] first applied the deep learning method to website fingerprinting attacks. In the AWF method, each access instance of the website is considered as a sequence consisting of ± 1 , in which the symbol indicates the direction of the data packet. Thus, the data packet in the export direction is $+1$ and the data packet in the entrance direction is -1 . A convolutional neural network (CNN) is utilized as a classifier for website fingerprinting attacks. The AWF technique takes advantage of the self-learning features of deep learning to prevent the feature engineering steps required by the k-FP and CUMUL approaches. In a closed world containing 100 websites, a classification accuracy of 96.3% can be achieved with the AWF method when each website has 2500 visit instances, while a TPR of 70.9% and an FPR of 3.8% can be obtained through the AWF method in an open world.

2.2. DF. A more complex CNN model was designed by Sirinam et al. based on the AWF model [8]. The network has a deeper structure while introducing more convolutional layers and batch normalization (BN). The DF method utilizes the same ± 1 time series as the AWF as the model input. In a closed world composed of 95 websites, the DF method can obtain a classification accuracy of 98.3% along with 95.7% TPR and 0.7% FPR in an open world with a size of 20,000.

2.3. TF. A TF method was proposed by Sirinam et al. [5] to implement WF attacks utilizing small sample training in 2019. A triplet network is designed in the TF method, including anchor (A), positive (P), and negative (N), as a subunit. It employs the same ± 1 time series as the AWF method as the model input and the same neural network structure as the DF method for model training. During training, A and P get closer to each other and A and N are farther apart in the embedding space generated by the model. Hence, the feature vectors generated by the same website traffic are close and those generated by various website traffics are far in distance in the model output space. The accuracy of 95% can be achieved using the trained model as the feature extractor and the k-NN as the classifier when each website provides fewer samples.

3. Proposed Approach

3.1. Attack Model. One of the objectives of Tor is to protect users from directly obtaining information on websites visited by eavesdropper users. The WF attacks' goal is to undermine Tor's protection of users through traffic analysis. In this work, our hypothesis is consistent with previous research work [2, 3, 5, 6, 8, 10, 26–28] revealing a passive and LAN-level attacker.

According to Figure 1, it is assumed that the attacker controls the entry node of the Tor network or can access the connection entity between the user and the entry node (such as Internet service provider, local network administrator, and autonomous system (AS)) [29]. The attacker only records the network data packets transmitted over the

FIGURE 1: The attacker can observe the encrypted communication between the user and the Tor network entrance when a user visits a website through the Tor network.

communication and does not perform modification, insertion, deletion, and other operations on the data packets. Moreover, the attacker cannot decrypt and analyze the content.

3.2. Learning Target. The traditional methods of utilizing deep learning to perform WF attacks are based on training a neural network through several data; hence, the network can learn the potential connections and representation levels between the sample data. The initial “low-level” feature representation is transformed into a “high-level” one through multilayer processing. Then, the websites are classified. For example, in the DF method [8], the data were collected for 1000 visits for each website to train the neural network. However, the classification accuracy of the trained model gradually reduces by increasing the time, and the accuracy decreases by 17%–29% after two months. Simultaneously, by altering the target set of the WF attack, the model should be retrained.

Thus, we try to change the learning target and train the neural network to obtain the ability of “learning to classify,” which is similar to the N-shot learning [30–34] in image recognition.

In Figure 2, S denotes support set containing C different website categories (assuming that the model learns the ability to categorize C websites), and each website category possesses N labeled data. Considering a query set Q at the same time, the unlabeled data are classified in Q according to the query set S . This can be called C -way N -shot classification, where $N < 20$ generally. Such a support set S and a query set Q are constructed in each iteration of deep learning. After tens of thousands of training, the model gets the ability to “learn to classify.” In the attack stage, the training model can be utilized to classify the data in the query set Q based on the support set S for completing the task of small sample classification.

3.3. Deep Nearest Neighbor Fingerprinting. To perform WF attacks through deep learning, the present technique is used to group the last convolution feature into a website-level representation (including the latest TF method) via a global average or a fully connected layer for the ultimate website classification. Here, the website-level representation will lose the identification information. By existing enough samples,

Pretrain

Support set (S)	Query set (Q)
Support set (S)	Query set (Q)
Support set (S)	Query set (Q)
Support set (S)	Query set (Q)

Attack

Support set (S)	Query set (Q)
-----------------	---------------

FIGURE 2: The model pretraining and attack process.

this loss can be recovered in the subsequent training and learning; thus, satisfactory classification results are obtained. However, this loss is irrecoverable for the relatively small training sample for each website; thus, poor final classification is achieved. Moreover, if the local invariance of website fingerprinting is directly utilized to measure the similarity of fingerprinting from one website to another and then to classify them, the effect is still poor. The reason is that the test website may be different from any sample of the same website owing to the effect of intraclass changes and background noises for the small training sample.

Thus, we try to combine the local features of all training samples created by the same website into a sample pool and assess the closeness of the local fingerprinting features of the test website to each class of sample pool (the nearest neighbor search, for example). By this method, the boundaries of the fingerprinting training data are broken on the same website. It presents a rich and flexible representation for each class by creating a collection of the local features of the website sample fingerprinting data; thus, a better classification is achieved under the premise of a small sample.

A deep nearest neighbor neural network (DN4) was proposed by Li et al. [34] based on the analysis of the naive-Bayes nearest neighbor method [35], and the effectiveness of the local features was proved in the N-shot learning problem of image recognition. We used the DN4 network in a more challenging WF attack to propose a new attack method as the deep nearest neighbor fingerprinting (DNNF).

According to Figure 3, there are mainly two parts in the DNNF model including the embedded feature extraction network Ψ and the k-nearest neighbor classifier Φ . The embedded feature extraction network creates deep local fingerprinting features of the website through neural network learning and training. The k-nearest neighbor classifier calculates the similarity of the local fingerprinting features of the test website to each class. The two modules are integrated into a unified network for training.

The convolutional neural network is one of the most widely used deep learning algorithms, and it has been shown to have powerful website fingerprinting feature extraction capabilities in previous studies [5, 8, 10]. In the embedded feature extraction network, any appropriate CNN can be utilized as the module Ψ ; however, the module Ψ only includes the convolutional layer in the neural network, not the

fully connected layer. The reason is that we only utilize the high-dimensional website local fingerprinting features of the output from the convolutional layer in the module Φ . Here, we used a one-dimensional CNN model. In other words, $\Psi(X)$ will be a $h \cdot d$ -dimensional tensor after a given website fingerprinting data X going through the module Ψ . If each feature vector with a length of d is considered as a deep local fingerprinting, a total of $m = h$ deep local fingerprinting with a length of d can be obtained:

$$\Psi(X) = [x_1, \dots, x_m] \in R^{d \cdot m}, \quad (1)$$

where x_i denotes the i -th deep local fingerprinting. This also indicates that m deep local fingerprinting can be created when a website is visited by a user once.

In the nearest neighbor classifier, the module Φ utilizes all the deep local fingerprinting created by a website class to construct the feature space of the class. In this space, the similarity between the deep local fingerprinting of the test website and the class is determined by the k -nearest neighbor (k-NN) algorithm. Specifically, we obtain the deep local fingerprinting of the queried website $\Psi(q) = [x_1, \dots, x_m] \in R^{d \cdot m}$ through the module Ψ . For each local fingerprinting x_i , k local fingerprinting closest to it (k nearest neighbors) $\hat{x}_i^j|_{j=1}^k$ can be found in each category c . Then, we determine the similarity between x_i and each $\hat{x}_i$ (here, cosine similarity is adopted; moreover, other similarity or distance measurement methods can also be used). The sum of $m \cdot k$ similarity values represents the similarity between q and category c . The similarity can be calculated using the following formula:

$$\Phi(\Psi(q), c) = \sum_{i=1}^m \sum_{j=1}^k \cos(x_i, \hat{x}_i^j), \quad (2)$$

$$\cos(x_i, \hat{x}_i) = \frac{x_i^T \hat{x}_i}{\|x_i\| \cdot \|\hat{x}_i\|}.$$

In each round of training, we inserted the support set S and query set Q into the model, to obtain the deep local fingerprinting of all the websites in the support set and query set through the embedded feature extraction module Ψ . Then, the similarity between the q_i in the query Q and each class is calculated. For a C-way N-shot classification task, we can obtain a similarity vector $z \in R^C$, in which the class corresponding to the largest component of z is the prediction of q_i .

3.4. Attack Realization. Our attack process includes two parts including the attack phase and the pretraining phase, for which the steps are presented in Figures 4 and 5, respectively.

3.4.1. Pretraining

Step 1: the attacker collects randomly M examples for each website for model pretraining, where $M = 30$.

FIGURE 3: The deep k-nearest neighbor fingerprinting model.

FIGURE 4: The pretraining steps of the DNNF model.

Step 2: the website data are divided into the support set S and the query set Q according to the C-way N-shot classification task, which is set in the pretraining stage.

Step 3: a support set and a query set are put into the DNNF as a round of training data. To form a sample pool, the support set extracts deep local fingerprinting through the module Ψ , and the query set enters the module to extract deep local fingerprinting. It then enters the module Φ , and the k-NN classification algorithm is used and the query set is predicted.

Step 4: each round of Step 3 is a round of deep learning, and Step 3 is repeated several times. To adjust the

parameters of the model, a loss function and an optimizer are utilized until the end of the training.

3.4.2. Attack Construction

Step 1: N examples are collected by the attacker for each monitored website and they are categorized into the support set (for example, 5 examples are collected by a 100-way 5-shot attack for each website, and the consistency between the monitored website and the website used for pretraining is not necessary).

Step 2: the data are categorized into the query set.

FIGURE 5: The attack steps of the DNNF model.

Step 3: the website fingerprinting data of the support set and the query set are put into the trained DNNF model and predictive classification is performed on the query set.

3.5. Network Structure and Model Parameters. As shown in Table 1, we redesigned a network structure for target tasks for DNNF, referring to the DN4 network structure proposed by Li et al. [34] and combining the experience of processing website fingerprinting data with deep learning [5, 8, 10]. Moreover, we follow the extensive candidate search method [6] to assess and select a model and the best hyperparameters in DNNF.

The base model in Figure 3 represents the CNN model utilized for feature extraction in DNNF. We tested and compared the DF model proposed by Sirinam et al. [6], the GoogleNet [36], ResNet [37] models (commonly utilized in image recognition), and the DN4 model proposed by Li et al. [34]. We found that the DF model has better performance in processing website fingerprint classification. Moreover, the training time is less compared with GoogleNet and ResNet, and the classification accuracy is significantly higher than the DN4 model. Ultimately, we selected an eight-layer convolutional neural network structure, which is the same as the DF model, as the basic model. To better adapt to DNNF, some modules in the DF model are adjusted. The batch normalization layer selects a layer normalization [38] which is different from the batch normalization [39] in the original model. The reason is that the global statistical distribution cannot be reflected by the mean square and variance of the sample when the sample is small. Thus, batch normalization is not appropriate for classification problems with a small sample. While layer normalization is an algorithm independent of batch size, its normalization statistics depend on the number of hidden layer nodes solving the problem of a small sample perfectly. Furthermore, we used LeakyReLU [40] as the activation function. On the one hand, the neuron “death” problem faced by ReLU during the training will not occur with LeakyReLU. On the other hand, compared to ELU, the hyperparameters required to be adjusted for

LeakyReLU are fewer; thus, the training cost is reduced. The total number of parameters of the model is 1,042,560 with the estimated total size of 20.56 Mb.

The K in the k -NN algorithm is the only parameter included in the k -nearest neighbor classifier Φ . By examining and comparing common K values, we found that the effect is best when $K=3$. Moreover, we assessed the two common similarity measurement methods of Euclidean distance and cosine distance in the k -NN algorithm. Cosine distance can present better results since it has similarity ordering semantics based on the mutual object frequency. This kind of semantics is similar to find common traffic bursts; hence, we selected cosine distance as the measurement method finally.

4. Dataset

To facilitate comparison with previous work, in our experiment the two datasets of AWF dataset [5] and Wang dataset [12] are used disclosed in the literature. The data are explained as follows: AWF dataset [5] was constructed by Rimmer et al. in 2016, and the TBB version used is 6.X. This dataset includes three parts of the closed world, open world, and concept drift:

- (i) AWF 100 is a closed-world dataset containing the top 100 Alexa websites, and each website has 2500 examples.
- (ii) AWF 775 is a closed-world dataset composed of top Alexa websites (excluding the top 100). This set includes 775 monitored websites, each with 2500 examples.
- (iii) AWF 9000 is an open-world dataset composed of top Alexa websites. This set contains 9000 unmonitored websites, each with 1 example.
- (iv) AWF 200 is a closed-world dataset comprising the top 200 Alexa websites. It is collected five times. The intervals between the 5 collections and the initial collection are 3 days, 10 days, 4 weeks, 6 weeks, and 8 weeks, respectively, and are marked with AWF

TABLE 1: The parameters of the DNNF network structure.

	Parameters	Search space	Selected all of value
Model Ψ	Base model	GoogleNet, ResNet, DN4, DF	DF
	Number of filters (DF)		
	Block 1 [Conv1, Conv2]	[8, ..., 64]	[32, 32]
	Block 2 [Conv3, Conv4]	[32, ..., 128]	[64, 64]
	Block 3 [Conv5, Conv6]	[64, ..., 256]	[128, 128]
	Block 4 [Conv7, Conv8]	[128, ..., 512]	[256, 256]
	Normalization studies	Batch normalization Layer normalization	layer normalization
Model Φ	Activation functions	ReLU, ELU, LeakyReLU	LeakyReLU
	k-NN (K)	1, 2, 3, 4, 5	3
	Similarity metrics	Euclidean, cosine	Cosine
	Loss function	Cross-entropy loss	Cross-entropy loss
	Optimizer	SGD, Adam, Adamax, RMPProp	Adam

200_3 d, AWF 200_10 d, AWF 200_4 w, AWF 200_6 w, and AWF 200_8 w, respectively. For each website, 100 examples are collected each time.

Wang dataset [3] was constructed by Wang et al. in 2013. They used the 3.X version of TBB, which contained two parts of the closed world and open world.

- (v) Wang 100: a closed-world dataset composed of the monitored websites including 100 websites selected from the list of blocked websites in the United Kingdom, Saudi Arabia, and China. Each website has 90 examples.

Previous work indicates that the direction and length of the data packet are the most important features of website fingerprinting [12, 13]. However, researchers found that utilizing the length in Tor's WF attack cannot significantly enhance the accuracy of the attack. Therefore, we used the same data representation as to the deep learning method for WF attacks [5, 8, 10]. Hence, the size of the data packet and the timestamp are ignored and the traffic is converted into a sequence. The inbound and outbound directions of each data packet are stored, where +1 and -1 represent the outbound and inbound, respectively. Moreover, the sequence length is controlled to a fixed length of 5000 through trimming and 0 padding.

5. Experimental Evaluation

In this section, we conduct a series of experiments to assess the performance of the DNNF attack technique in various scenarios and compare it with the previous small sample WF attack method (TF) utilizing deep learning.

In each experiment, we will examine 10 times and use the average value to represent the final performance. Furthermore, the data in each round of training and test evaluation will be randomly sampled to ensure that the experimental results are not only assessed at specific data points and guarantee experimental objectivity.

5.1. Model Pretraining. Since the objective of the DNNF model is to "learn to classify," the breadth of the sample is the key factor affecting the learning effect. By the small sample class space, the model will overfit. Thus, the dataset at

the pretraining stage must contain several classes. In the previous datasets, the dataset collected by Rimmer et al. [10] contains the majority of websites. We attempted to randomly extract 30 examples from AWF775 for each website to construct the dataset for pretraining.

We conducted the model pretraining based on the design in Section 3.4.1. During the model training, we utilized the Adam algorithm. The initial learning rate is set to 1×10^{-3} , which is reduced by half every 10,000 rounds. Figure 6 shows the loss function curve. When the number of rounds is greater than 50,000, the loss remains the same; hence, the final number of pretraining rounds is set as 100,000. In the pretraining phase, we set the classification task to a 20-way 3-shot. At the attack stage, we take the identification of 100 website categories as an example and test the accuracy of each website when the number of samples is $N = [1, 5, 10, 15, 20]$. Through experiments in Section 5.6, the impact of the setting of the classification task C-way N-shot in the pretraining phase on the classification accuracy in the final attack phase will be discussed.

5.2. Closed-World Evaluation. In the first experiment, we first assessed the performance of the method in the traditionally closed world. In this scenario, a fixed set was used as the website URL in the model training and testing phases; moreover, the mutually exclusive data were used for testing samples and the data used for training samples for each website. Here, the AWF 775 dataset is utilized to randomly select 70 examples for each website (mutually exclusive with the pretraining phase) to create the dataset of the attack phase. We utilized 100 websites as the standard to assess and extract 1, 5, 10, 15, and 20 examples for each website to constitute the support set of the attack stage. The remaining samples are used to constitute the query set. Then, the experiment is performed based on steps in Section 3.4.2 with the trained model in Section 5.1.

In Table 2, it is observed that in a closed environment, the accuracy rate of DNNF can reach 78.3% under each 1 example providing by the website. By increasing the number of samples, the interference of the sample burst features on the predictive classification of the model is gradually

FIGURE 6: The training loss function curve of the DNNF model.

TABLE 2: In a closed-world environment, the performance (accuracy) of the CUMUL, k-FP, TF, and DNNF models when the number of examples $N = [1, 5, 10, 15, 20]$.

Type of experiment	Number of examples N				
	1	5	10	15	20
CUMUL	41.7	72.3	77.9	81.4	84.9
k-FP	35.7	78.4	84.1	86.2	87.8
TF	79.1	92.4	93.9	94.1	94.5
DNNF	78.3	92.7	94.3	95.6	96.2

reduced. The accuracy rate can be increased to 92.7% when the number of examples is 5, and the accuracy rate can reach 96.2% when the number of examples per website is increased to 20. At the same time, we found that the accuracy rate will be stabilized at around 96% when the number of examples $N \geq 15$.

Since more than 20 examples are required for the traditional WF attacks using deep learning for each website, we only compare the CUMUL, k-FP, and TF methods. Table 2 indicates that under a closed condition, the accuracy of the CUMUL and k-FP methods is less than 50% when $N = 1$, which is much lower compared to the TF and DNNF methods. By increasing the number of examples for each website, the classification accuracy of the four models is all improved obviously. However, the accuracy of the TF and DNNF models is still obviously higher compared to the CUMUL and k-FP methods. It is observed that compared to the high-dimensional features extracted by neural networks, it is easier to be influenced by the sudden feature of a single sample when utilizing manual features for classification. Next, the DNNF model is compared with the small sample WF attack model TF proposed in 2019. When $N = 1$, the accuracy of the DNNF model is slightly lower. The DNNF model can achieve higher classification accuracy than the TF model by $N \geq 5$. The accuracy increases more obviously by increasing N . When $N = 20$, the accuracy can be increased by 1.7%. It is observed that in a small sample WF attack, the performance of the website local fingerprinting feature is generally better compared to the website hybrid feature. However, when the number of samples is very small, site-

level hybrid features can better represent the monitored websites. Furthermore, the similarity of burst features of different websites can confuse the classifier's accurate judgment of website categories. When the number of samples gradually increases, the fault tolerance for emergent features is better, and the similarity between website fingerprints can be evaluated more fine-grained using local features to achieve accurate classification of websites.

5.3. Transfer Learning Assessment. Transfer learning [41] is a method of machine learning applying a model trained on one task to another task. In deep learning, the pretraining model is a common method as the starting point of a new model in computer vision tasks and natural language processing tasks, which has been improved to achieve good results. The TF method proposed by Sirinam et al. [5] adopted the same idea. They go through retraining the pretrained model to handle different tasks. It is experimentally demonstrated that retraining the model after freezing all layers of the model except the last layer with softmax can improve the attack accuracy. Hence, we propose to focus on the model's ability to learn to classify.

Therefore, after completing the model pretraining, there is no need to perform "second training" on the trained model to cope with classification tasks similar to those at the pretraining stage. The trained model can be directly used to deal with new tasks.

Similar to the TF method [5], we take into account two attack scenarios including similar but mutually exclusive

datasets and differently distributed datasets. Two experiments are performed to assess the performance of the DNNF model in facing transfer learning tasks and compared the DNNF model and the TF model as well as the traditional model.

5.3.1. Similar but Mutually Exclusive Datasets. In this experiment, we tried to train the model on a dataset and classify various datasets in the testing phase. The training data and test data have similar distribution characteristics. For instance, the data used in a close period are collected with the same TBB version. However, the URL websites examined in the training and testing phases are mutually exclusive. In this experiment, we used AWF 100 as the dataset at the attack stage and extracted 1, 5, 10, 15, and 20 examples for each website as the support set; then, we used the remaining samples as the query set. We performed the experiments following the steps in Section 3.4.2 through the trained model in Section 5.1.

Table 3 reveals that the accuracy of the DNNF model is lower compared to the TF model when there is only 1 example under similar but mutually exclusive datasets in the attack and training phases. Meanwhile, the classification accuracy of the traditional model is only 26.7%. When $N \geq 5$, the DNNF model's accuracy is slightly higher than the TF model. When $N \geq 15$, the accuracy tends to be stabilized around 96.0%, which is 1.8% higher than the TF model.

5.3.2. Differently Distributed Datasets. In this experiment, we take into account the data collected at various times through different TBB versions as the training and test sets. Hence, the data used for training and testing have different distributions. The obvious difference between the Wang dataset and the AWF dataset was proved by Sirinam et al.'s cosine similarity analysis [5]. Thus, we selected Wang 100 as the dataset of the attack phase, extracted 1, 5, 10, 15, and 20 examples for each website as the support set, and used the remaining samples as the query set. Then, we conduct experiments following the steps in Section 3.4.2 through the trained model in Section 5.1.

Table 4 demonstrates that the DNNF model's accuracy under more challenging conditions is only 61.3% when $N=1$ s, which is lower than the 73.1% accuracy of the TF model. However, the highest classification accuracy of the traditional model was 56.3%. When $N=5$, the accuracy of the DNNF model is greatly enhanced, which is still lower than the TF model. Nevertheless, the gap between the two models is reduced significantly to less than 1%. When $N \geq 10$, the accuracy of the DNNF model starts to be higher compared to the TF model. When $N=20$, the accuracy reaches 90.5%, which is 3.5% higher than the TF model.

Based on the results, the traditional deep learning model has poor migration ability when performing website fingerprinting attacks, especially to a completely different set of monitored websites. There is a significant decrease in the classification accuracy. Although the classification performance improves significantly by increasing the number of samples, it is still much lower than the DNNF model. This

TABLE 3: The WF attack performance (accuracy) of TF and DNNF with similar but mutually exclusive datasets.

Type of experiment	Number of examples N				
	1	5	10	15	20
Traditional	26.7	87.6	93.4	95.2	95.1
TF	78.6	92.2	93.5	93.9	94.1
DNNF	76.0	92.5	94.2	95.3	95.9

TABLE 4: The WF attack performance (accuracy) of TF and DNNF with differently distributed datasets.

Type of experiment	Number of examples N				
	1	5	10	15	20
Traditional	8.3	31.4	49.1	51.6	56.3
TF	73.1	84.5	86.2	86.6	87.0
DNNF	61.3	83.6	87.1	89.7	90.5

indicates that the classification accuracy of the traditional deep model is heavily dependent on the number of training samples. The model is not retained by the DNNF when applying to a new task. The sample size of the deep fingerprinting generated by the support set is too small when $N \leq 5$. Hence, the fingerprinting characteristics of the website cannot be well summarized, and the accuracy is relatively lower compared to the TF model. When coping with differently distributed datasets, the gap is particularly obvious. However, the DNNF model is generally superior to the TF model based on transfer learning. It is also a reliable choice for attackers since the DNNF model can adapt to new tasks without model secondary training, thereby reducing model training costs. By increasing the sample size, the local features of website fingerprinting are almost included completely. Thus, the DNNF model achieves higher accuracy compared to the TF model when $N \geq 10$. In the more challenging scenario with differently distributed datasets, the performance of the DNNF model is significantly better compared to the TF model when $N=20$, and the accuracy is over 90%.

5.4. Concept Drift Evaluation. Concept drift [42] indicates that the statistical properties of the target variable change in an unforeseeable manner over time. Thus, by increasing the time, the prediction accuracy of the model generally will be reduced. Juarez et al. [43] first found that the classification accuracy dropped significantly when examining the traffic through the WF attack model after training for 10 days. The reason is that the content of the website keeps changing definitely affecting fingerprinting recognition.

In this experiment, we try to use the same dataset AWF 200 used by Sirinam et al. [5]. It is ultimately formed by two months of data collection from the websites in the same closed world. Hence, the website URL utilized in the training and testing phases during the experiment is a fixed set, and the same TBB version is adopted. The collection time and data training are different between the test data. We consider AWF 200_3d, AWF 200_10d, AWF 200_4w, AWF

200_6 w, and AWF 200_8 w as the five datasets in the attack phase. From the five datasets, 1, 5, 10, 15, and 20 examples are chosen for each website as the support set, and the remaining samples are regarded as the query set. Then, experiments are performed based on the steps in Section 3.4.2 through the trained model in Section 5.1.

As seen in Figure 7, the traditional CUMUL, CNN, LSTM, and SADE models showed a sharp decline in classification accuracy within two months, with CUMUL having the worst performance in coping with the concept drift problem and an accuracy drop of up to 35%. However, the CNN, LSTM, SADE, and CNN models also showed different degrees of decrease in classification accuracy from 17% to 29%, which were significantly lower than the DNNF model overall. DNNF has a high generalization ability facing the concept drift challenge. The classification accuracy of the DNNF model almost remains unchanged in the first 10 days. After two months, the accuracy further dropped up to 8% when $N=1$. However, when $N \geq 10$, the classification accuracy can still reach more than 92% and only drop 4% compared to two months ago. In this method, instead of using a deep model to extract features and output classification probability directly as in the traditional model, we compared the similarity between the measured sample and the local features in the monitored website sample pool during training. We then compared the optimal prediction to obtain the ability to “learn to classify.” It can be concluded that by the fixed monitored website, the attacker only needs to collect 10 examples for each monitored website for a long time after completing the model pretraining. Hence, the accuracy of the attack is maintained at a higher level, while reducing the attack cost effectively.

5.5. Open-World Evaluation. In the previous experiments, we discussed the classification of monitored website traffic, while such an assumption is unrealistic since higher website traffic is included compared to the monitored part in the real environment. Then, we will discuss the performance of DNNF attacks in a more realistic open world, that is, the ability of classifiers to distinguish between nonmonitored and monitored website traffic. We evaluate the model using the precision and recall rates. The precision rate denotes the probability of correctly categorizing the monitored pages within the total identified pages, and the recall rate indicates the probability of monitored pages classified as correctly monitored pages. The two rates affect each other and help to better assess the performance of the model in the open world.

In the previous studies on WF attacks in the open world, the researchers added the traffic data of nonmonitored websites in the pretraining phase to better distinguish the traffic of monitored and nonmonitored websites, which is normally called the standard model. We also used the standard model to conduct an open-world evaluation and utilized the sample data from the nonmonitored websites as additional tags in the pretraining phase. Here, AWF100 and AWF9000 were used as the data generated by the monitored websites and the non-monitored websites, respectively.

FIGURE 7: The comparison of classification accuracy loss of models within 56 days.

The precision-recall curve obtained by the DNNF model in the open world is represented in Figure 8. It is indicated that the accuracy of the DNNF WF attack gets close to 0.9 when $N \geq 10$, and the recall rate is close to 0.85. The accuracy will drop significantly, if the recall rate is adjusted to around 0.96, and it is only 0.7. Table 5 shows the tuning results of the precision or recall rates when $N = [5, 10, 15, 20]$. When the accuracy is adjusted, taking $N=10$ as an example, the accuracy and recall rates of the attack reach 0.907 and 0.826, respectively. The accuracy and recall rates of the attack reach 0.692 and 0.969, respectively, when the recall rate is adjusted. The attacker can tune the system by measuring the accuracy and recall rate. If the target of the attack is to accurately determine visiting a monitored website by a user, the recall rate can be appropriately reduced to enhance the accuracy. If the target of the attack is to identify potential visitors to the monitored websites in the network data, then the recall rate is more important, and the accuracy can be sacrificed appropriately to enhance the recall rate.

5.6. Model Performance Evaluation. Ultimately, we assessed the performance of the model through two experiments. Two aspects are mainly considered including the impact of the training mode set in the pretraining phase (or the size of the C-way N-shot in the training phase) on the classification accuracy of the attack phase and the impact of the size of the monitored website at the attack stage on the accuracy. Here, the number of website categories and the training scale should be the same at the pretraining stage.

In the first experiment, we utilized AWF775 as the dataset in the pretraining phase and AWF100 as the dataset in the attack phase. In the pretraining phase, we modified the training mode C-way N-shot, set $C = [5, 10, 20, 40]$ and $N = [1, 3, 5, 10]$, respectively, and generated 16 models following

FIGURE 8: The accuracy and recall curve of the DNNF model in the open world.

TABLE 5: The tuning results of DNNF in the open world.

Number of examples N	Tuned for precision		Tuned for recall	
	Precision	Recall	Precision	Recall
5	0.863	0.812	0.801	0.887
10	0.889	0.831	0.754	0.931
15	0.907	0.826	0.692	0.969
20	0.896	0.858	0.703	0.967

the steps in Section 3.4.1. In the attack phase, we provide 5 samples for each website as the support set and utilize the remaining samples as the query set. Then, the experimental analysis is conducted with the model generated in the pretraining phase based on steps in Section 3.4.2.

Table 6 reveals that in the pretraining phase, the setting of the classification task will influence the accuracy of the attack phase. Generally, by the higher C and N of the classification task in the pretraining phase, the model is trained better and the accuracy of the attack stage gets higher. Eventually, its value is stabilized between 92.5% and 93.0%. Here, the classification accuracy cannot be improved by altering the values of C and N . Moreover, the larger values of C and N yield more time required by the model in the pretraining phase. We recorded the time required to pretrain the model with an RTX2070 graphics card represented in Table 7. According to Tables 6 and 7 and comprehensive consideration of the attack accuracy and training time, we adopt the mode of 20-way 3-shot in the model pretraining in Section 5.1.

In the second experiment, we divided the AWF100 dataset into 4 datasets based on the number of website categories $C = [25, 50, 75, 100]$ and used them as the four datasets at the attack stage. In the five datasets, 1, 5, 10, 15, and 20 samples are chosen for each website as the support set, and the remaining samples are utilized as the query set. The experiment was conducted with the model trained in Section 5.1 following the steps in Section 3.4.2. Five samples

TABLE 6: The impact of the C -way N -shot classification task set in the pretraining phase on the attack (accuracy).

C-way	N-shot			
	1	3	5	10
5	82.2	84.1	84.9	87.6
10	87.5	90.1	90.4	89.3
20	91.8	92.5	92.7	92.6
40	92.6	92.7	92.8	92.7

TABLE 7: The impact of the C -way N -shot classification task setting on the training time of the DNNF model (time).

C-way	N-shot			
	1	3	5	10
5	23 min	29 min	34 min	51 min
10	55 min	1 h 15 m	1 h 29 m	2 h 16 min
20	2 h 51 min	4 h 3 m	4 h 46 m	7 h 16 min
40	9 h 46 min	13 h 51 min	19 h 26 min	25 h 45 min

are extracted for each website as the support set, the remaining samples are regarded as the query set. Then, the experiment is conducted with the model trained in Section 5.1 following the steps in Section 3.4.2.

Figure 9 indicates that the classification accuracy of the DNNF model reduces by increasing the number of monitored websites from 25 to 100. When $N = 1$, the maximum drop is 10% and the accuracy rate is only 78% with the number of monitored websites increasing from 25 to 100. The decrease is between 2% and 5% when $N \geq 5$. The classification accuracy can be maintained over 95% if $N \geq 15$ when the number of monitored websites is less than 100. In general, there is a close relationship between the accuracy of DNNF attacks and the number of monitored websites, and the number of samples provided by each website. The more monitored websites and the fewer samples provided by each website result in lower accuracy and vice versa. Therefore, attackers can adjust the scale of data collection based on the scale of their monitored websites and their expected accuracy to decrease the attack cost as much as possible.

6. Discussion and Future Works

In this section, the possible limitations in our work and the direction of future work will be discussed.

6.1. Computational Complexity. After comparing various classical CNN network model structures, we finally adopted the DF model proposed by Sirinam et al. as the feature extraction module Ψ . We achieved good experimental results in classification; however, the complexity of the network model was relatively high, and the number of parameters for training was in the order of millions. By increasing the collection of monitored websites and the number of samples, the time and space resources consumed by the model will be multiplied. Designing a network structure with lower complexity is an important problem faced by this method in the process of practical use.

FIGURE 9: The variation curve of DNNF attack accuracy as the number of monitored websites increases.

6.2. Segmentation of Anonymous Network Data. The segmentation of anonymous network data including the representative dataset collected by previous researchers in 2016 and 2013 is utilized in our experiments. To ensure the purity of data in the data collection process, we add a prerequisite of users only opening a web page each time, which is not possible in the real situation. There is a huge deal of noisy traffic accompanied when users open a web page in actual attack scenarios. Attackers need to filter out the noisy traffic in a short time to realize the anonymous network data segmentation.

6.3. Overall Fingerprinting of the Website. The definition of the website fingerprinting attack in this paper is consistent with that in previous research. Hence, it is only for the identification of the fingerprinting of a certain website homepage, not including the hyperlinks and other subpages on the website homepage. However, there are few studies on identifying attacks on the overall website fingerprinting.

6.4. Disruptive against Defensive Technologies. Regarding the destructive nature of defense technology, no specific examples denote destroying the anonymity of the Tor network by the website fingerprinting attack technologies. However, researchers proposed several website fingerprinting defense technologies normally utilizing antimachine learning methods to combat website fingerprinting attacks. Further studies are required to assess whether the DNNF method can maintain a high classification accuracy against the existing defense technologies.

6.5. Website Fingerprint Similarity Metric. To measure the website fingerprinting similarity, although the cosine distance (better than the Euclidean distance) is adopted in this study, it is not clear whether direct use of a linear similarity measurement function like the cosine distance will destroy the information of website fingerprinting features.

Therefore, one of our future works is to design a more suitable similarity measurement function.

7. Conclusion

In this study, we proposed a new type of website fingerprinting attack technology, DNNF, to maintain a high accuracy rate when a small sample is provided by attackers for each website. This technology performs well, particularly when coping with transfer learning and probability drift problems. It allows attackers to apply the trained model to new website fingerprinting classification tasks without secondary training and maintains an accuracy of over 90%. Moreover, the accuracy only drops by 3% for the same website dataset, when classifying the data collected two months later. Furthermore, we proved that the DNNF method is effective in the open world. These results demonstrate that attackers can still obtain effective website fingerprinting attacks with fewer resources.

Data Availability

The data 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 Key R&D Program of China (no. 2019QY1302, 2019QY1305). The authors would like to express their gratitude to EditSprings (<https://www.editsprings.com/>) for the expert linguistic services provided.

References

- [1] A. Mani, T. Wilson-Brown, R. Jansen, A. Johnson, and M. Sherr, "Understanding tor usage with privacy-preserving measurement," in *Proceedings of the Internet Measurement Conference 2018*, pp. 175–187, Boston, MA, USA, 2018.
- [2] M. S. Rahman, P. Sirinam, N. Mathews, K. G. Gangadhara, and M. Wright, "Tik-Tok: the utility of packet timing in website fingerprinting attacks," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 3, pp. 5–24, 2020.
- [3] T. Pulls and R. Dahlberg, "Website fingerprinting with website oracles," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 1, pp. 235–255, 2020.
- [4] Y. Meng and J. Fei, "Hidden service website response fingerprinting attacks based on response time feature," *Security and Communication Networks*, vol. 2020, Article ID 8850472, 21 pages, 2020.
- [5] 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, London, UK, 2019.
- [6] 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*, vol. 2019, no. 4, pp. 292–310, 2019.

- [7] R. Attarian, L. Abdi, and S. Hashemi, "AdaWFP: adaptive online website fingerprinting attack for tor anonymous network: a stream-wise paradigm," *Computer Communications*, vol. 148, 2019.
- [8] 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, Toronto, Canada, 2018.
- [9] Z. Zhuo, Y. Zhang, Z.-L. Zhang, X. Zhang, and J. Zhang, "Website fingerprinting attack on anonymity networks based on profile hidden Markov model," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1081–1095, 2017.
- [10] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," 2017, <https://arxiv.org/pdf/1708.06376.pdf>.
- [11] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society*, pp. 103–114, Chicago, IL, USA, 2011.
- [12] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *Proceedings of the 23rd USENIX Security Symposium*, pp. 143–157, San Diego, CA, USA, 2014.
- [13] J. Hayes and G. Danezis, "k-fingerprinting: a robust scalable website fingerprinting technique," in *Proceedings of the 25th USENIX Security Symposium*, pp. 1187–1203, Austin, TX, USA, 2016.
- [14] C. Wang, J. Dani, X. Li, X. Jia, and B. Wang, "Adaptive fingerprinting: website fingerprinting over few encrypted traffic," in *Proceedings of the 11th ACM Conference on Data and Application Security and Privacy*, pp. 149–160, Baltimore, MD, USA, 2021.
- [15] W. Li and L. F. Kwok, "Challenge-based collaborative intrusion detection networks under passive message fingerprint attack: a further analysis," *Journal of Information Security and Applications*, vol. 47, pp. 1–7, 2019.
- [16] W. Li, W. Meng, L.-F. Kwok, and H. H. S. Ip, "Developing advanced fingerprint attacks on challenge-based collaborative intrusion detection networks," *Cluster Computing*, vol. 21, no. 1, pp. 299–310, 2018.
- [17] A. Hintz, "Fingerprinting websites using traffic analysis," in *Proceedings of the 2002 International Workshop on Privacy Enhancing Technologies*, pp. 171–178, Springer, San Francisco, CA, USA, 2002.
- [18] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy vulnerabilities in encrypted HTTP streams," in *Proceedings of the 2005 International Workshop on Privacy Enhancing Technologies*, pp. 1–11, Springer, Cavtat, Croatia, 2005.
- [19] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical identification of encrypted web browsing traffic," in *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pp. 19–30, IEEE, Oakland, CA, USA, 2002.
- [20] S. Yu, W. Zhou, W. Jia, and J. Hu, "Attacking anonymous web browsing at local area networks through browsing dynamics," *The Computer Journal*, vol. 55, no. 4, pp. 410–421, 2012.
- [21] M. Liberatore and B. N. Levine, "Inferring the source of encrypted HTTP connections," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 255–263, Alexandria, VA, USA, 2006.
- [22] Z. Ling, J. Luo, Y. Zhang, M. Yang, X. Fu, and W. Yu, "A novel network delay based side-channel attack: modeling and defense," in *Proceedings of the 2012 IEEE INFOCOM*, pp. 2390–2398, IEEE, Orlando, FL, USA, 2012.
- [23] L. Lu, E.-C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences," in *Proceedings of the 2010 European Symposium on Research in Computer Security*, pp. 199–214, Springer, Athens, Greece, 2010.
- [24] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, pp. 31–42, Chicago, IL, USA, 2009.
- [25] A. Panchenko, F. Lanze, A. Zinnen et al., "Website fingerprinting at internet scale," in *Proceedings of the NDSS*, San Diego, CA, USA, 2016.
- [26] M. Nasr, A. Bahramali, and A. Houmansadr, "Deepcorr: strong flow correlation attacks on tor using deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1962–1976, Toronto, Canada, 2018.
- [27] K. Abe and S. Goto, "Fingerprinting attack on tor anonymity using deep learning," *Proceedings of the Asia-Pacific Advanced Network*, vol. 42, pp. 15–20, 2016.
- [28] S. E. Oh, S. Sunkam, and N. Hopper, "p 1-FP: extraction, classification, and prediction of website fingerprints with deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 191–209, 2019.
- [29] T. Wang and I. Goldberg, "Improved website fingerprinting on tor," in *Proceedings of the 12th ACM Workshop on Privacy in the Electronic Society*, pp. 201–212, Berlin, Germany, 2013.
- [30] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 4077–4087, MIT Press, Cambridge, MA, USA, 2017.
- [31] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: relation network for few-shot learning," in *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, Portland, OR, USA, 2018.
- [32] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12203–12213, Seattle, WA, USA, 2020.
- [33] C. Simon, P. Koniusz, R. Nock, and M. Harandi, "Adaptive subspaces for few-shot learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4136–4145, Seattle, WA, USA, 2020.
- [34] W. Li, L. Wang, J. Xu, J. Huo, Y. Gao, and J. Luo, "Revisiting local descriptor based image-to-class measure for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7260–7268, Long Beach, CA, USA, 2019.
- [35] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, Anchorage, AK, USA, 2008.
- [36] 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, Boston, MA, USA, 2015.

- [37] Y. Taigman, M. Yang, M. A. Ranzato, and L. Wolf, "Deepface: closing the gap to human-level performance in face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, Columbus, OH, USA, 2014.
- [38] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, <https://arxiv.org/abs/1607.06450>.
- [39] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 2015 International Conference on Machine Learning*, pp. 448–456, Lille, France, 2015.
- [40] X. Zhang, Y. Zou, and W. Shi, "Dilated convolution neural network with LeakyReLU for environmental sound classification," in *Proceedings of the 2017 22nd International Conference on Digital Signal Processing (DSP)*, pp. 1–5, IEEE, London, UK, 2017.
- [41] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pp. 242–264, IGI Global, Philadelphia, PA, USA, 2010.
- [42] A. Tsymbal, *The Problem of Concept Drift: Definitions and Related Work*, Vol. 106, Computer Science Department, Trinity College, Dublin, Germany, 2004.
- [43] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 263–274, Scottsdale, AZ, USA, 2014.