Hindawi
Security and Communication Networks
Volume 2021, Article ID 2840289, 13 pages
https://doi.org/10.1155/2021/2840289

Research Article

Few-Shot Website Fingerprinting Attack with Data Augmentation

Mantun Chen,[1] **Yongjun Wang**,[1] **Zhiquan Qin**,[1] **and Xiatian Zhu**[2]

[1]*College of Computer, National University of Defense Technology, Changsha 410073, China*
[2]*University of Surrey, Stag Hill, University Campus, Guildford GU2 7XH, UK*

Correspondence should be addressed to Yongjun Wang; wangyongjun@nudt.edu.cn

Received 7 June 2021; Revised 28 July 2021; Accepted 20 August 2021; Published 16 September 2021

Academic Editor: Weizhi Meng

This work introduces a novel data augmentation method for few-shot website fingerprinting (WF) attack where only a handful of training samples per website are available for deep learning model optimization. Moving beyond earlier WF methods relying on manually-engineered feature representations, more advanced deep learning alternatives demonstrate that learning feature representations automatically from training data is superior. Nonetheless, this advantage is subject to an *unrealistic* assumption that there exist many training samples per website, which otherwise will disappear. To address this, we introduce a model-agnostic, efficient, and *harmonious data augmentation* (HDA) method that can improve deep WF attacking methods significantly. HDA involves both intrasample and intersample data transformations that can be used in a harmonious manner to expand a tiny training dataset to an arbitrarily large collection, therefore effectively and explicitly addressing the intrinsic data scarcity problem. We conducted expensive experiments to validate our HDA for boosting state-of-the-art deep learning WF attack models in both closed-world and open-world attacking scenarios, at absence and presence of strong defense. For instance, in the more challenging and realistic evaluation scenario with WTF-PAD-based defense, our HDA method surpasses the previous state-of-the-art results by nearly 3% in classification accuracy in the 20-shot learning case. An earlier version of this work Chen et al. (2021) has been presented as preprint in ArXiv (https://arxiv.org/abs/2101.10063).

1. Introduction

For privacy protection in accessing the Internet, an increasing number of users have turned to anonymous networks. The Onion Router (Tor) [1, 2] is one of the most popular choices [3].

As a free and open-source software, Tor boosts anonymous communication. It directs Internet traffic through a free, worldwide, and volunteer overlay network with thousands of relays, concealing a user's location and usage from anyone conducting network surveillance or traffic analysis. Concretely, it encrypts the content of communication and sends the data through a route comprised of successive random-selected Tor nodes. However, this remains not completely secure due to exposure of data transportation patterns before reaching Tor servers. For instance, a local attacker would eavesdrop on the connection between a user and the guard node of the Tor network, with the attacking positions including any devices in the same LAN or wireless network, switch, router, and compromised Tor guard node (see Figure 1). By just analyzing the patterns of data packets traffic without observing the content inside, the attacker is likely to reason about which website a target user is visiting. This is often known as *website fingerprinting* (WF) attack [4].

To implement a WF attack, the attacker needs first to create a particular digital fingerprint for every individual website and then learn some intrinsic pattern characteristics of these fingerprints for accomplishing attack. Earlier attacking methods rely on manually designed features based on expert domain knowledge [4–13]. They are not only inflexible but also susceptible to environmental changes over time. This limitation can now be solved by using more advanced deep learning techniques [14]. This is because other than utilizing manually designed features, deep learning methods can automatically learn feature representations directly from training data and are more scalable provided that up-to-date training data are accessible. A

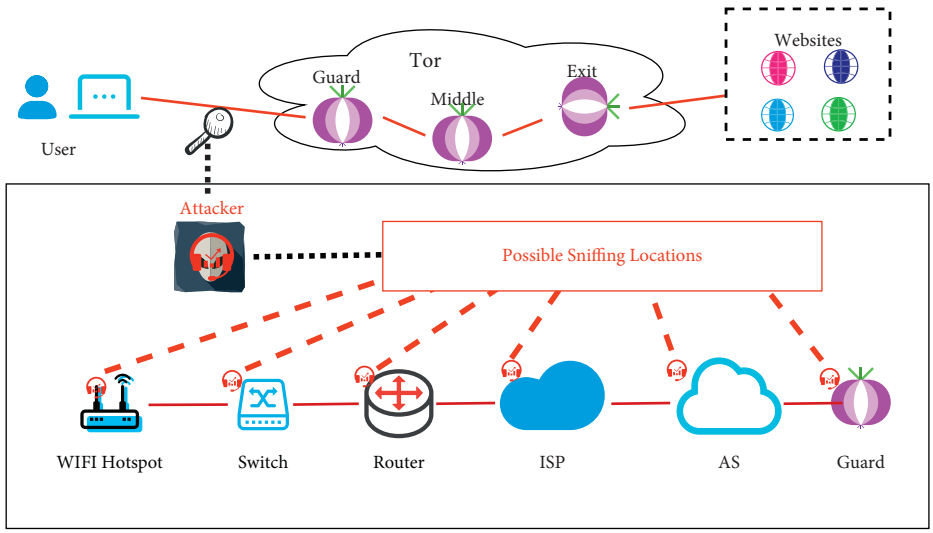

FIGURE 1: Illustration of data flow traffic between a user and target websites with a Tor network in-between. Despite being more secure by anonymity, website fingerprinting attackers are still able to reason about which website a victim user is visiting by analyzing the data traffic characteristics at multiple locations, as specified by red dash lines.

couple of latest state-of-the-art studies, deep fingerprinting (DF) [15] and Var-CNN [16], have demonstrated this potential in comparison to manual feature-based methods. However, these deep learning solutions are not perfect, as their success is established upon an unrealistic assumption that a sufficiently large number (e.g. hundreds) of training samples per website are available, that is, data hungry. When only a small training dataset is given as typical in practical use, their performances are not necessarily superior to traditional methods [11–13]. It is always expensive, tedious, or even infeasible to collect a vast training set in reality due to highly frequent and continuous changes in Internet environments. Consequently, WF attack is fundamentally a *few-shot learning* problem, which nevertheless is largely unrecognized in the literature.

The nature of few-shot WF attack is also considered in the recent triplet fingerprinting method [17], under a condition that there is a large set of relevant auxiliary training samples for model pretraining. It is essentially a transfer learning setting. This will significantly limit its scalability in practice as *in-the-wild* changes of Internet data traffic conditions would render such assumptions to be invalid at high probabilities. On the contrary, we introduce a realistic, generic few-shot WF attack setting where only a handful of training samples are available for every target website, without making any domain-specific assumptions. Clearly, triplet fingerprinting is not applicable in our setting due to the need of auxiliary training data.

We summarize the **contributions** of this paper as follows:

(I) We introduce a novel, practical *few-shot website fingerprinting attack* problem, in which only a few training samples are available without rich auxiliary data. This respects the intrinsic nature of highly dynamic Internet traffic conditions and high cost of collecting extensive training data in practice. Highlighting the importance of *few-shot learning* without any auxiliary data assumption for the first time, we hope more future efforts would be dedicated for solving this practically significant WF attack challenge.

(II) To solve the proposed few-shot learning challenges, we embrace the enormous potentials and advantages of deep learning for WF attack by introducing a new *harmonious data augmentation* (HDA) method to explicitly solve the training data scarcity problem in deep learning. Specifically, we augment the original training data by rotating and masking-out randomly individual samples and mixing (linearly combining) sample pairs in arbitrary proportions. With such intrasample and intersample data transformations, our HDA method can efficiently expand a tiny training dataset at any scales.

(III) We benchmark the performance of few-shot WF attack and demonstrate the efficacy of our data augmentation method using existing state-of-the-art deep learning models. In particular, we consider 5–20 shots per website/class in closed-world and open-world settings, with and without defense. The results show that our method can improve the performances of previous state-of-the-art deep learning solutions [16, 18] significantly.

2. Related Work

2.1. Objectives, Scenarios, and Assumptions. The objective of WF attack is to identify which website a victim user is interacting with among a set of monitored target websites that the adversary is interested in detecting. Conceptually, it is a multiclass classification problem with each website regarded as a unique class. There are several scenarios with

different assumptions. The most common scenario is *closed-world* attack that assumes the user can only visit a small set of websites and that the adversary collects samples to train on all of them. Given that the websites in a *closed-world* setting are far less than in the real world, this assumption is not realistic. In an *open-world* scenario, the victim user is considered to likely visit any other websites including those monitored ones, as typically experienced in real-world applications. As a result, the adversary cannot collect data and train for every website.

The above two scenarios are focused on the range of websites involved in WF attack, independent of WF defense.

The WF defense means that the user takes some actions to defend against a potential attack. This would lead to greater attack difficulty. Representative defense techniques include Buflo [19], Tamaraw [20], Walkie-Talkie [21], WTF-PAD [22], BiMorphing [23], DFD [24], FRONT [25], and GLUE [25]. Among them, WTF-PAD is not the newest defense method, but the main candidate to be deployed in Tor. We considered WTF-PAD-based defense in our evaluations.

In the literature, several common assumptions are made. We briefly discussed three main assumptions. In *user behavior*, it is assumed that all Tor users browsed websites sequentially, only opening a single tab at a time. In *background traffic*, it is assumed that the attacker is able to collect all the clean traces generated by the victim's visits against dynamic background traffic. This is increasingly possible, as shown in [26], and the multiplexed TLS traffic can be split into individual encrypted connections to each website. In *network condition*, the attacker is assumed to have the same conditions as the victim, including traffic conditions and settings. To compare with the benchmark results, we follow these general assumptions for fair evaluations.

Instead, we focus on addressing the following assumption. Often, the attacker assumes that the training data fall into a similar distribution as the deployment data. This is a particularly strong and artificial assumption as the network condition is actually changing and evolving frequently. Such a property enforces the attacker to update the training data in order to have a robust attacking model over time. This implies that the attacker is not possible to collect a large set of training data at each time due to high acquiring costs. However, existing WF attack methods often ignore this factor by assuming the availability of large training data. In contrast, we study the largely ignored few-shot learning setting in the WF attack. Specifically, we approach this problem by explicitly solving the small training data issue via synthesizing new labelled training data.

2.2. Website Fingerprinting Attack Methods. The first pioneer attack against the Tor network was evaluated by Herrmann et al. [7] in 2009. It achieved an accuracy of 2.96% using around 20 training samples per website in the closed-world scenario. Later, Wang and Goldberg [10] proposed to represent the traffic data using more fundamental Tor cells (i.e., direction data) as a unit rather than TCP/IP packets. This representation is rather meaningful and informative as it encodes essential characteristics of Tor data. By training a kernel SVM classifier, a ground-breaking performance with 90.9% accuracy was achieved on 100 sites each with 40 training samples. In 2016, Panchenko et al. [13] proposed an idea of sampling the features from a cumulative trace representation and achieved 91.38% accuracy with 90 training instances per website. Hayes and Danezis [12] exploited random decision forests to achieve similar results. A typical design of these above methods is a two-stage strategy including feature design and classifier learning. This is not only constrained by the limitations of hand features but also lacks interaction between the two stages, making the model performance inferior.

Motivated by the remarkable success of deep learning techniques in computer vision and natural language processing [27, 28], several deep learning WF attack methods have been introduced which can well solve the weakness mentioned above. This is because deep learning methods carry out feature learning and classification optimization from the raw training data end-to-end. For example, Rimmer et al. [29] applied deep learning methods (e.g., stacked-denoising autoencoders, recurrent neural networks, and convolutional neural networks to WF attacks, assuming sufficient training data. Later, Oh et al. [30] utilized autoencoder (AE) to generate low-dimensional features to improve the performance of WF attacks. Meanwhile, using a popular neural network architecture called VGG network [31] as the backbone, Sirinam et al. [15] proposed a deep fingerprinting attack (DF) model that attains 90% accuracy on 95 websites. However, this method needs at least a low-data training set (e.g., 50 training samples per website); otherwise, it will suffer from significant performance drop. When using 20 training samples per website, DF can only hit around 80% accuracy.

To overcome this limitation, Bhat et al. [16] developed the Var-CNN model based on ResNet [18] and dilated causal convolution [32, 33]. When small training sets (e.g., 100 samples per website) are available, it achieves superior performance over DF but at dependence on less-realistic time features and less-scalable hand-crafted statistical information. Meanwhile, Rahman et al. [34] focused on how to utilize timing-related features in WF attacks.

A solution to few-shot learning is a recently proposed triplet fingerprinting (TF) method [17]. The key idea of TF is to pretrain a metric model that can measure pairwise distances on new classes. When the pretraining dataset is similar to the target data in distribution, TF can hit the accuracy of 94.5% on 100 websites using only 20 training samples per website. This is a strong transfer learning scenario. However, considering that the dynamics of network conditions is highly unknown and uncontrollable, such a transfer learning assumption is hardly valid in practice. In light of this observation, in this work, we propose a more realistic few-shot learning setting without assuming any auxiliary data with similar data characteristics for model pretraining. Hence, it is more scalable and generic for real-world deployments. Under the proposed more challenging few-shot setting, TF is unable to work properly due to insufficient network initialization.

2.3. Data Augmentation. Data augmentation is an important element in deep learning due to its data-hungry nature [14]. For example, random insertion, random swap, and random deletion for text classification in natural language processing [35], or geometric transformations (e.g., flipping, rotation, translation, cropping, and scaling), color space transformations (e.g., color casting, varying brightness, and noise injection), and interimage mixup [36] for image analysis [37–39]. These previous attempts have shown the significance of different augmenting methods for model performance on the respective tasks. Inspired by these findings, we investigate the effectiveness of training data augmentation extensively by adapting existing operations for deep learning WF attacks in few-shot learning settings. To the best of our knowledge, this is the first attempt of its kind. Crucially, we demonstrate that the existing state-of-the-art deep WF attack method [16] significantly benefits from using the proposed data augmentation operations in varying evaluation scenarios. This result would be encouraging and influential for future investigation of deep learning WF attack methods in particular.

3. Method

3.1. Problem Definition. In website fingerprinting (WF) attack, the *objective* is to detect which website a target user is visiting. The common observations are data traffic traces x produced by one visit to a website y. Taking each website as a specific class, this is essentially a multiclass classification problem. For model training, a labelled training set $D = \{(x_i, y_i)\}_{i=1}^{N}$ is often provided, where $y_i \in \{1, 2, \ldots, K\}$ specifies one of K target websites. Two different settings are often considered in model testing: (1) *closed-world* attack where any test sample is assumed to belong to the target websites/classes, and (2) *open-world* attack where the above assumption is eliminated, i.e., a test trace may be produced by a *nontarget* (unmonitored) website. The latter is a more realistic setting, yet presenting a more challenging task as identifying if a test sample falls into target classes or not is nontrivial.

3.1.1. Feature Representation. For the Tor network, the raw representation of a specific traffic trace consists of a sequence of temporally successive Tor cells travelling between a target user and a website visited. It is derived from TCP/IP data. Specifically, after those TCP/IP packets retransmitted are discarded, TLS records are first reconstructed, and their lengths are then rounded down to the nearest multiple of 512 to form the final sequence data $\mathbf{x}$. In value, each $\mathbf{x}$ is a sequence of 1 (outgoing cell) and −1 (incoming cell), with a variable length. This raw representation is hence known as the direction sample. Besides, temporal information about interpacket time is another modality of data used, but limited by high reliance on network conditions, i.e., not stable and much more noise. Consequently, we mainly consider the direction data samples in this study, which are more scalable and generic.

3.2. Deep Learning for Website Fingerprinting Attack. Most of existing WF attack methods rely on hand-crafted feature representations [4–13]. This strategy is not only unscalable but also unsatisfactory in performance due to limited and incomplete domain knowledge. Deep learning methods provide a viable solution via learning directly more effective and expressive representation from training data, as shown in a few recent studies [15, 16]. In this work, we advance this new direction further.

1D convolutional neural networks (CNN) [40] are usually explored for WF attacks as the raw data are temporal sequences. Building on the success of deep learning in computer vision, we adopt the same high-level network designs of standard 2D CNN models [41], whilst translating them into 1D counterparts. This is similar to [15, 16].

As shown in Figure 2, a CNN model consists of multiple convolutional layers with nonlinear activation functions such as ReLU [42] and fully-connected (FC) layers, characterized by end-to-end feature extraction and classification. With convolutional operations, the filters of each layer transform input sequences using learnable parameters and output new feature sequences. This feature transformation is conducted layer by layer in a hierarchical fashion. The receptive field (kernel) with size 3 is often used in each layer to capture local feature patterns. By stacking more layers and pooling operations, the model can perceive the information of larger regions and achieve translational invariance. Another effective method for enlarging the receptive field is dilated causal convolutions [32, 33], which has been exploited in [16].

The feature representations $\mathbf{f}$ of WF samples are the output of the global average pooling layer on top of the last convolution layer. To obtain the classification probability vector $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K\} \in \mathscr{R}^K$ over K target classes, $\mathbf{f}$ is fed into a FC layer and normalized by a softmax function.

For model training, we compute a cross-entropy objective loss function with the classification vector against the ground-truth class label over all N training samples as

$$\mathscr{L} = \sum_{i=1}^{N} \sum_{j=1}^{K} \delta(j = y_i) \log \hat{y}_{i,j}, \tag{1}$$

where y_i refers to the ground-truth class label of a training sample $\mathbf{x}_i$ and δ is a Dirac function. The objective is to maximize the probability of the ground-truth class in prediction. This loss function is differentiable, with its gradients backpropagated to update all the learnable model parameters.

Once the deep model is trained, we forward a given test sample, obtain a classification probability vector, and take the most likely class as a prediction in both closed-world and open-world settings. For open-world setting, all unmonitored websites are considered to belong to a background class.

3.2.1. Discussion. While deep learning techniques have advanced significantly in the last several years, it is still assumed that a large set of labelled training samples is

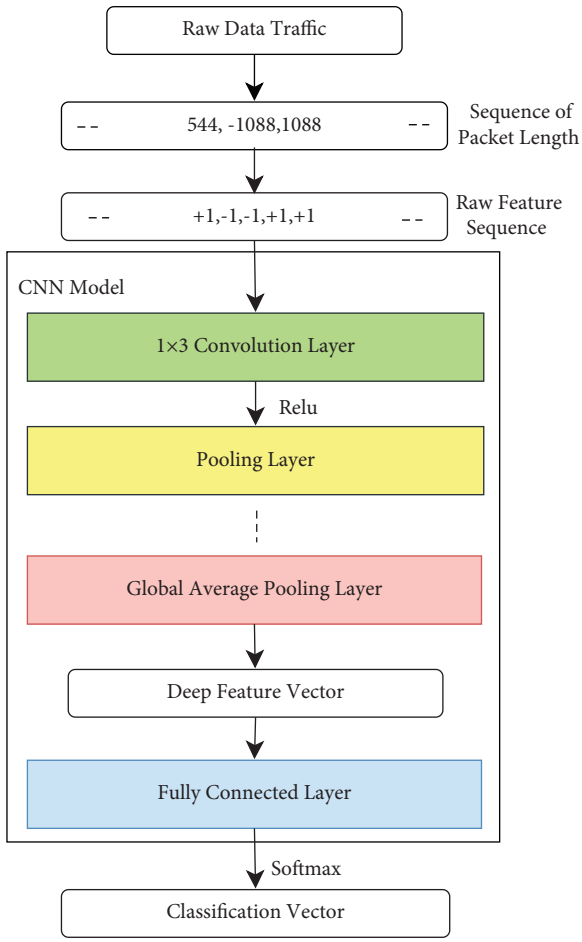

FIGURE 2: A deep learning pipeline for website fingerprinting attack that conducts feature representation and website classification end-to-end in a joint learning manner.

available. This is not always true, for example, for the WF attack problems. In real-world applications, an attacker is usually faced with highly dynamic network environments. It means that the distribution of raw features is evolving continuously. As such, the training data need to update frequently, which disables collection of large training data with labels in practice due to prohibitively high labelling costs. Consequently, only a small training set is accessible in reality, making deep learning methods ineffective.

3.3. Harmonious Website Fingerprinting Data Augmentation. To address the above small training data challenge, we propose an intuitive, novel *harmonious data augmentation* (HDA) method. We introduce both *intrasample* and *intersample* augmentation operations that can be applied in a joint and harmonious manner for more effective data expansion.

3.3.1. Intrasample Augmentation. The key idea of intrasample augmentation is that given an individual training sample, we introduce a certain degree of *random* data perturbation and/or variation whilst keeping the same class labels. Doing so allows us to generate an infinite number of labelled training samples due to the nature of randomness. We consider two perturbation operations: random rotation and random masking.

Random rotation-based data augmentation means rotating an original training sample forward or backward by random steps to generate virtual samples (Figure 3(a)):

$$\text{Rotate}\left(\mathbf{x}, n_{\text{step}}, \text{dir}\right), \tag{2}$$

where n_{step} and $\text{dir} \in \{\text{forward}, \text{backward}\}$ specify the steps and the direction to rotate on an input sample $\mathbf{x}$. The hypothesis behind is that class-sensitive information encoded in a sample is distributed across different subsequences and data traffic order is less important than signal patterns. After a sample is rotated, the original class information is largely preserved, i.e., semantically invariant. Hence, the same class can be annotated for the rotated variants. However, this hypothesis is more likely to stand under some certain (unknown) degrees. We therefore introduce an upper bound parameter R_{max} so that the rotation range is limited at most R_{max} steps in both directions, $n_{\text{step}} \leq R_{\text{max}}$.

In contrast, *random masking* introduces localized corruption to an original training sample by setting a random subsequence to zero (Figure 3(b)). This data augmentation is written as

$$\text{Mask}\left(\mathbf{x}, n_{\text{len}}, \text{loc}\right), \tag{3}$$

where n_{len} and loc denote the length and location of the subsequence that is masked out from an original sample $\mathbf{x}$. Rather than in form of subsequence, another strategy is to randomly select individual positions to mask. We consider this may introduce more significant corruption to the underlying semantic information.

Conceptually, random masking simulates varying traffic measurement errors in data transportation. Meanwhile, with the same above hypothesis, such masking would not dramatically change the semantic class information provided that the masking is subject to some limit, e.g., the length of subsequences masked out M_{len}. It hence offers a complementary data perturbation choice with respect to random rotation.

3.3.2. Intersample Augmentation. Apart from data augmentation on individual samples, we further introduce data perturbation across two different samples to enrich the limited training set.

We propose *random mixing* that generates virtual samples and class labels by linear interpolation between two original samples $\mathbf{x}_i$ and $\mathbf{x}_j$ as

$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j, \tag{4}$$

$$\tilde{\mathbf{y}} = \lambda \mathbf{y}_i + (1 - \lambda)\mathbf{y}_j, \tag{5}$$

where $(\mathbf{y}_i, \mathbf{y}_j)$ are the one-hot class labels of $\mathbf{x}_i$ and $\mathbf{x}_j$. The mixing parameter $\lambda \in [0, 1]$ follows a Beta distribution: $\lambda \sim \beta(\alpha, \alpha)$ with $\alpha > 0$ the parameter that controls the strength of interpolation. This is in a similar spirit of mixup

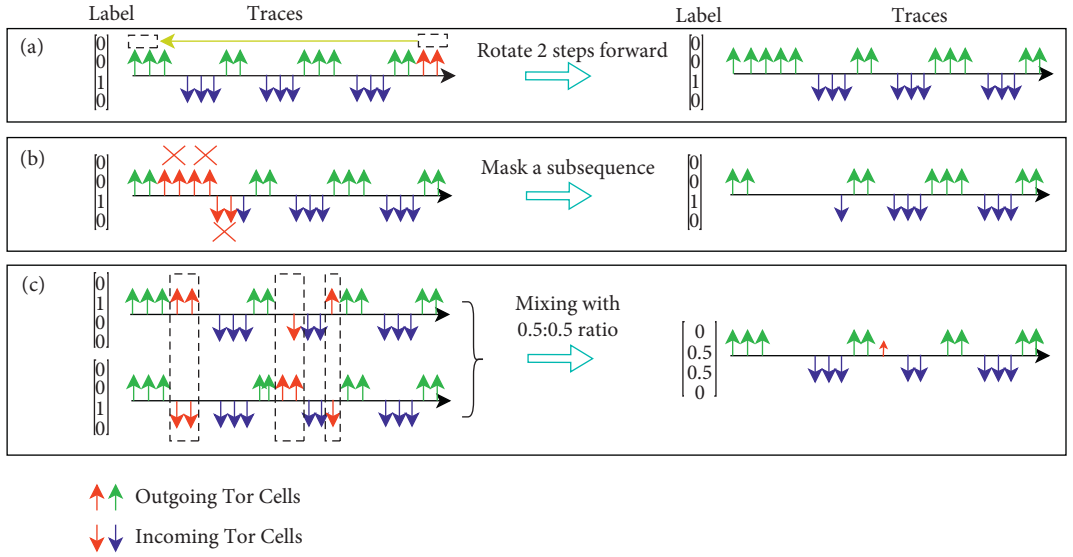

FIGURE 3: Illustration of our data augmentation operations for deep learning WF attack, including (a) random rotation, (b) random masking, and (c) random mixing.

in image understanding domain [36]. Unlike intrasample augmentation above, random mixing changes the semantic class information since original samples may be drawn from different classes. It simplifies the data distribution by imposing a linear relationship between classes for complexity minimization. As shown in Figure 3(c), only the common features are remained in the mixed sample. If two original samples are generated from visiting the same website, the mixed sample reflects the shared characteristics with respect to this website. Otherwise, it reflects the commonality of two different websites.

While seemingly counterintuitive, we will show that such a method brings positive contributions on top of random masking and random rotation.

3.3.3. Combination and Compatibility. Different augmentation operations can be applied on the same samples without conflict to each other in a harmony. There is also no particular constraint on the order of applying all the three data augmentation operations in a combination. Given a fixed set of parameters as discussed above, different augmentation orders will result in different virtual samples. This makes little conceptual difference as the space of sample is just infinite.

3.3.4. Augmentation Optimization. In our harmonious data augmentation (HDA), three hyperparameters $\{R_{\max}, M_{\text{len}}, \alpha\}$ are introduced. To generate meaningful virtual samples, obtaining their optimal values is necessary; otherwise, adversarial effects may even be imposed.

Instead of manual tuning, we adopt an automatic Bayesian estimator, called Tree of Parzen Estimators (TPE) [43]. The conventional TPE can take only a single parameter alone at a time. So, we need to optimize each of the three hyperparameters independently. This differs from our data augmentation process where the three augmentation operations are typically applied together, making the independently tuned parameters of TPE suboptimal. This is because jointly applying three augmentations together makes them interdependent.

For solving this problem, we propose a sequential optimization process that takes into account the interdependence property of different augmentation operations gradually (see Algorithm 1). Specifically, we start with a random, fixed order of applying our random rotation, masking, and mixing operations. Then, we optimize from the first one with TPE, move to the next one with all the previous ones optimized and fixed, and stop by finishing the last one. Each time, we still optimize a single hyperparameter whilst keeping all the previous optimized ones fixed. In this way, we expand the interdependence among different operations sequentially.

3.4. Theoretical Foundation and Formulation. The objective of learning a WF attack model is equivalent to deriving a function $h \in H$ that fits the latent translation relationship between raw feature vectors $\mathbf{x} \in X$ and corresponding website class labels $\mathbf{y} \in Y$, that is, fitting a joint distribution $P(X, Y)$. To this end, in deep learning, we often leverage a loss function L defined to penalize the differences between predictions $h(\mathbf{x})$ and targets $\mathbf{y}$. We minimize the average loss over the joint distribution:

$$R(h) = \int L(h(\mathbf{x}), \mathbf{y}) \mathrm{d}P(\mathbf{x}, \mathbf{y}), \tag{6}$$

which is known as expected risk minimization [44].

However, the joint distribution is often unknown, particularly for WF attacks with small training data. Given a limited training dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$, the joint distribution can only be approximated by an empirical distribution as

Input: A training $\{\mathbf{X_t}, \mathbf{Y_t}\}$, and validation $\{\mathbf{X_v}, \mathbf{Y_v}\}$ set.
Output: Data augmentation with optimal parameters B_{aug}.
1: Setting $B_{\text{aug}} = \phi$ (empty set);
2: Sequencing data augmentation operations randomly;
3: **while** Enumerating augmentation operations **do**
4: Get the search space S_{aug} of current augmentation A;
5: Using TPE on S_{aug} to obtain the optimal parameter b_{aug}, with the model trained by B_{aug} and A;
6: $B_{\text{aug}} = B_{\text{aug}} \cup b_{\text{aug}}$
7: **end while**
8: **return** B_{aug}

Algorithm 1: Data augmentation optimization.

$$P_\delta(\mathbf{x}, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{x} = \mathbf{x}_i, \mathbf{y} = \mathbf{y}_i), \tag{7}$$

where $\delta(\mathbf{x} = \mathbf{x}_i, \mathbf{y} = \mathbf{y}_i)$ is a Dirac mass centered at a sample $(\mathbf{x}_i, \mathbf{y}_i)$. Accordingly, the expected risk can now be approximated by an empirical risk:

$$\begin{aligned} R_\delta(h) &= \int L(h(\mathbf{x}), \mathbf{y})\mathrm{d}P_\delta(\mathbf{x}, \mathbf{y}) \\ &= \frac{1}{N}\sum_{i=1}^{N} L(h(\mathbf{x}_i), \mathbf{y}_i). \end{aligned} \tag{8}$$

The above approximation is in the empirical risk minimization (ERM) principle [44]. The cross-entropy loss (1) is a representative example, which essentially minimizes $R_\delta(h)$ for the classification task.

While ERM is a common strategy, it suffers from a high risk of poor generalization due to the tendency of memorization, mainly when a large model is used [45]. To mitigate this issue, we adopt the notion of vicinal distribution [46] which can better approximate the true joint distribution. In particular, the vicinal distribution P_υ in the data space is defined as

$$P_\upsilon(\tilde{\mathbf{x}}, \tilde{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n}\upsilon(\tilde{\mathbf{x}}_\mathbf{i}, \tilde{\mathbf{y}}_\mathbf{i} \mid \mathbf{x}_i, \mathbf{y}_i). \tag{9}$$

Intuitively, P_υ measures the probability of finding a virtual labelled sample $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ in the vicinity around an original training sample $(\mathbf{x}_i, \mathbf{y}_i)$.

Given such vicinal distributions, we first construct a virtual dataset $D_\upsilon \coloneqq (\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i)_{i=1}^m$ by sampling P_υ randomly and then minimize an empirical vicinal risk to learn h as

$$R_\upsilon(h) = \frac{1}{m}\sum_{i=1}^{m} L(h(\tilde{\mathbf{x}}_i), \tilde{\mathbf{y}}_i). \tag{10}$$

Clearly, at the core of this strategy is performing data augmentation around original training samples. Rather than computing a loss value for every single training sample, it derives a local distribution centered at each individual sample and generates more virtual training samples to reduce the negative memorization effect of deep learning. This is the key rationale of our data augmentation method.

3.4.1. Augmentation Formulation. We formulate the proposed harmonious data augmentation operations in the vicinal distribution manner. For intrasample augmentation (including random rotation and masking), the vicinal distribution is defined as

$$\upsilon(\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \mid \mathbf{x}, \mathbf{y}) = T(\mathbf{x})\delta(\tilde{\mathbf{y}} = \mathbf{y}), \tag{11}$$

where T is a transformation operator.

For *random rotation*, given any length-n sample $\mathbf{x} = \{x_0, \ldots, x_i, \ldots, x_{n-1}\}$, we first define a circle matrix B for forward rotation as

$$B(\mathbf{x}) = \begin{bmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ x_{n-1} & x_0 & \cdots & x_{n-2} \\ \vdots & \vdots & & \vdots \\ x_{n-1} & x_{n-2} & \cdots & x_0 \end{bmatrix}. \tag{12}$$

Then, we sample the step size n_{step} *uniformly* from a range of $\{1, \ldots, R_{\max}\}$. By one-hot representation of n_{step}, we can obtain a rotation transformation as

$$T_{\text{rot}}(\mathbf{x}) = \text{one-hot}\left(n_{\text{step}}\right)B(\mathbf{x}). \tag{13}$$

For the backward case, we perform the same process as above but with a backward rotation matrix instead.

For *random masking*, we similarly sample the start position s *uniformly* in the range of $\{1, \ldots, n - n_{\text{len}}\}$ where n_{len} is the length of the masked subsequence. The masking transformation can be represented by a matrix as

$$M_{\text{mask}} = \text{diag}\left(1 - \sum_{i=s}^{s+n_{\text{len}}} \text{Row}_i(I)\right), \tag{14}$$

where I is the identity matrix, 1 is the all-one vector, $\text{Row}_i()$ selects the ith row of a matrix, and diag() transforms a vector to a diagonal matrix. Masking operation is finally conducted by matrix multiplication as

$$T_{\text{mask}}(\mathbf{x}) = \mathbf{x}M_{\text{mask}}. \tag{15}$$

For intersample augmentation, *random mixing* in our case, the vicinal distribution is defined as

$$v\left(\tilde{\mathbf{x}}, \tilde{\mathbf{y}} \mid \mathbf{x}_i, \mathbf{y}_i, \mathbf{x}_j, \mathbf{y}_j\right) = \delta\left(\tilde{\mathbf{x}} = \lambda \cdot \mathbf{x}_i + (1-\lambda) \cdot \mathbf{x}_j, \tilde{\mathbf{y}} = \lambda \cdot \mathbf{y}_i + (1-\lambda) \cdot \mathbf{y}_j\right), \tag{16}$$

where λ is a random variable drawn from a Beta distribution $\beta(\alpha, \alpha)$ and $\mathbf{y}$ is one-hot class label vector. This local vicinity is assumed to respect a linear structure with respect to class labels.

4. Experiments

4.1. Experimental Setup

4.1.1. Datasets. We evaluated our data augmentation method HDA on four standard WF attack datasets as below. (1) AWF_{100} [29]: this dataset provides a total of 100 monitored target websites, each with 2,500 raw feature traces. (2) Wang_{100} [26]: this dataset gives 100 monitored websites with each contributing 90 feature traces. (3) $\text{DF}_{95,\text{Nodef}}$ [15]: this dataset gives 95 monitored websites with each contributing 1,000 feature traces. (4) **ROWUM** [29]: this dataset includes CW_{100} and a large set of samples, each was generated by a visit to a page of top 400,000 Alexa websites. (5) $\text{DF}_{95,\text{wtf-pad}}$ [15]: unlike all the above datasets, this is a more challenging dataset due to the presence of WTF-PAD-based defense against WF attack. It has 95,000 raw feature samples from 95 websites. We considered both closed-world and open-world WF attack scenarios using the above datasets.

4.1.2. Network Architectures. We used two different network architectures for testing the generic benefits of the proposed HDA method. (1) Var-CNN [16] is the current state-of-the-art deep learning WF method. (2) ResNet-34 [18] is a strong and popular network widely deployed in many different fields such as computer vision.

4.1.3. Implementation Details. We conducted our experiments in Keras [47]. In our experiments, we used the standard training, validation, and test splits for all competitors for fair comparisons. HDA was applied only to the training set. We optimized HDA's hyperparameters using Var-CNN [16] as the deep learning model on CW_{100} in closed-world setting and applied the same parameter setting for all the other deep learning methods, datasets, and settings. This allows testing the generality and scalability of our HDA method. For augmentation optimization, we set the search space as $1 \sim 50$ with step 5 for forward/backward $R_{\max}$ (random rotation) $1 \sim 200$ with step 20 for M_{len} (random masking) and $[0, 1]$ with step 0.1 for α (random mixing). We selected the best value for each of these parameters with respect to the validation performance. The optimal parameter values we obtained are $R_{\max} = 20$, $M_{\text{len}} = 180$, and $\alpha = 0.1$. We applied the same parameter setting tuned on CW_{100} to all other datasets for both simplicity and generalization test.

For saving storage, we performed online data augmentation within each mini-batch without any data preprocessing. In each experiment, we trained every deep learning model for 150 epochs and used the checkpoint with the best performance on the validation set for the model test. We only used the direction feature data, without time sequences and hand-crafted features. We ran each experiment 10 times and reported the mean results and standard deviation as the final performance.

4.1.4. Why Not We Apply HDA to DF? On the one hand, we found that DF is unstable while optimized by HDA. In some experiments, DF + **HDA** can get better results than original HDA, but not always so. On the other hand, the feature extractor of TF is from DF. Hence, we just provide the best result of TF following its recommended setting as baseline.

4.2. Closed-World WF Attack

4.2.1. Setting. We conducted the closed-world attack on AWF_{100}, Wang_{100}, and $\text{DF}_{95,\text{Nodef}}$. We separated each dataset into training and test (70 samples per class) splits. We considered few-shot settings with $n \in \{5, 10, 15, 20\}$ training samples per class. The validation set was used to select the best performing model for test. We used classification accuracy as the performance metric. Besides deep network models, we also compared our method with two conventional hand-crafted feature-based methods: CUMUL [13] and k-FP [12].

4.2.2. Results. The results of different methods are compared in Tables 1–3. We have the following observations: (1) TF remains the best few-shot WF attack algorithm, especially pretrained with similar datasets (pretrained and test with the AWF dataset and test with the Wang dataset). (2) However, deep learning methods (Var-CNN) become clearly stronger when pretrained TF is faced with different distributions across training and testing datasets (pretraining on AWF and testing on Wang_{100} and $\text{DF}_{95,\text{Nodef}}$), suggesting a great deal of potentials. In 10/15/20-shot cases, Var-CNN + HDA achieves the best overall result on both Wang_{100} and $\text{DF}_{95,\text{Nodef}}$. In particular, on $\text{DF}_{95,\text{Nodef}}$, the benefit from HDA is significant, and Var-CNN + HDA surpasses TF with a big margin of 13.2% in 20-shot case. (3) With our HDA method for training data augmentation, every deep learning method improves in all few-shot cases. For example, the 20-shot accuracy of Var-CNN is increased from 78.7% to 90.7% on AWF_{100}, from 88.4% to 90.6% on Wang_{100} and from 68.1% to 91.3% on $\text{DF}_{95,\text{Nodef}}$. Similarly, the 20-shot accuracy of ResNet-34 is improved from 51.3% to 86.4% on AWF_{100}, from 85.9% to 87.4% on Wang_{100} and from 61.4% to 85.8% on $\text{DF}_{95,\text{Nodef}}$. (4) Our HDA can consistently improve different methods on varying datasets, suggesting good generality. (5) The performance deviation of Var-CNN assisted by our method HDA is the least among all the competitors, implying strong stability.

Table 1: Results of *closed-world* WF attack on AWF_{100}. Metrics: accuracy.

Method	5-shot	10-shot	15-shot	20-shot
CUMUL [13]	72.2 ± 1.7	79.7 ± 1.4	83.3 ± 2.0	85.9 ± 0.6
k-FP [12]	79.3 ± 1.0	83.9 ± 1.0	85.9 ± 0.6	87.5 ± 0.8
DF [15]	1.0 ± 0	1.4 ± 0.3	37.3 ± 10.0	70.0 ± 4.4
TF [17]	92.2 ± 0.6	93.9 ± 0.2	94.4 ± 0.3	94.5 ± 0.2
ResNet-34 [18]	14.5 ± 0.7	24.3 ± 1.5	40.3 ± 3.1	51.3 ± 6.4
Var-CNN [16]	17.9 ± 1.5	41.4 ± 4.0	65.6 ± 1.9	78.7 ± 1.5
ResNet-34 + **HDA**	34.8 ± 6.2	62.3 ± 8.1	78.8 ± 7.1	86.4 ± 2.8
Var-CNN + **HDA**	59.7 ± 1.5	74.7 ± 2.6	86.4 ± 1.3	90.7 ± 0.8

Table 2: Results of *closed-world* WF attack on $Wang_{100}$. Metrics: accuracy.

Method	5-shot	10-shot	15-shot	20-shot
TF [17]	84.5 ± 0.4	86.2 ± 0.4	86.6 ± 0.3	87.0 ± 0.3
ResNet-34 [18]	37.9 ± 7.0	60.1 ± 6.1	80.4 ± 0.9	85.9 ± 0.6
Var-CNN [16]	37.4 ± 2.8	72.5 ± 1.8	83.6 ± 1.2	88.4 ± 0.4
ResNet-34 + **HDA**	63.4 ± 6.3	82.6 ± 2.5	85.7 ± 0.7	87.4 ± 0.8
Var-CNN + **HDA**	76.9 ± 2.4	87.1 ± 0.6	89.8 ± 0.4	90.6 ± 0.4

Table 3: Results of *closed-world* WF attack on $DF_{95,Nodef}$. Metrics: accuracy.

Method	5-shot	10-shot	15-shot	20-shot
TF [17]	72.5 ± 0.5	76.4 ± 0.5	77.9 ± 0.3	78.1 ± 0.3
ResNet-34 [18]	22.3 ± 7.9	44.0 ± 2.8	53.7 ± 4.1	64.4 ± 3.8
Var-CNN [16]	21.1 ± 3.4	42.0 ± 5.2	57.6 ± 2.0	68.1 ± 4.8
ResNet-34 + **HDA**	58.2 ± 6.6	77.1 ± 4.0	81.8 ± 1.8	85.8 ± 1.8
Var-CNN + **HDA**	64.8 ± 5.1	85.3 ± 1.0	87.7 ± 1.7	91.3 ± 0.4

4.3. Open-World WF Attack

4.3.1. Setting. We conducted the open-world attack experiments on the combination of $ROWWUM_{400,000}$ and AWF_{100}. We treat the websites of AWF_{100} as target (monitored) classes and those of $ROWWUM_{400,000}$ as nontarget (unmonitored) classes. In this test, we selected randomly 8,020 out of 400,000 unmonitored websites and separated them into three disjoint sets sized at 20/1,000/7,000 for training, validation, and test, respectively. In this scenario, the precision and recall rates were used to evaluate model performance due to the need for detecting nontarget classes [48]. We considered the same two deep learning methods (Resnet-34 and Var-CNN [16]) for comparisons.

4.3.2. Results. The results of different methods are reported in Table 4. We considered two settings, one is tuned for best precision, and one for best recall. Overall, we obtained similar trends as above that our HDA is highly effective for improving both deep learning methods. It is noted that unlike the closed-world scenario, Var-CNN + HDA achieves very top results at most cases under both tuning settings, even if it may not be the best one. Similarly, Var-CNN + HDA remains to be more stable and less sensitive to training sample size. Significantly, our HDA method further enhances these strengths by efficient data augmentation, leading to the more robust WF attack solutions.

4.4. WF Attack against Defense

4.4.1. Setting. In contrast to the two above experiments, we further tested a more challenging WF attack scenario with defense involved. Defense changes the data traffic patterns to be more similar to one another, therefore making the attack more difficult. We considered the most popular defense, WTF-PAD, widely deployed in the Tor network. We used the $DF_{95,wtf-pad}$ dataset in this experiment. We used 100 random samples per website and divided them into three sets for training (20 samples), validation (10 samples), and test (70 samples), respectively. We reported the classification accuracy as performance metric in the closed-world scenario. We help the previous two deep learning methods (Resnet-34 and Var-CNN [16]) with HDA, compared with the pretrained few-shot method (TF [17]) and hand-crafted feature-based methods (*k*-NN [11], *k*-FP [12], and CUMUL [13]).

4.4.2. Results. We reported the results of closed-world WF attack under WTF-PAD-based defense in Table 5. We made the following observations. (1) Some hand-crafted feature-based methods (CUMUL) are superior over recent deep learning methods (ResNet-34 and Var-CNN) at the few-shot learning scenarios. This is mainly because the latter suffers from lacking enough training samples, resulting in model overfitting. (2) Using our HDA for training data augmentation, we can directly solve the data scarcity problem and significantly boost the performances of previous deep learning methods. As a result, Var-CNN + HDA outperforms the other competitors by a moderate margin, e.g., 2.9% gap over the best competitor CUMUL. (3) ResNet-34 is surpassed by Var-CNN continuously. By benefiting more from our data augmentation, Var-CNN achieves the best results across all different shot cases. This implies that Var-CNN has a higher desire for large training data with higher performance potential, as compared to ResNet-34. (4) If TF is not pretrained with a similar dataset, it will lose the advantage when a few more samples (20-shot) are provided.

4.5. Ablation Studies.

We carried out a set of component analysis experiments to examine the exact effect of different designs of our method (HDA). We adopted the most common closed-world attack scenario *without* defense on

Table 4: Results of *open-world* WF attack on AWF_{100} (target classes) + $ROWWUM_{400,000}$ (nontarget classes). Pre : precision and Rec : recall. We reported two settings: one is tuned for best precision (top), and one for recall (bottom).

Method	5-shot		10-shot		15-shot		20-shot	
	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec
Tuned for precision								
ResNet-34 [18]	32.5	4.7	42.1	6.6	50.4	21.8	61.3	39.6
Var-CNN [16]	39.7	2.7	58.8	9.2	74.2	35.8	78.0	54.4
ResNet-34 + **HDA**	72.7	0.8	91.7	8.7	91.5	43.8	92.6	55.1
Var-CNN + **HDA**	77.4	6.9	91.2	47.2	91.4	64.3	92.9	66.6
Tuned for recall								
ResNet-34 [18]	12	25.6	19.2	37.8	30.2	54.5	34.9	68.6
Var-CNN [16]	13.9	27.1	26.7	52.8	37.4	73.9	42.2	82.6
ResNet-34 + **HDA**	21.1	37.4	36.2	68.9	45.7	89.2	47.2	91.4
Var-CNN + **HDA**	28.0	55.8	46.9	88.7	49.3	92.2	49.1	92.5

Table 5: Results of *closed-world* WF attack with *WTF-PAD-based defense* on $DF_{95,wtf-pad}$. Metrics: accuracy.

Method	5-shot	10-shot	15-shot	20-shot
k-NN [11]	—	—	—	16.0
k-FP [12]	—	—	—	57.0
CUMUL [13]	—	—	—	60.3
TF [17]	39.8 ± 0.5	47.2 ± 1.1	50.1 ± 0.4	51.7 ± 0.5
ResNet-34 [18]	7.4 ± 0.5	9.4 ± 0.7	13.3 ± 1.3	12.3 ± 1.2
Var-CNN [16]	6.6 ± 0.3	9.2 ± 0.7	12.5 ± 0.8	19.2 ± 1.7
ResNet-34 + **HDA**	12.3 ± 1.9	28.1 ± 3.7	38.2 ± 6.2	47.7 ± 5.1
Var-CNN + **HDA**	25.3 ± 2.2	46.9 ± 1.9	48.7 ± 1.4	63.2 ± 1.8

the AWF_{100} dataset, following the same setting as Section 4.2. It is noteworthy that this dataset AWF_{100} is different from the dataset in Section 4.2 because they are different subsets. In this section, we evaluated the 15-shot learning case in particular, using Var-CNN [16] as the deep learning model backbone.

4.5.1. Individual Augmentation Operations. Recalling that our data augmentation method (HDA) consists of three different operations (random rotation, masking, and mixing), we have demonstrated their performance advantages of them as a whole in varying test settings above. For in-depth insights, examining their individual contributions would be informative and necessary as well as different combinations. We conducted these experiments with an exhaustive set of operation combinations and reported the results in Table 6.

It is observed that (1) each of the three operations makes a significant difference in performance, with rotation and masking the best individual operations that improve the classification accuracy by 17.4%. (2) When jointly using any two augmentation operations, the performance can be further increased. The combination of masking and mixing gives the highest accuracy among them. (3) Combining all three operations (HDA) achieves the best result with a smaller deviation. This suggests that all different operations are complementary and compatible with each other.

Table 6: Effect of individual augmentation operations.

Augmentation operation	Accuracy
None	75.0 ± 3.0
Random rotation	92.4 ± 0.3
Random masking	92.4 ± 0.8
Random mixing	86.7 ± 0.6
Random rotation + masking	92.7 ± 0.4
Random rotation + mixing	92.6 ± 0.7
Random masking + mixing	93.4 ± 0.7
HDA (**ours**)	93.5 ± 0.4

Table 7: Effect of augmentation optimization.

Augmentation optimization	Accuracy
Independent	92.1 ± 0.5
Sequential (**ours**)	93.5 ± 0.4

4.5.2. Augmentation Optimization. For optimal data augmentation, we propose a sequential optimization strategy (see Algorithm 1) for capturing the interdependence between different augmentation operations applied. To evaluate its effect, we compared with a baseline algorithm that *independently* optimizes each augmentation parameter.

As shown in Table 7, the proposed optimization algorithm (see Algorithm 1) is clearly superior, validating our consideration that there exists interdependence between different augmentation operations when applied jointly on the same samples. Note that we obtained this performance gain at the same cost as the baseline counterpart. Besides, it is worth noting that even with the simpler optimization, our data augmentation method (HDA) can still greatly improve the previous deep learning model Var-CNN and achieve new state-of-the-art results (Table 7 vs. Table 1). This further validates that the proposed augmentation operations are highly compatible with one another and can be applied together well.

5. Conclusion

We presented a model-agnostic, simple yet surprisingly effective data augmentation method, called HDA, for the few-shot website fingerprinting attack. This is an understudied and realistically critical problem, as in practice only a handful of training samples per website can be feasibly collected due to the inherent high dynamics of Internet networks and expensive label collection cost. Importantly, we focus on deep learning-based methods, a line of new research efforts with vast potentials for future investigations. In particular, our HDA method offers three different data augmentation operations, including random rotation, masking, and mixing in intrasample and intersample fashion. They can be applied to the same training samples harmoniously with high complement and compatibility. Moreover, we introduce a sequential augmentation parameter optimization method that captures the interdependence nature between different operations when applied jointly. With recent state-of-the-art deep learning WF attack models, we conducted extensive experiments on four

benchmark datasets to validate the efficacy of our HDA method in both closed-world and open-world scenarios, with and without defense. The results show that the proposed data augmentation method makes dramatic differences in performance and enables previous deep learning methods to outperform hand-crafted feature-based counterparts in the few-shot learning setting for the first time, often by a large margin, while pretrained-based few-shot WF attack (TF) is placed in a new environment, it cannot outperform our augmented method. This is achieved without making any artificial assumptions of relevant, large auxiliary training data for model pre-training. With our HDA method, collecting large training data frequently is eliminated, whilst still achieving stronger and more robust WF attacks. Finally, we performed detailed component analysis to diagnose the effect of individual model components.

5.1. Additional Discussion. Except data augmentation for reducing the demand of data annotation in a few-shot learning context, an alternative approach is semisupervised learning, which has been extensively studied in e-mail classification [49], intrusion detection [50], authorship attribution [51], computer vision [52, 53], and so force. The key idea is to explore the structural knowledge (manifold and cluster structures) of unlabeled data to increase the volume of training data. Crucially, we believe that our proposed HDA can benefit existing semisupervised learning methods due to its algorithm agnostic nature. One limitation with our HDA is that more training data will lead to higher training cost. However, this is a general and common problem with all data augmentation methods including ours. To further boost the research of website fingerprinting, it is necessary to connect website fingerprinting with other fingerprinting fields, from the traditional fingerprint-based biometric systems [54] to the newest collaborative intrusion detection networks under passive message fingerprint attack [55, 56]. Through introducing the strategy which has produced marked effect in related fingerprinting fields, website fingerprinting especially few-shot website fingerprinting would go further.

Data Availability

The principal datasets used in this research can be downloaded from the websites (https://github.com/DistriNet/DLWF and https://www.cse.ust.hk/~taow/wf/data/).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Hunan Province, China (no. 2021JJ40682), the National Key Research and Development Program of China (no. 2018YFB0204301), and National Natural Science Foundation of China (no. 61472439).

References

[1] M. Chen, Y. Wang, Z. Qin, and X. Zhu, "Few-shot website fingerprinting attack," 2021, https://arxiv.org/abs/2101.10063.

[2] D. Roger, N. Mathewson, and S. Paul, "Tor: the second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, pp. 303–320, San Diego, CA, USA, 2004-January.

[3] Tor Developers, "Tor metrics portal," 2018, https://metrics.torproject.org.

[4] A. Hintz, "Fingerprinting websites using traffic analysis," in *Proceedings of the Privacy Enhancing Technologies*, pp. 171–178, San Francisco, CA, USA, April 2003.

[5] 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, October 2006.

[6] D. George Bissias, M. Liberatore, D. Jensen, and B. Levine, "Privacy vulnerabilities in encrypted http streams," in *Privacy Enhancing Technologies*, vol. 3856, pp. 1–11, Springer, Berlin, Germany, 2006.

[7] D. Herrmann, W. Rolf, and H. Federrath, "Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier," in *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science*, pp. 31–42, Chicago, IL, USA, November 2009.

[8] 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, Chicago, IL, USA, October 2011.

[9] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: website fingerprinting attacks and defenses," in *Preceeding of the ACM Conference on Computer and Communications Security*, pp. 605–616, Raleigh, NC, USA, October 2012.

[10] 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, Berlin Germany, November 2013.

[11] T. Wang, X. Cai, I. Johnson, and R. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *Proceedings of the 23rd USENIX Security Symposium*, pp. 143–157, San Diego, CA, USA, August 2014.

[12] J. Hayes and G. Danezis, "K-fingerprinting: a robust scalable website fingerprinting technique," in *Proceddings of the 25th USEUIX Security Symposium*, pp. 1187–1203, Austin, TX, USA, August 2016.

[13] A. Panchenko, F. Lanze, and H. Martin, "Website fingerprinting at internet scale," in *Proceedings of the 16th Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2016.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[16] 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, p. 310, 2019.

[17] 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 ACM Conference on Computer and Communications Security*, pp. 1131–1148, London, UK, November 2019.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA.

[19] K. P. Dyer and S. E. Coull, "Peek-a-boo, i still see you: why efficient traffic analysis countermeasures fail," in *Proceedings of the 33rd Annual IEEE Symposium on Security and Privacy*, pp. 332–346, San Francisco, CA, USA, May 2012.

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

[21] T. Wang and I. Goldberg, "Walkie-talkie: an effective and efcient defense against website fingerprinting," in *Proceeding of the 26th USENIX Security Symposium*, pp. 1375–1390, Vancouver, Canada, August 2017.

[22] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *Proceeding of the European Symposium on Research in Computer Security*, vol. 9878, pp. 27–46, Guildford, UK, September 2016.

[23] K. Al-Naami, A. El-Ghamry, and M. S. Islam, "Bimorphing: a bi-directional bursting defense against website fingerprinting attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 2, pp. 505–517, 2019.

[24] A. Ahmed, R. Jang, A. Khormali, D. Nyang, and A. David, "DFD: adversarial learning-based approach to defend against website fingerprinting," in *Proceeding of the 39th IEEE Conference on Computer Communications, INFOCOM 2020*, pp. 2459–2468, Toronto, ON, Canada, July 2020.

[25] J. Gong and T. Wang, "Zero-delay lightweight defenses against website fingerprinting," in *Proceedings of the 29th USENIX Security Symposium*, pp. 717–734, Boston, MA, USA, August 2020.

[26] T. Wang and I. Goldberg, "On realistically attacking tor with website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 4, pp. 21–36, 2016.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," vol. 60, pp. 1097–1105, 2012.

[28] R. Collobert, J. Weston, and B. Leon, "Natural language processing (almost) from scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[29] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," in *Proceedings of the Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2018.

[30] S. E. Oh, S. Sunkam, and N. Hopper, "p1-fp: extraction, classification, and prediction of website fingerprints with deep learning," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, p. 209, 2019.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proccedings of the International Conference on Learning Representations*, Banff, Canada, April 2014.

[32] A. Van Den Oord, D. Sander, and H. Zen, "Wavenet: a generative model for raw audio," 2016, https://arxiv.org/abs/1609.03499.

[33] Y. Fisher and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2016, https://arxiv.org/abs/1511.07122.

[34] M. S. Rahman, P. Sirinam, N. Mathews, and M. Wright, "Tik-tok: the utility of packet timing in website fingerprinting attacks," *Proceedings on Privacy Enhancing Technologies*, vol. 3, pp. 5–24, 2020.

[35] J. Wei and K. Zou, "Eda: easy data augmentation techniques for boosting performance on text classification tasks," in *Proceedings of the International Joint Conference on Natural Language Processing*, pp. 6381–6387, Hong Kong, China, November 2019.

[36] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: beyond empirical risk minimization," in *Proceedings of the International Conference on Learning Representations*, Vancouver, Canada, May 2018.

[37] F. J. Morenobarea, F. Strazzera, J. M. Jerez, D. Urda, and L. Franco, "Forward noise adjustment scheme for data augmentation," in *Proceedings of the IEEE Symposium Series on Computational Intelligence*, pp. 728–734, Bangalore, India, November 2018.

[38] L. Taylor and G. Nitschke, "Improving deep learning using generic data augmentation," 2017, https://arxiv.org/abs/1708.06020.

[39] S. Connor and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[40] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj, "Convolutional neural networks for patient-specific ecg classification," in *Proceedings of the 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, vol. 2015, p. 2611, Milan, Italy, Aug 2015.

[41] Y. Lecun and Y. Leon Bottou, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, 1998.

[42] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2018, https://arxiv.org/abs/1803.08375.

[43] B. James, R. Bardenet, B. Kégl, and Y. Bengio, "Algorithms for hyper-parameter optimization," in *Proceedings of the NeurIPS'24: Procedding of the 24th Neural Information Processing Systems*, Granada Spain, December 2011.

[44] V. Vapnik, *Statistical Learning Theory*, Wiley-Interscience, Hoboken, NJ, USA, 1998.

[45] C. Szegedy, W. Zaremba, and I. Sutskever, "Intriguing properties of neural networks," 2013, https://arxiv.org/abs/1312.6199.

[46] C. Olivier, J. Weston, B. Leon, and V. Vapnik, "Vicinal risk minimization," *Neural Information Processing Systems*, pp. 416–422, MIT Press, Cambridge, MA, USA, 2000.

[47] F. Chollet, "Keras," 2015, https://keras.io.

[48] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the Computer and Communications Security*, Scottsdale, AZ, USA, November 2014.

[49] Y. Meng, W. Li, and L.-F. Kwok, "Enhancing email classification using data reduction and disagreement-based semi-supervised learning," in *Proceedings of the 2014 IEEE International Conference on Communications (ICC)*, pp. 622–627, June 2014.

[50] W. Li, W. Meng, and M. HoAu, "Enhancing collaborative intrusion detection via disagreement-based semi-supervised learning in iot environments," *Journal of Network and Computer Applications*, vol. 161, Article ID 102631, 2020.

[51] T. Qian, B. Liu, L. Chen, Z. Peng, and M. Zhong, "Tri-training for authorship attribution with limited training data: a

comprehensive study," *Neurocomputing*, vol. 171, pp. 798–806, 2016.

[52] Y. Chen, X. Zhu, and S. Gong, "Semi-supervised deep learning with memory," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 268–283, Munich, Germany, September 2018.

[53] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 3569–3576, New York, NY, USA, February 2020.

[54] L. Janier, M. Gomez-Barrero, L. Chang, A. Pérez-Suárez, and C. Busch, "Fingerprint presentation attack detection based on local features encoding for unknown attacks," *IEEE Access*, vol. 9, pp. 5806–5820, 2021.

[55] W. Li and L. For 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.

[56] W. Li, W. Meng, and H. Horace, "Developing advanced fingerprint attacks on challenge-based collaborative intrusion detection networks," *Cluster Computing*, vol. 21, no. 1, pp. 299–310, 2018.