Unsupervised and Adaptive Tor Website Fingerprinting

Guoqiang Zhang[1], Jiahao Cao[2(✉)], Mingwei Xu[2(✉)], and Xinhao Deng[2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China
zgq17@mails.tsinghua.edu.cn

[2] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China
caojh15@gmail.com, xumw@tsinghua.edu.cn, dxh20@mails.tsinghua.edu.cn

Abstract. Over the past few years, deep-learning based approaches for Tor website fingerprinting have experienced a significant breakthrough in prediction accuracy. However, many of these approaches suppose that their training and testing datasets share similar distributions, i.e. they belong to the same domain. Unfortunately, this assumption is unrealistic since Tor users' distinctive environmental settings have exerted diverse influence on website trace generation. Although several recent methods attempt to address this problem by utilizing transfer learning techniques, they assume that the adversary has some of the trace labels for each website class in the testing dataset, which is typically irrational in real-world scenarios. In this paper, we propose a novel Tor website fingerprinting framework called **U**nsupervised and **A**daptive Tor Website **F**ingerprinting (UAF), which minimizes the distribution discrepancies between the training (denoted as the source domain) and testing (denoted as the target domain) datasets by training a "domain-invariant" feature extractor in an unsupervised manner. UAF employs three trace representations on raw Tor traffic to retain discriminative information for classification and combines multiple source-specific classifiers based on their trace length distributions. The experimental results show that UAF outperforms multiple state-of-the-art Tor website fingerprinting approaches in identifying shifted and unlabeled target domains.

Keywords: Tor · Multi-source domain adaptation · Website fingerprinting · Unsupervised · Transfer learning

1 Introduction

Tor [2] is the leading low-latency anonymity network that shields millions of users worldwide against censorship and surveillance on the Internet. Despite its strong encryption, adversaries can still eavesdrop on encrypted Tor website traffic and infer the website that a Tor user is browsing, resulting in **w**ebsite **f**ingerprinting (WF) attacks. To carry out WF attacks, adversaries visit websites of interest and

H. Duan et al. (Eds.): SecureComm 2023, LNICST 568, pp. 209–229, 2025.
https://doi.org/10.1007/978-3-031-64954-7_11

record the traces to train a classifier. Subsequently, they deploy this classifier to match the traces of a victim Tor user.

Initially, WF researchers manually extract a set of features to represent Tor website traces and process these features with classical machine learning algorithms, aiming to train a Tor website classifier. Although some features, such as packet statistics and traffic burst patterns, are informative and understandable, feature engineering necessitates expert knowledge and imposes too much overhead. Consequently, state-of-the-art WF approaches such as AWF [10] and DF [11] are gradually migrating from feature-engineering to deep-learning (DL), demonstrating remarkable success in terms of classification accuracy.

Nonetheless, Juarez et al. [6] have criticized prior WF methods for their underlying assumption that the distributions of training and testing datasets are similar, which is improbable in real-world scenarios. In other words, while data samples collected under specific user environmental settings are defined as constituting a *domain*, these methods assume that both their training and testing datasets are part of the same domain. However, given the diverse and private nature of Tor users' environmental settings, it is unrealistic for adversaries to generate training datasets that match the exact environmental conditions as the victim Tor users. Thus, the adversary-generated training dataset (denoted as the source domain) and user-targeted testing dataset (denoted as the target domain) should belong to different domains.

To tackle the above problem, recent WF methods such as TF [12] and AF [13] leverage transfer-learning (TL) techniques to extract invariant features that can be shared across source and target domains. However, despite pretraining the feature extractors with source domains, both methods still rely on labeled traces from each website class in the target domain to train their classifiers. This reliance on labeled target traces is irrational as it violates the empirical sense that the adversary cannot obtain the victim users' website trace labels in the real world.

In this paper, we propose a novel Tor website fingerprinting framework, named **U**nsupervised and **A**daptive Tor Website **F**ingerprinting (UAF), to minimize the impact of domain shifts between the labeled source domain and the unlabeled target domain. UAF does not directly transfer models across different domains, which often leads to suboptimal performance. Instead, it aligns trace features between multiple source domains and a single unlabeled target domain, i.e. it leverages multi-source domain adaptation (MDA) to mitigate the distribution discrepancies.

Domain adaptation is a subfield of transfer learning which are used to tackle domain shifts in other research areas. Unlike TL-based methods mentioned above, domain adaptation leverages both source and target domains as inputs and trains the feature extractor by minimizing feature discrepancies between their traces. Therefore, while UAF strives to develop its "domain-invariant" feature extractors, it eliminates the dependence on website trace labels from the target domain. Considering that in many applications the target domain do not match any one available source well, we extend UAF to multiple source domains.

Nonetheless, training "domain-invariant" feature extractors inevitably results in discriminative information loss, which ultimately hampers the classification task's performance. UAF utilizes three effective trace representations and aggregates their predictions to minimize the likelihood of collapse in classification performance due to susceptible data representation. Furthermore, UAF combines source-specific classifiers according to domain relations reflected on trace lengths, one of the top-rank features in WF.

To the best of our knowledge, our research is the first to fingerprint shifted and unlabeled Tor website traces. We constructed a representative dataset that comprises 40 meta-datasets to evaluate UAF. These meta-datasets are generated under diverse environmental settings such as Tor browser bundle (TBB) versions, user locations, and defense methods, thereby belonging to different domains. We conducted comprehensive experiments to evaluate UAF and compared its performance with traditional DL-based WF approaches. Our experimental results demonstrate that UAF has successfully adapted to shifted and unlabeled target domains and outperforms state-of-the-art WF approaches. Specifically, it achieves a performance improvement of up to 43% compared to TF.

The main contributions of this paper can be summarized as follows:

- To address the issue of unrealistic assumptions in Tor website fingerprinting, we present a more practical and challenging adversary model. In this model, target dataset traces are unlabeled and generated under different conditions with the traces of the source datasets.
- We propose a novel Tor WF framework called UAF based on MDA to adapt the shifted and unlabeled target domains. UAF reuses informative handcrafted features of Tor traces that have been deprecated in the deep learning era to improve its performance.
- We built a representative dataset that comprises different domains by generating Tor website traces under various environmental settings. Based on this dataset, we conducted extensive experiments to evaluate UAF and compared its performance with traditional DL-based WF approaches.

2 Threat Model and Related Work

2.1 Threat Model of WF Attacks

Prior researches on WF assume a passive adversary who eavesdrops on the Tor web browsing traffic between the victim user and the entry node of the Tor network, as illustrated in Fig. 1. The adversary does not delay, modify, drop, or decrypt any packets, rendering the attacks difficult to detect.

To initiate a WF attack, the adversary captures Tor website traffic from his own visits to a group of website homepages. The adversary uses these website traces to train a classifier, which is later applied to the website traffic of target Tor users. In a closed-world setting, the victim Tor users only visit a specific set of websites that the adversary utilized to capture traces. On the other hand, in an open-world setting, the victim Tor users can visit websites beyond the adversary's training set.

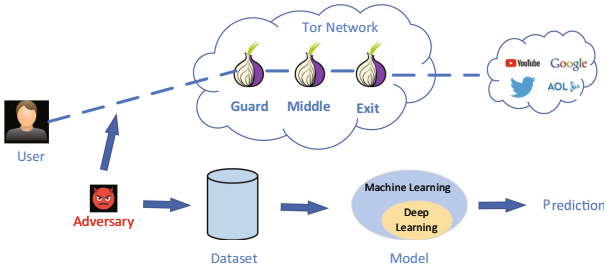


Fig. 1. The threat model for Tor website fingerprinting: a local and passive adversary attempts to infer which website a Tor user is browsing by leveraging machine learning techniques.

2.2 Data Representations in WF Attacks

Early research on WF relied mainly on traditional machine learning classifiers like k-nearest neighbors [14], support vector machine [15], and random forest [5]. The effectiveness of these methods depends heavily on manually selecting discriminative features from Tor website traces. For example, Wang et al. [14] proposed a k-NN attack that extracts a wide combination of features, such as packet ordering, number of bursts, and concentration of outgoing packets. With over 3000 traffic features, they achieved 95% accuracy on 100 websites.

However, feature engineering heavily relies on human intuition, experience, and expert knowledge of the Tor protocol's working mechanism, leading to excessive overheads for researchers. Consequently, researchers have turned to deep-learning (DL) techniques that automate feature engineering and alleviate the burden of feature selection. Rimmer et al. [10] presented three DL models—stacked denoising autoencoder (SDAE), convolutional neural networks (CNN), and long short term memory (LSTM). They demonstrated that their DL-based WF attacks achieved comparable high success rates to state-of-the-art traditional ML-based methods. Similarly, Sirinam et al. [11] proposed Deep Fingerprinting (DF), a CNN-based model implemented with a sophisticated architecture, which demonstrated a promising accuracy of up to 98% and has since been adopted by many WF studies.

As Tor repackages all data packets into constant-sized *cells*, DL-based WF approaches generally represent each Tor website trace by the forwarded cell count and their corresponding directionality. These methods transform each trace into a cell sequence in which $+1$ represents an outgoing cell, -1 indicates an incoming cell, and 0 denotes padding cells which used to align all traces to the same length. Some studies, such as Var-cnn [1] and Tik-tok [8], leverage timing features to represent Tor traces and have also achieved promising results in terms of accuracy.

2.3 Limitations of Traditional WF Methods

The WF evaluation model assumes an adversary that locates between a Tor user and its corresponding entry guard in the Tor network. Since Tor traffic is encrypted, the adversary has no ground truth information on the victim user's website labels. Therefore, adversaries typically use automated browser crawlers to collect traces from websites of interest for both training and evaluation purposes. These Tor website traces are generated under a fixed set of experimental settings, such as browser configuration and network condition. In other words, these WF methods assume their training and testing datasets have similar distributions.

However, Tor traffic patterns heavily depend on the specific environmental conditions in which the traffic is produced. Different environmental conditions can cause significant distribution discrepancies in the Tor traces of the same website. Given the diverse and private nature of users' local environments, it is irrational to assume that the attacker has the same environmental setting as the victim Tor user. Therefore, the training and testing datasets should belong to different domains, resulting in dissimilar distributions.

Juarez et al. [6] claimed that the unrealistic evaluation assumption about user's environmental settings unfairly favors the adversary, leading researchers to significantly overestimate the accuracy of WF attacks. Their experimental results demonstrated that certain factors such as user's browsing habits and TBB versions have significant impacts on the effectiveness of WF attacks. Table 1 lists their evaluation results which show the apparent effect of selecting different TBB versions in the training and testing phases.

Table 1. Prior evaluation results on different TBB versions

TBB	2.4.7 (Test)	3.5 (Test)	3.5.2.1 (Test)
2.4.7 (Train)	**62.7%**	29.9%	12.3%
3.5 (Train)	16.3%	**76.4%**	72.4%
3.5.2.1 (Train)	6.5%	66.8%	**79.6%**

Transfer learning techniques [16] allow for the application of models trained on one task to another. Several recent TL-based WF approaches, such as TF [12] and AF [13], train an independent feature extractor with largescale data from source domains. Then they freeze the feature extractor and train their task classifiers with labeled data from the target domain. The generated vectors are stored and serve as references for their classifiers in subsequent testing using data from the target domain. The workflow of TL-based website fingerprinting is illustrated in Fig. 2.

Nevertheless, these TL-based WF methods mainly concentrate on transferring knowledge learned from a largescale history dataset to a new target dataset, which has only a few samples for each website class. They do not tackle distribution discrepancies between the source and target domains. While directly transferring models to another task often results in suboptimal performance [17], their

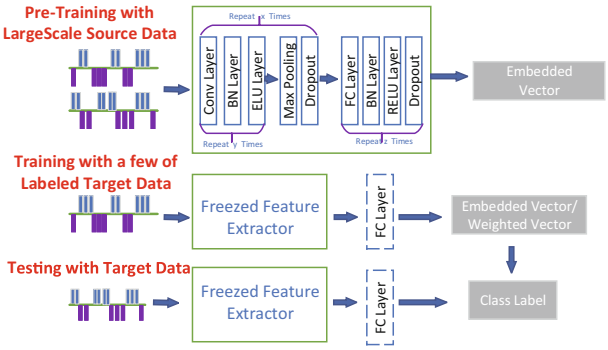


Fig. 2. The workflow of typical transfer-learning based WF approaches.

reliance on labeled target data conflicts with the assumptions that the adversary has no ground truth information on the victim user's website labels in the real world.

3 Unsupervised Adaptive Fingerprinting

3.1 Problem Definition

We use typical MDA settings, assuming each domain consists of a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = (x_1, \cdots, x_m) \in \mathcal{X}$. For a specific domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a classification task $\mathcal{T}$ consists of a feature space $\mathcal{Y}$ and a conditional probability distribution $P(Y|X)$. While each domain is together with its task, we can generally get $P(Y|X)$ with supervised learning from the labeled data (x_i, y_i), where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$.

Assuming that we have labeled training data from K source domains $\mathcal{D}^{S_1}$, $\mathcal{D}^{S_2}, \cdots, \mathcal{D}^{S_K}$ and unlabeled testing data from one target domain $\mathcal{D}^T$. Although all domains share the same feature space ($\mathcal{D}^{S_i} = \mathcal{D}^T$) with the same dimension d, they are generally different in marginal probability distributions. Despite the distribution differences, all classification tasks are identical with the same feature space $\mathcal{Y}$ and conditional probability distribution $P(Y|X)$.

3.2 Model Architecture

In this paper, we aim to learn generic Tor trace features across different domains and classify them accurately. To this end, our proposed WF framework contains two objectives: domain-specific feature alignment and task-specific classification accuracy. However, these two objectives are conflictive as pursuing "domain-invariant" Tor trace features across all domains will inevitably sacrifice discriminative trace characteristics of individual domains, which are critical to task-specific classification. In other words, we must extract features that are both

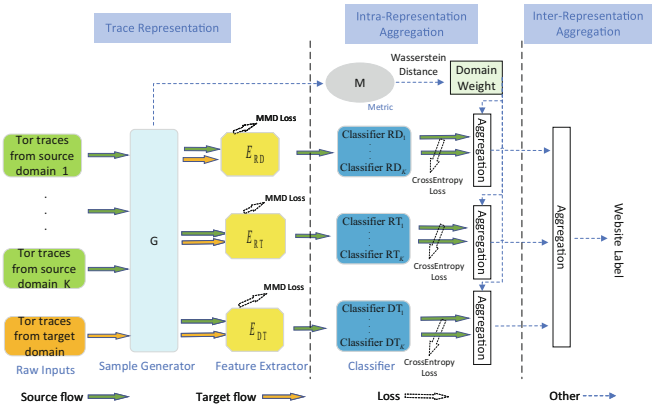


Fig. 3. The architecture of UAF.

discriminative for the learning task and indiscriminate concerning the distribution shifts between the source and target domains.

To address this challenge, we propose three main components in UAF as shown in Fig. 3: (1) *Trace representation* processes raw Tor traces and encodes them with multiple representations. It trains the feature extractors by optimizing the discrepancy loss, aiming to align the latent space of source and target domains. (2) *Intra-representation aggregation* works within each trace representation. To improve the task-specific classification accuracy, it combines the K trained source-specific classifiers by quantifying their relations with the target domain. (3) *Inter-representation aggregation* assembles the outputs of all individual representations and predicts the website labels.

Trace Representation. Typical MDA architectures employ a single global feature extractor that aligns features for both source and target domains. By contrast, MDA methods that simultaneously consider domain-specific alignment and task-specific learning, such as FTD-MSDA [9], MDDA [18], and two-stage MUDA [19], employ an independent domain-specific feature extractor for each source domain. While these approaches indeed improve classification accuracy, they also introduce considerable complexity to their frameworks.

Instead, UAF attempts to retain more discriminative information contained in individual domain traces by enriching the representations of input traces. It utilizes three effective representations—raw directions (RD), raw timestamps (RT), and directional timestamps (DT)—to encode raw Tor traces. The *sample generator (G)* is responsible for transforming the input traces that comprise relative timestamps and directional lengths to normalized samples of these three representations. Before normalizing to the same length, G records the original length of each raw trace in the source and target domains for the *Metric (M)*.

The normalized samples of the source and target domains within each representation are then used to train their corresponding *feature extractor (E)*,

which leverages DF [11] as the basic model. E learns domain-invariant features by reducing domain discrepancy through **m**aximum **m**ean **d**iscrepancy (MMD) metric [3]. MMD is the most widely used metric for unsupervised adaptation that measures the mean statistics between the features of different domains in a projected space, typically the **r**eproducing **k**ernel **H**ilbert **s**pace (RKHS). Specifically, given K batches $\left\{x^{S_1}, ..., x^{S_K}\right\}$ and a batch x^T from source and target domains respectively, E computes the MMD loss $\mathcal{L}_{ali}$ for domain alignment:

$$\mathcal{L}_{ali} = MMD^2\left(x^{S_1} \bigcup ... \bigcup x^{S_K}, x^T\right) \tag{1}$$

where

$$MMD^2\left(\mathcal{D}^S, \mathcal{D}^T\right) = \\ \| \frac{1}{|\mathcal{D}^S|} \sum_{x_s \in \mathcal{D}^S} \phi\left(E\left(x_s\right)\right) - \frac{1}{|\mathcal{D}^T|} \sum_{x_t \in \mathcal{D}^T} \phi\left(E\left(x_t\right)\right) \|^2_{\mathcal{H}} \tag{2}$$

Intra-representation Aggregation. The aligned trace features of each source domain within individual representations will be used to train a specific *classifier (C)* in a supervised manner. Aggregating these source-specific classifiers based on their relations with the target domain is critical for improving task-specific classification accuracy. Several prior approaches have assumed that all source domains are equally important to the target domain [4], overlooking the discrepancy between source and target domains. Alternatively, some methods learn global domain similarity metrics through techniques such as meta-learning [4].

By contrast, UAF endeavors to define similarity measures based on Tor trace length, one of the top-rank features in traditional ML-based WF methods. Our insight is that, while different environmental settings have various impacts on the Tor trace length, all trace lengths of a domain as a whole can serve as an effective relation metric basis. On the other hand, the Wasserstein distance is the commonly used metric to measure domains' similarity, which is theoretically tighter and empirically more stable than other distance metrics in domain adaptation [18]. Therefore, M utilizes the Tor trace lengths provided by G to measure the Wasserstein distances of trace length distributions between each source and the target domain. The distances are normalized with the softmax function and sorted in reverse order. The top N_K classifiers are validated and equally weighted when aggregating predictions within each representation.

Meanwhile, C is trained by reducing the intra-representation loss, a weighted combination of the feature alignment loss $\mathcal{L}_{ali}$ and the classification loss $\mathcal{L}_{cls}$:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \cdot \mathcal{L}_{ali} \tag{3}$$

where λ controls the balance of two losses.

Inter-representation Aggregation. After model training, components within each representation work together to independently predict website labels for

target domain traces. To improve the task-specific classification accuracy, UAF aggregates these predictions and outputs the final website labels. UAF equally utilizes these individual predictions for voting. In case of failure, it simply favors the individual prediction result with the highest probability.

4 Performance Evaluation

4.1 Data Collection

We rented 20 elastic compute service (ECS) hosts from *aliyun.com*, an Internet hosting company and data center operator, to collect Tor traces from the top 100 Alexa websites. The website traces are used to properly evaluate UAF in the closed-world setting. We deployed five relatively new TBB versions across four regions on these hosts, each was provisioned with two CPUs, 8 GB of RAM, and 4Mbps bandwidth. To facilitate data collection, we used *selenium* to automate website browsing and recorded the corresponding traces with *tcpdump*. We visited each website 80 times on each host with the load time and interval set for a single website visit as 200 and 10 s respectively, thus spending 24 days on data collection.

After completing the website crawls, we discarded corrupted traffic traces and processed raw traffic according to the methodology in prior work k-NN [14]. We regularized each trace as a sequence of time deltas and directional packet lengths (negative for incoming and positive for outgoing). The crawled traces were then copied and obfuscated by WTF-PAD [7], a typical WF defense approach, simulating another environmental setting beyond TBB version and user location.

We finally obtained 40 meta-datasets, all consisting of 90 websites with 50 valid traces each. We denote a specific meta-dataset as $D_{i,j,k}$, with the subscripts indicating user location, TBB version, and defense method respectively. Details are listed in Table 2.

Table 2. Subscript symbols of meta-datasets

Meaning of symbols	0	1	2	3	4	*
User Location	Bombay	Frankfurt	Los Angeles	Singapore	N/A	all[a]
TBB Version	11.0.1	11.0.15	11.5.1	11.5.8	12.0a4	all
Defense method	Undefended	WTF-PAD	N/A	N/A	N/A	all

[a]Combining all valid colunms of current row.

4.2 Hyperparameter Tuning

Hyperparameters play a crucial role in model performance. However, an exhaustive search through the hyperparameter space is typically time-consuming, especially when there are tradeoffs between several conflict metrics such as variance,

bias, and classification accuracy. While UAF carefully tuned parameters that are closely related to our datasets, it empirically kept parts of hyperparameters unchanged for its basic model. We acknowledge that someone with more resources might be able to optimize our model further. The search space and the final selected hyperparameter values are shown in Table 3.

Table 3. Hyperparameter tuning of UAF

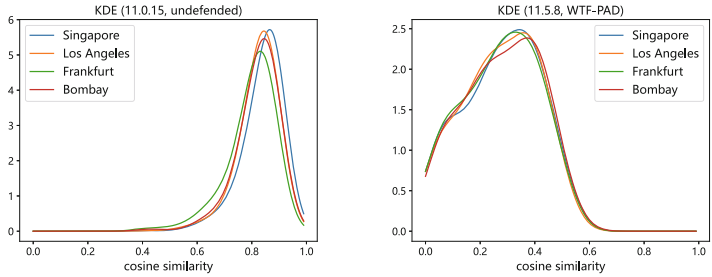
Hyperparameters	Space	Value
training epochs	[20 ... 200]	50
batch size	[16 ... 256]	64
embedding dimension	[64 ... 512]	512
λ	[0.1 ... 2.0]	0.4
distance metric	Wasserstein, Mahalanobis	Wasserstein
optimizer	SGD, ADam	SGD
learning rate	[0.0001 ... 0.1]	0.03
N_K	[2 ... 9]	4

4.3 Domain Drifts in Tor WF

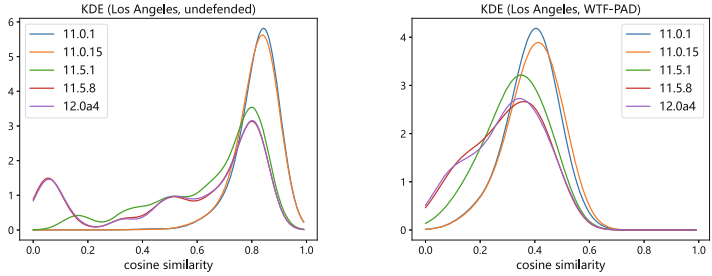
Experiment 1: compare the distributions of traces generated under different conditions.

To validate the significance of our research, we compared the trace distributions across different meta-datasets that were generated under different conditions such as TBB version, user location, and defense method. Firstly, we identified the central trace for each class in one meta-dataset by using cosine similarity as the distance metric and recorded these distance values. Next, we computed cosine similarities between this central trace and each trace in the second meta-dataset. Finally, we plotted **k**ernel **d**ensity **e**stimation (KDE) for values in the two distance sets. We compared trace distributions eight times, each with a unique combination of meta-datasets. Figure 4 illustrates the comparison results.

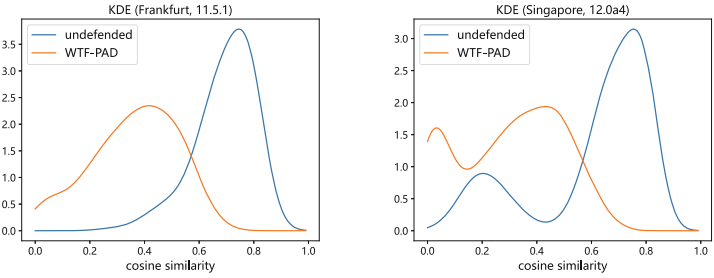
The comparison results show that different factors impact Tor trace distributions to varying degrees. While defense-related changes in Fig. 4c reshape Tor traces to the maximum extent, distribution shifts caused by locational disparities in Fig. 4a are almost negligible. One possible explanation is that our data collection hosts, despite being placed in different locations, share standard physical configurations such as CPUs, memory, and Internet bandwidth to facilitate data collection. Since the impact of location differences is insignificant, we combined meta-datasets of different locations to create ten larger domains in subsequent experiments.



(a) differ in user location



(b) differ in TBB version



(c) differ in defense method

Fig. 4. Compare the distributions of traces generated under different conditions.

On the other hand, upgrading TBB versions leads to asymptotical changes in most cases. However, significant discrepancies can be observed in certain instances due to the introduction of new TBB versions. For example, TBB version *11.5.1* in Fig. 4b demonstrates drastic differences compared to version *11.0.15*.

Experiment 2: evaluate traditional Dl-based WF approaches with shifted domains.

Juarez et al. [6] conducted a WF evaluation using traditional SVM classifiers on different combinations of TBB versions in 2014. Their findings demonstrated the apparent effect of selecting different TBB versions in the training and testing phases. Nonetheless, compared to the ML-based classifiers used in this evaluation, current state-of-the-art WF approaches that benefit from deep learning techniques are more likely to capture invariant trace features, regardless of the TBB versions used.

To find out whether these DL-based methods can perform well in case of domain shifts, we evaluated DF and TF with different combinations of undefended meta-datasets. Tabel 4 shows the attack accuracy of DF and TF when they trained on traces generated by TBB in the row and tested on the traces from the same websites generated by the TBB in the columns. The evaluation results demonstrate that DL-based WF methods remain incompetent to tackle domain shifts caused by different TBB versions.

Table 4. Evaluation results of DF and TF on different TBB versions

TBB	Methods	11.0.1 (Test)	11.5.1 (Test)	12.0a4 (Test)
11.0.1 (Train)	DF	**88.3%**	72.9%	63.7%
	TF	**92.5%**	79.9%	66.1%
11.5.1 (Train)	DF	32.7%	**94.3%**	74.2%
	TF	47.1%	**94.8%**	79.5%
12.0a4 (Train)	DF	32.5%	76.2%	**82.4%**
	TF	42.6%	80.3%	**98.3%**

4.4 UAF Evaluation

Feature Alignment. In this section, we aim to determine an optimal value for the hyperparameter λ and demonstrate the effect of feature alignment through t-SNE visualization.

Experiment 3: determine the hyperparameter λ.

The hyperparameter λ is a weighting factor that balances the cross-entropy loss and the MMD loss. For each value of λ, we selected the target domain in turn and computed the average classification accuracy. The experimental results depicted in Fig. 5 demonstrate that the classification performance is stable when λ ranges between 0 and 1, with the best accuracy obtained at $\lambda = 0.4$.

Experiment 4: t-SNE visualization for non-adapted and adapted features.

In experiment 4, we investigated the impact of MMD on feature distributions through t-SNE visualization. We created source-target domain pairs from undefended meta-datasets and limited the representation to *RD*. To facilitate visualization, we selected no more than two source domains and randomly selected ten website classes from these domains. We obtained non-adapted and adapted

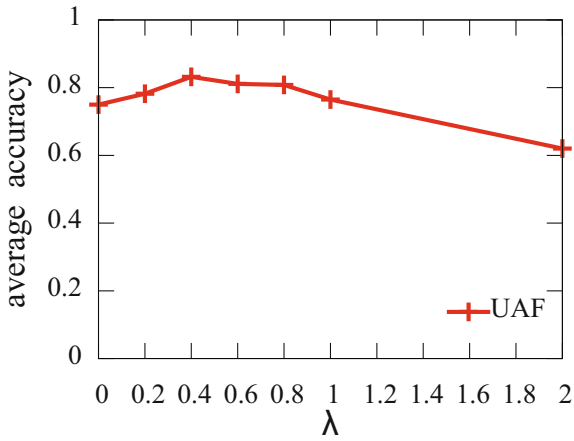


Fig. 5. The average accuracy of UAF changes over hyperparameter λ

features by setting λ to 0 and 0.4 in model training. After the training, we extracted features for source and target domains from the feature extractor. These features were then processed through t-SNE and plotted with different colors for the source (blue-color points) and target (green-color points) domains, as illustrated in Fig. 6.

Our results indicate that the source and target domain features are initially not aligned as shown in the first figure of Fig. 6a and Fig. 6b, but the alignment is improved obviously when MMD is applied in the second figure.

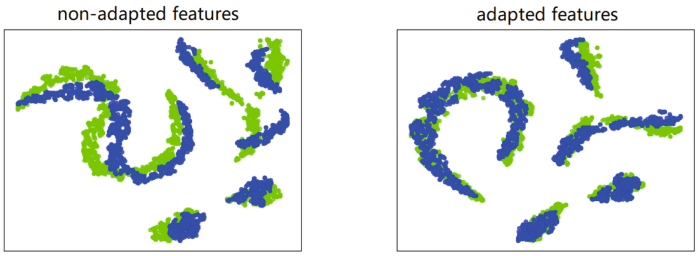
Single-Source Domain Adaptation. MDA can be easily reduced to single-source domain adaptation (SDA). However, SDA-based approaches usually suffer from sub-optimal performance since in many applications the target domain do not match any one available source well. Therefore, we carried out experiment 5 and 6 to evaluate UAF under degraded scenarios.

Experiment 5: evaluate UAF with a single source domain.

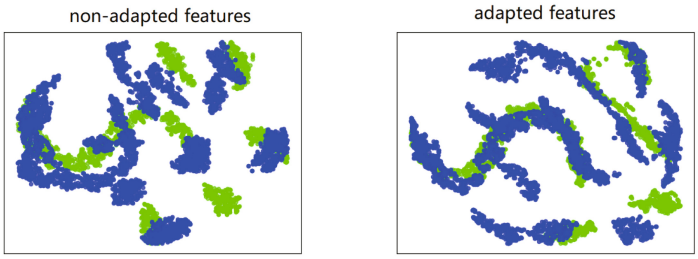
In experiment 5, we evaluated UAF with 4 domains: $D_{*,1,0}$, $D_{*,2,0}$, $D_{*,3,0}$, and $D_{*,4,0}$. Each time we selected one of them as the target domain, with the rest of them as the source domain in turn. The attack accuracy of the single-source UAF is illustrated in Fig. 7. The results reveal that the performance of the single-source UAF fluctuates across a wide range, highly depending on the relations between the source and target domain.

Experiment 6: bidirectional evaluation of the single-source UAF.

In this experiment, we created three domain pairs ($D_{*,2,0}$, $D_{*,2,1}$), ($D_{*,3,0}$, $D_{*,3,1}$), and ($D_{*,4,0}$, $D_{*,4,1}$), each composed of an undefended domain and its padded counterpart. The experiment was conducted under two scenarios. In the first scenario, we trained the single-source UAF with undefended domains and



(a) Training on TBB version 3, testing on TBB version 4



(b) Training on TBB version 2 and 4, testing on TBB version 3

Fig. 6. T-SNE visualization of the non-adapted and adapted features. For the sake of visualization, up to two source domains are used for training. (Color figure online)

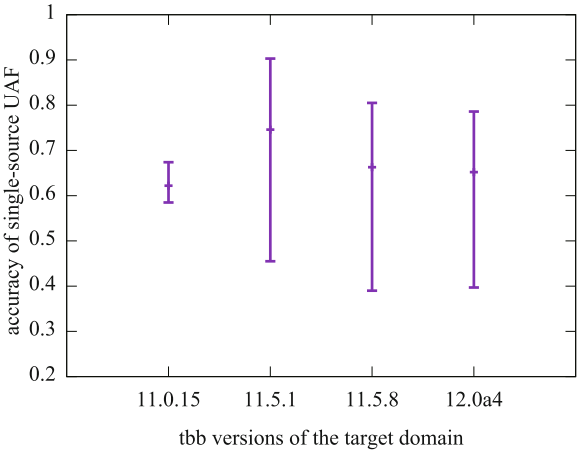
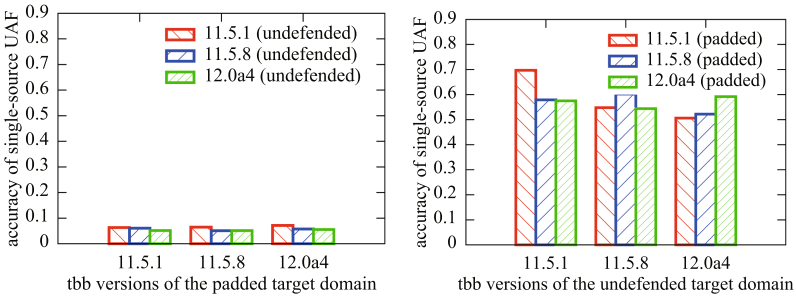


Fig. 7. Evaluation results of the single-source UAF

tested their conterparts. In the second scenario, we swapped the source and target domains. The bidirectional evaluation results are illustrated in Fig. 8.



(a) training on undefended domains, testing on padded domains

(b) training on padded domains, testing on undefended domains

Fig. 8. Bidirectional evaluation results of single-source UAF

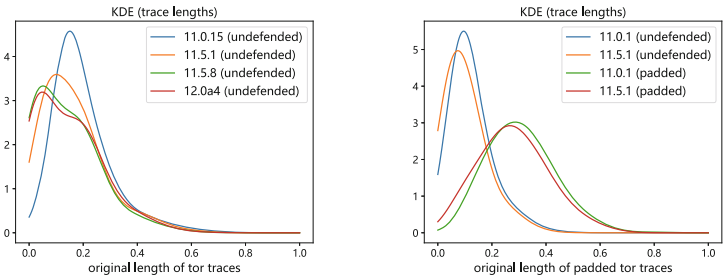
The results indicate that high similarity between the source and target domains is a necessary but not sufficient condition to improve the performance of the single-source UAF. For instance, Fig. 8b shows that testing undefended domains by using UAF trained with padded domains produces modest results. However, despite their acceptable similarity, testing padded domains by using UAF trained with undefended domains, as illustrated in Fig. 8a, produces almost negligible accuracy.

Top N_K Weighting Approach. While most prior MDA approaches either treat all source domains equally or attempt to learn their relations through their marginal distribution similarity, we combine multiple source domains through our top N_K weighting approach. The insight behind this is that different factors such as the TBB version and defense method have clear impacts on Tor trace lengths, thus trace lengths as a whole can serve as an appropriate metric to aggregate domain-specific classifiers. On the other hand, as similarity in length distribution is essential but not adequate, the top N_K weighting approach equally weights the top N_K domain-specific classifiers.

Experiment 7: The distributions of Tor trace lengths.

In this experiment we study the impact of different TBB versions and defense methods on trace lengths. We extracted the length values for Tor traces from both undefended and padded domains. These length values were normalized to 1, and their Kernel Density Estimation(KDE) is plotted in Fig. 9a and Fig. 9b respectively.

The results show that both TBB version and defense method have significant impacts on the lengths of Tor traces. While padding always drastically reshapes the lengths of Tor traces for the sake of security, neighboring TBB versions generally have similar length distributions. Though, in certain cases, newer



(a) trace length distributions of different TBB versions

(b) trace length distributions of different defense methods

Fig. 9. The trace lengths' KDE for different source domains

TBB versions undergo noteworthy updates leading to apparent changes in trace lengths. Presumably due to the intent to improve website loading times, and thereby enhance user experiences, the upgrading of TBB versions is more likely to shorten Tor trace lengths.

Experiment 8: Quantify domain relations based on their trace length distributions.

To quantify the relations between K different domains, we first split them into source and target domains. Next, we computed the Wasserstein distance of trace length distributions for each $(source, target)$ domain pair. We then normalized these distances with the softmax function and extracted the top N_K source domains in reverse order. Finally, we utilized equal weights for the classifiers trained from the chosen top N_K source domains and disregarded other classifiers during the intra-representation aggregation.

Figure 10 illustrates the normalized Wasserstein distances of the top N_K source domains for each target domain. The five TBB versions are denoted as A to E in turn to keep the visualization concise while the corresponding padded versions are attached with mark. The results reveal that the majority of similar source domains for a specific target domain come from its neighboring TBB versions. This finding is consistent with our previous conclusion that neighboring TBB versions generally have similar distributions. Furthermore, the undefended and padded domains of the same TBB version also have closer relations.

Experiment 9: evaluate the top N_K weighting approach.

The experiment 9 attempts to assess the effectiveness of the top N_K weighting approach. While our proposed top N_K weighting approach is denoted as Uni_TN, the plain weighting strategy that treats all source domains equally is denoted as Uni_MS. We denote the best and the average results of UAF under single-source adaptation as $best_SS$ and avg_SS. The evaluation results are listed in Table 5.

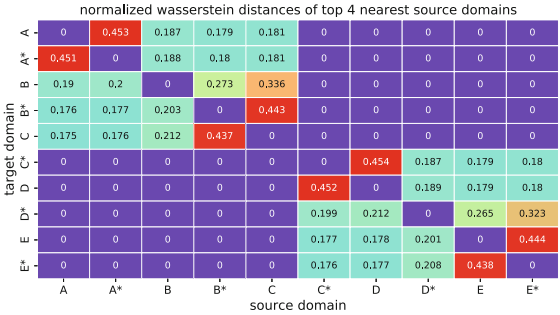


Fig. 10. Heatmap of the top 4 source domains for each target domain. To keep the visualization concise, we denote the 5 TBB versions as A to E in turn. The mark represents padded versions.

Table 5. Evaluate the top N_K weighting approach

Target Domain	best_SS	avg_SS	Uni_MS	Uni_TN
$D_{*,0,0}$	0.676	0.345	0.690	**0.744**
$D_{*,0,1}$	0.466	0.177	**0.489**	0.475
$D_{*,1,0}$	0.674	0.372	0.687	**0.727**
$D_{*,1,1}$	0.477	0.128	0.460	**0.497**
$D_{*,2,0}$	**0.903**	0.535	0.873	0.886
$D_{*,2,1}$	0.672	0.298	**0.758**	0.743
$D_{*,3,0}$	0.805	0.536	0.824	**0.867**
$D_{*,3,1}$	0.686	0.278	0.749	**0.762**
$D_{*,4,0}$	0.786	0.543	0.843	**0.867**
$D_{*,4,1}$	0.674	0.272	0.737	**0.760**

As shown in Table 5, UAF outperforms other baselines in most cases. It should be noted that *best_SS* sometimes outperforms UAF when a specific source domain has high similarities with the target domain, which is consistent with our observations regarding neighboring TBB versions in previous experiments. Nevertheless, as we cannot obtain sufficient information on the target domain, it is impossible to select the best individual source domain in advance.

Multiple Trace Representations. *Experiment 10: aggregate predictions of multiple trace representations.*

UAF aggregates multiple trace representations to retain discriminative information of Tor traces during feature alignment. In this experiment, we examine the effectiveness of representation aggregation compared to individual representations. We follow the basic experimental settings as experiment 9, but with UAF

degraded to one single trace representation. The evaluation results are listed in Table 6.

Table 6. Compare representation aggregation with individual representations

Target Domain	RD	DT	RT	Aggregation
$D_{*,0,0}$	0.725	0.732	0.687	**0.744**
$D_{*,0,1}$	0.451	0.443	0.402	**0.475**
$D_{*,2,0}$	0.848	0.844	0.773	**0.886**
$D_{*,2,1}$	0.726	0.715	0.646	**0.743**
$D_{*,4,0}$	0.853	0.844	0.792	**0.867**
$D_{*,4,1}$	0.726	0.735	0.646	**0.760**

The results indicate that, while *RD* and *DT* are comparable in terms of attack accuracy, *RT* performs worst. By aggregating the predictions from all three representations, UAF demonstrates better attack accuracy than individual representations.

4.5 Compare with Traditional DL-Based WF Approaches

Experiment 11: evaluate DL-based WF baselines.

Deep Fingerprinting (DF) and Triplet Fingerprinting (TF) are two state-of-the-art Tor website fingerprinting approaches. While DF leverages convolutional neural networks with a sophisticated architecture design to extract features and classify Tor traces, TF implements a triplet network that trains its independent feature extractor and predicts website labels with a k-NN classifier. In experiment 11, we compare the attack performance of UAF with these two traditional DL-based WF methods, which were denoted as *mix-DF* and *mix-TF* since they both directly mix all source domains as their training datasets.

Considering that both UAF and TF use DF as their basic model, we evaluate DF with the same parameters as UAF and TF. For TF, M represents the number of traces per class used in pre-training, N represents the number of traces per class used in target training data, and T represents the number of traces per class in the target testing data. Following the approach in TF [12], we randomly select the same number of Tor traces for each website class from each meta-dataset. While the pre-training dataset consists of $M = 80$ traces per class, the target dataset consists of $T = 200$ traces in each class for testing.

We train the k-NN classifier using a number of N traces from the pre-training dataset, where $N = \{5, 10, 20\}$. The results are showed in Table 7.

The results demonstrate that UAF is the most effective method in cross-domain website fingerprinting. Specifically, it outperforms TF by up to 43%. We believe the potential reason is that TF cannot extract effective features from multiple domains with several shots of samples per class.

Table 7. Attack performance of UAF and traditional DL-based approaches

Target Domain	Traditional DL-based Methods				DA-based Methods
	Mix-DF	Mix-TF			UAF
		N = 5	N = 10	N = 20	
$D_{*,2,0}$	0.858	0.595	0.612	0.614	**0.886**
$D_{*,2,1}$	0.714	0.487	0.496	0.501	**0.743**
$D_{*,3,0}$	0.832	0.623	0.618	0.634	**0.867**
$D_{*,3,1}$	0.736	0.495	0.502	0.517	**0.762**
$D_{*,4,0}$	0.824	0.605	0.637	0.629	**0.867**
$D_{*,4,1}$	0.739	0.535	0.541	0.532	**0.760**
Average	0.784	0.557	0.568	0.571	**0.814**

5 Discussion

In this study, we found that changes in environmental settings, or domain shifts, have significant impacts on the generation of Tor traces. In addition, the inability to obtain labels for target traces in the real world exacerbates the training conditions of website fingerprinting. We conducted experiments with state-of-the-art DL-based WF approaches and found that they are incompetent at coping with such practical scenarios. Therefore, prior WF methods may not be realistic when applied to the actual Tor network.

We employed weighted multi-source domain adaptation, a typical model that tackles domain shifts in other fields, to address this issue. Our new approach, named UAF, takes advantage of Tor trace characteristics verified in the WF research domain to optimize the performance of both feature-specific alignment and task-specific classification. The experiment results show that UAF effectively improves the attack accuracy of website fingerprinting against unlabeled and dissimilar target domains.

However, we have not conducted systematic research on the factors that affect Tor trace generation. Our datasets only accounted for several limited factors, such as TBB version and defense method. Furthermore, we conducted experiments under a closed-world setting using a restricted number of reference methodologies. Despite these limitations, our study suggests that domain adaptation is a feasible direction to tackle domain shifts in WF. Rather than a one-size-fits-all approach, further investigation is needed to determine the best strategies for specific WF tasks.

6 Conclusion

Prior WF attacks have been criticized for producing optimistic evaluation results based on unrealistic assumptions. In particular, these attacks assume that the training and testing traces have similar distributions. Therefore, they collect

these traces under the same environmental settings, forming a single domain for all traces. By contrast, the real Tor website traces that an adversary would observe in practice should belong to multiple domains since they are generated under significantly more complex and diverse environmental settings.

In this paper, we attempt to fingerprint Tor websites in a target domain that is both unlabeled and dissimilar from all source domains. Correspondingly, we present a new framework based on weighted multi-source domain adaption that improves the robustness of Tor website fingerprinting in such practical scenarios. Our new method, namely UAF, strives to improve both feature-level and task-level performances by utilizing Tor trace characteristics. The results of our extensive experiments show that UAF effectively tackles unsupervised domain adaptation and consistently outperforms traditional state-of-the-art DL-based WF approaches.

Acknowledgments. We are grateful to all the anonymous reviewers for their insightful comments. The research is supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62202260 and 62221003.

References

1. Bhat, S., Lu, D., Kwon, A., Devadas, S.: Var-CNN: a data-efficient website fingerprinting attack based on deep learning. Proc. Priv. Enhancing Technol. **2019**(4), 292–310 (2019)
2. Dingledine, R., Mathewson, N., Syverson, P.: Tor: the second-generation onion router. Technical report, Naval Research Lab, Washington DC (2004)
3. Gretton, A., Borgwardt, K.M., Rasch, M.J., Schölkopf, B., Smola, A.: A kernel two-sample test. J. Mach. Learn. Res. **13**(1), 723–773 (2012)
4. Guo, J., Shah, D.J., Barzilay, R.: Multi-source domain adaptation with mixture of experts. arXiv preprint arXiv:1809.02256 (2018)
5. Hayes, J., Danezis, G.: k-fingerprinting: a robust scalable website fingerprinting technique. In: 25th USENIX Security Symposium (USENIX Security 2016), pp. 1187–1203 (2016)
6. Juarez, M., Afroz, S., Acar, G., Diaz, C., Greenstadt, R.: A critical evaluation of website fingerprinting attacks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, pp. 263–274 (2014)
7. Juarez, M., Imani, M., Perry, M., Diaz, C., Wright, M.: Toward an efficient website fingerprinting defense. In: Askoxylakis, I., Ioannidis, S., Katsikas, S., Meadows, C. (eds.) ESORICS 2016, Part I. LNCS, vol. 9878, pp. 27–46. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-45744-4_2
8. Rahman, M.S., Sirinam, P., Mathews, N., Gangadhara, K.G., Wright, M.: Tik-Tok: the utility of packet timing in website fingerprinting attacks. arXiv preprint arXiv:1902.06421 (2019)
9. Rezaeianjouybari, B., Shang, Y.: A novel deep multi-source domain adaptation framework for bearing fault diagnosis based on feature-level and task-specific distribution alignment. Measurement **178**, 109359 (2021)
10. Rimmer, V., Preuveneers, D., Juarez, M., Van Goethem, T., Joosen, W.: Automated website fingerprinting through deep learning. arXiv preprint arXiv:1708.06376 (2017)

11. Sirinam, P., Imani, M., Juarez, M., Wright, M.: 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 (2018)
12. Sirinam, P., Mathews, N., Rahman, M.S., Wright, M.: 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 (2019)
13. Wang, C., Dani, J., Li, X., Jia, X., Wang, B.: Adaptive fingerprinting: website fingerprinting over few encrypted traffic. In: Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, pp. 149–160 (2021)
14. Wang, T., Cai, X., Nithyanand, R., Johnson, R., Goldberg, I.: Effective attacks and provable defenses for website fingerprinting. In: 23rd USENIX Security Symposium (USENIX Security 2014), pp. 143–157 (2014)
15. Wang, T., Goldberg, I.: Improved website fingerprinting on Tor. In: Proceedings of the 12th ACM Workshop on Privacy in the Electronic Society, pp. 201–212 (2013)
16. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, vol. 27 (2014)
17. Zhao, S., Li, B., Xu, P., Keutzer, K.: Multi-source domain adaptation in the deep learning era: a systematic survey. arXiv preprint arXiv:2002.12169 (2020)
18. Zhao, S., et al.: Multi-source distilling domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 12975–12983 (2020)
19. Zhu, Y., Zhuang, F., Wang, D.: Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5989–5996 (2019)