Stop, Don't Click Here Anymore: Boosting Website Fingerprinting By Considering Sets of Subpages

Asya Mitseva
Brandenburg University of Technology
(BTU Cottbus, Germany)

Andriy Panchenko
Brandenburg University of Technology
(BTU Cottbus, Germany)

Abstract

A type of traffic analysis, *website fingerprinting (WFP)*, aims to reveal the website a user visits over an encrypted and anonymized connection by observing and analyzing data flow patterns. Its efficiency against anonymization networks such as Tor has been widely studied, resulting in methods that have steadily increased in both complexity and power. While modern WFP attacks have proven to be highly accurate in laboratory settings, their real-world feasibility is highly debated. These attacks also exclude valuable information by ignoring typical user browsing behavior: users often visit multiple pages of a single website sequentially, e.g., by following links.

In this paper, we aim to provide a more realistic assessment of the degree to which Tor users are exposed to WFP. We propose both a novel WFP attack and efficient strategies for adapting existing methods to account for sequential visits of pages within a website. While existing WFP attacks fail to detect almost any website in real-world settings, our novel methods achieve F1-scores of 1.0 for more than half of the target websites. Our attacks remain robust against state-of-the-art WFP defenses, achieving 2.5 to 5 times the accuracy of prior work, and in some cases even rendering the defenses useless. Our methods enable to estimate and to communicate to the user the risk of successive page visits within a website (even in the presence of noise pages) to stop before the WFP attack reaches a critical level of confidence.

1 Introduction

Over the past two decades, more and more of users' daily activities have moved online. As a result, online privacy has become increasingly important. While encryption helps protect the confidentiality of user data exchanged over the Internet, it does not hide the identities (i.e., IP addresses) of the communicating parties and the relationship between them. The latter is often abused by corporations to profile and track Internet users, by repressive governments to engage in nationwide censorship, and even to track and persecute individuals exercising their fundamental right to freedom of speech and expression. In response, several methods for anonymous communication have been proposed [31,47]. Today, Tor [15] is the most popular low-latency anonymization network. To provide anonymity, it uses the principle of onion routing, where users build a virtual tunnel via a chain of three volunteer nodes, called onion relays (ORs), and send their data encrypted in multiple layers. This design ensures that no OR in the path knows both the user and the destination at the same time.

Despite Tor's promise to hide the relationship between users and their destinations from a *local passive observer* (located on the link between the Tor user and the first anonymization node, i.e., the entry node), e.g., an Internet service provider—one of the weakest attackers in the Tor's adversarial model [15], Tor leaks information about the number, direction, and timing of packets transmitted. This can be exploited for sophisticated attacks such as website fingerprinting (WFP) [24,37,39,45]. WFP is a type of traffic analysis that aims to reveal the website a user is visiting by observing and analyzing data flow patterns. Over the years, a large number of studies [24,38,41,43,45,48] has systematically showed the continuous improvement in the effectiveness of WFP attacks. Today, modern WFP methods achieve a classification accuracy of more than 90% in laboratory settings. However, their real-world efficiency is highly controversial because they rely on unrealistic assumptions and have unknown scalability due to the huge universe size of the World Wide Web (WWW).

The main limitation of existing WFP attacks is their focus on detecting *individual pages* (typically index pages) via *isolated* page visits, rather than the websites that an accessed page belongs to. First, these attacks omit valuable information by ignoring typical user browsing behavior, where users usually visit multiple pages of a single website in sequence [11], e.g., by following links. Second, fingerprinting only individual pages within the entire WWW universe results in attacks with severely limited scalability in practice as websites are comprised of numerous pages, making the size of the universe even larger. Although previous research [24, 38, 41, 43, 45] assumes silently that detecting the index page of a website is

sufficient to recognize this website as users always access the website's content through it, the validity of this assumption often does not align with the current user browsing habits. For instance, users frequently access various websites' content through browser-cached URLs [11] or URLs received via email or instant messaging, bypassing the index pages of those websites. Even if the attacker attempts to enumerate all individual pages of a given website, this is often practically impossible due to their vast number. However, if the attacker can exploit the additional information leaked by the set of individual pages belonging to the same website and visited by a user in sequence, WFP will become significantly more accurate and scalable, and thus more dangerous than assumed.

In this paper, we provide a more realistic assessment of the degree to which Tor users are exposed to WFP and propose both a novel WFP attack and efficient strategies for adapting existing classifiers to account for sequential visits to pages of a single website. Our novel methods do not require knowing the order of pages visited, making WFP even more realistic. We show that two, at most three, clicks within a website allow the attacker to increase the accuracy by more than 20% in a closed-world scenario, bringing it into the alarming range. Using our collected dataset, we investigate the limitations of our methods in real-world settings and show that our attacks achieve F1-scores of 1.0 for more than half of the websites of interest, while existing techniques fail to detect almost any website. Our attacks remain robust against state-of-the-art WFP defenses, achieving up to five times the accuracy of prior work, and in some cases even rendering the defenses useless. Overall, our analysis shows that detecting websites based on sequential page visits, rather than individual pages, poses a more serious threat to Tor users in the real world, and appeals for research to rethink the existing assumptions in the attacker model and to reconsider existing evaluations and WFP defenses. The contribution of this paper is as follows:

1. We perform the first systematic analysis of the suitability of existing classifiers for *website* fingerprinting (i.e., detecting websites by analyzing patterns from their individual pages). We evaluate several strategies and conclude that naive methods of adapting webpage classifiers for website fingerprinting are insufficient. Specifically, we observe a 20 to 30% decrease in accuracy for *website* fingerprinting compared to webpage fingerprinting.
2. We design a novel dedicated set-aware method for website fingerprinting based on multiple instance learning, which can naturally handle the set of pages visited by a user sequentially. We further propose several efficient voting-based strategies to adapt existing WFP classifiers to take into account the set of pages of a website. We analyze the effectiveness of both our strategies and our set-aware WFP attack in an open-world scenario and show that WFP poses a serious threat to users who click on multiple links within the same website.
3. We evaluate the effectiveness of state-of-the-art defenses in preventing our novel WFP methods. We show that WFP defenses are up to five times less effective than expected, or even useless in some cases. Thus, considering sets of pages is a powerful tool to successfully enhance the WFP attack even in the context of newer defenses.

Finally, we discuss the limitations of classifiers, the weaknesses of existing defenses, and the need for further research.

2 Threat Model

Unlike most previous research [4, 43, 45, 46], which has concentrated on fingerprinting individual pages (mainly the index pages of websites), we examine an attacker whose goal is to identify the *website* visited by a user, regardless of the specific page accessed on that website. We strictly distinguish between a *website* and a *page*. A website comprises pages, hosted under the same domain and identifiable by a unique URL. We focus on the worst-case scenario from user's perspective, in which users browse a website in a single tab, visiting multiple pages in sequence by following links. Crichton et al. [11] have recently shown that specific user groups frequently exhibit this browsing behavior in their daily routine. Thus, we believe that some Tor users also choose to use this strategy, in part for better performance. Nonetheless, we examine the impact of including pages in the set of visited pages that do not belong to the target website and discuss the effects of multi-tab browsing on our WFP methods in Section 6.4.

For our analysis, we assume that the attacker specifies a set of targeted websites and retrieves a certain number of pages from them. For each page, multiple traffic traces are collected to extract potentially significant patterns, i.e., *features*, needed to train a supervised machine learning (ML) method and to create a model. The model is then used to detect an accessed website, which corresponds to an unknown trace of a real user. WFP is examined through two threat models: *closed* and *open* world. In a closed-world scenario, the user is restricted to access a limited set of websites, and the attacker has patterns for these websites. In an open-world scenario, the attacker has only traces for *foreground* websites of interest, while the user can visit an unlimited *background* set of websites.

Finally, we assume the attacker is a passive observer who does not decrypt, modify, or interrupt transmitted packets. He is positioned either between the user and the entry node or at the entry node itself, and monitors traffic exchanged with a Tor user and is aware of its IP address. Similarly to other works [24, 38, 41, 43, 45], the attacker can identify the start and end of each page load [50].

3 Related Work

Unlike our work, most previous research has focused on developing attacks **fingerprinting individual pages**.

Traditional ML-based Attacks. Herrmann et al. [25] studied the first WFP attack on Tor utilizing packet size distributions

to train a Multinomial Naïve Bayes classifier. Their results showed a detection rate of less than 3%. Panchenko et al. [40] enhanced the attack accuracy to nearly 55% by using Support Vector Machines (SVM) and multiple feature sets based on packet volume, time, and direction. They also presented the first open-world analysis of WFP. Dyer et al. [17] proposed a variable n-gram classifier and analyzed new feature sets based on time and bandwidth. However, their accuracy did not exceed [40]. Cai et al. [7] used a SVM classifier that employed Damerau-Levenshtein edit distance between fingerprints and increased the accuracy to over 80%. This attack was further improved by removing Tor management cells from traffic traces and using a new set of distance-based metrics to distinguish between fingerprints [49]. Despite achieving a substantial accuracy improvement, these methods were rendered impractical due to their high computational costs.

In response, a new generation of attacks was developed. The k-Nearest Neighbor (k-NN) classifier [48] relies on a large and diverse feature set and attained an accuracy of 91% in a closed world and an 85% detection rate when using over 5,000 background pages in an open world. The CUMUL classifier [39] relies on a feature set comprising the cumulative sizes of transmitted packets for a given page load and SVM. By sampling cumulative packet sequence, it implicitly considers the page load pattern in its feature space. The k-FP method [24] uses a random decision forest classifier to produce feature vectors and a k-NN classifier for classification. Like CUMUL, k-FP achieved an accuracy of more than 90% in a closed world. Gálvez et al. [19] studied the use of clustering in the WFP domain. Unlike our work, where we employ clustering to partition multiple pages within a website, Gálvez et al. apply clustering to detect index pages of websites.

DL-based Attacks. More recent studies have examined the use of deep learning (DL) to enable automated selection and fine-tuning of features. The first DL-based WFP attack [2] was based on a stacked denoising autoencoder and achieved 88% accuracy in a closed world. Rimmer et al. [42] analyzed three different DL methods enhancing the accuracy of the previous attack to 96%. They showed that features generated automatically by certain DL methods are more resilient to constantly evolving web content. Still, these methods required a much larger number of training samples than traditional ML classifiers. Sirinam et al. [45] first tackled this problem with deep fingerprinting (DF). DF is based on a convolutional neural network (CNN) with activation functions specifically adapted to WFP and a number of precautions against overfitting. Var-CNN [4] is built using a Residual Neural Network with automated feature extraction and a set of handcrafted features. It achieves higher accuracy than DF by including direction and timing data from packets, especially when working with limited training data. Tik-Tok [41] represents another DL-based attack that uses both packet directions and timing of bursts in raw data. Recently, Shen et al. [43] proposed robust fingerprinting (RF) comprising a traffic representation method for capturing fine-grained features and a CNN-based classifier. RF has shown an improved capability to overcome defenses. Oh et al. [38] employed unsupervised DL methods solely for automated feature extraction. They showed that traditional ML-based classifiers (e.g., k-NN, CUMUL, k-FP) improved in accuracy when fed with features derived automatically.

Other studies have tackled the challenge of applying WFP with limited data. Triplet fingerprinting [46] is based on N-shot learning and requires only a few traffic traces per page to train a model. However, building a model for the feature extractor using large training sets remains a necessary step in extracting features from the raw data in its pre-training phase. GANDaLF [37] employs a semi-supervised attack through a generative adversarial network (GAN), which produces a large set of synthetic traces for training a deep neural network. It requires only a small labeled dataset, but needs a more extensive second dataset for GAN training.

Fingerprinting Websites. A limited number of studies have focused on detecting websites. Cai et al. [7] attempted to fingerprint websites by analyzing multiple pages within the same domain and the impact of clicking on embedded links. They used a Hidden Markov Model (HMM) to simulate a typical user behavior and obtained high accuracy by examining only two websites. Zhuo et al. [54] used a Profile HMM—an extended HMM capable of, e.g., zero transitions—to model websites. However, the assumption of having prior knowledge of the order of pages accessed by the user questions the scalability of these attacks in practice.

Panchenko et al. [39] analyzed the effectiveness of CUMUL using a small dataset when the attacker aims to detect a visited website regardless of the specific page accessed by the user. Oh et al. [37] performed similar evaluation on GANDaLF, where the attack achieved an accuracy of up to 62% in a closed world. In this work, we continue the research of [37, 39] and perform the first systematic analysis of webpage classifiers for detecting websites. We identify multiple strategies how to adapt these methods to achieve effective website fingerprinting attacks. Our work is the first to examine the risks of WFP when users sequentially browse multiple pages of a single website. Once a website has been detected by our methods, existing works such as [36, 44] can be used to pinpoint the exact pages that were visited by the user.

Criticism on Existing WFP Attacks. WFP attacks have been criticized for making impractical assumptions. Juarez et al. [29] argued that prerequisites such as non-changing web content and use of same vantage point, made by previous work, do not necessarily hold in real world. Panchenko et al. [39] demonstrated that current methods for fingerprinting individual pages do not scale in realistic settings, i.e., for every page there are at least several others that look similar to the classifier and cause confusion. Cherubin et al. [10] analyzed the potential of WFP attacks in the operational Tor network. By observing real user traffic, they achieved a high detection rate for a limited number of pages, but observed

a significant drop in accuracy as the number of monitored pages increased. Our work tackles the scalability issue of existing attacks through fingerprinting websites, instead of individual pages, and using additional information obtained by sequential visits to multiple pages belonging to a single website.

Multi-tab Website Fingerprinting. Some research has designed WFP attacks for multi-tab setting. A few studies [21, 51, 52] have focused mainly on retrieving and analyzing only those portions of transmitted packets that correspond to single page traffic. Other works [12, 22] have used majority voting and attention scoring to differentiate traffic chunks of concurrently loaded pages and to classify them. In contrast, we use different voting strategies for multiple pages to detect the website visited. Recently, Deng et al. [14] took a different approach to tackle the usage of multiple tabs. They considered it as a multi-label classification problem and proposed a transformer model that uses several short traffic patterns extracted from each page loaded in a tab. While these attacks show feasibility of WFP even when using multiple tabs, our novel methods are able to further boost these attacks and overcome their limitations. Further implications of multi-tab browsing on our WFP methods are discussed in Section 6.4.

4 Our Novel Fingerprinting Methods

The novelty of our WFP attacks is their ability to incorporate previously ignored leakage obtained from a user's sequential browsing activity across multiple pages within a website. Thus, our methods can more accurately detect the visited website, regardless of the individual pages accessed, resulting in increased scalability in the real world. We first suggest strategies to empower existing webpage classifiers to deal with websites and, then, introduce our novel set-aware method.

4.1 Our Novel Voting-based Strategies

Consecutive page visits can expose valuable information, which can be combined with individual page predictions made by webpage classifiers, to identify websites. We examine six different voting-based strategies that incorporate predictions for individual pages within a given website, without even the need to consider their visiting order. For all of our strategies, we assume the usage of a webpage classifier that is trained on multiple website classes and can compute a set of probability values for each testing page, determining the likelihood of the given page to originate from each of these websites.

Our most basic strategy, *majority voting*, counts the number of pages predicted to be associated with each website class and selects the most frequently predicted website class as the final choice. If two or more website classes have equal numbers of predictions, we randomly select one of them as the final prediction. In *probability voting*, we multiply the probabilities assigned by the classifier to each page in the testing set. For each website class, we compute a probability for each testing set of pages and the website class with the highest probability is then chosen as the final prediction. The *mean voting* is similar to probability voting, except that it entails adding the probabilities of the testing pages instead of multiplying them.

Contrary to the methods above, we propose three additional strategies that account for varying levels of confidence in separate predictions. The *weighting methods* employed in these strategies ensure that predictions of individual pages within a testing set, which stem from a decision made with high confidence, carry more weight in the final prediction than predictions based on less certain classifier's decisions. We use three metrics to weight each probability value when computing the average for our mean voting strategy: (i) variance, (ii) standard deviation, and (iii) Gini coefficient [9, 34]. We refer to the corresponding voting-based strategies as *variance-weighted mean voting*, *standard deviation-weighted mean voting*, and *Gini-weighted mean voting*, respectively. The metrics are computed over the list of probability values assigned to a given page indicating the likelihood that the page belongs to each of the target websites. High values for one or more of these metrics indicate significant discrepancy between probability values for different website classes of a given page and point out strong confidence in the classifier's decision.

4.2 Multiple Instance Learning for WFP

Our voting-based strategies strongly depend on the efficacy of the underlying webpage classifiers that were not specifically tailored to fingerprint websites. Alternatively, we propose to apply a special branch of supervised learning, called *multiple instance learning (MIL)* [8]. In MIL, a group of instances is referred to as a *bag* and is treated as a single instance similar to the classical instance-based methods. The main goal is to extract features that capture essential patterns of the entire set and to train a classifier capable of predicting a single label for each bag, rather than separate labels for the individual instances. In our work, we propose a MIL-based WFP attack that employs an architecture inspired by a state-of-the-art attention-based MIL approach [27]. Our MIL-based attack utilizes weights to identify particular pages in each bag, which can improve the confidence of the classifier's decision and, thus, increase the accuracy of associating a given set of pages with a given website. The novelty of our method is its flexible, multi-layered neural network that computes the weighted average from the traffic traces of all pages in a bag.

Figure 1 illustrates the architecture of our method. Initially, the method is provided with several website classes containing a subset of pages. We adhere to the data representation used in previous work [42, 45, 46], where each traffic trace comprises a sequence of $+1$ and -1, indicating outgoing and incoming packets, respectively, and excludes the size and timestamps of those packets. Similar to [4, 45, 46], the traffic traces are

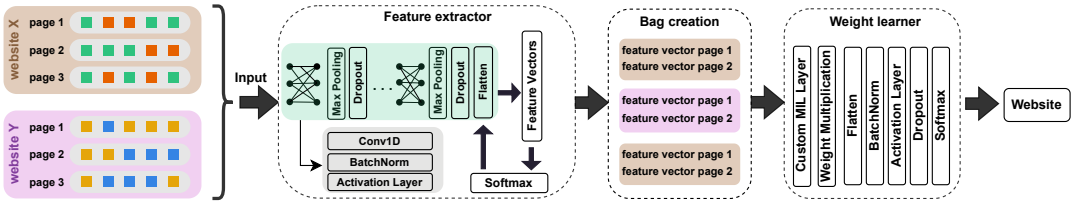


Figure 1: A general overview of the architecture of our MIL-based WFP attack.

either truncated or padded with zeroes to conform to a fixed packet length of 5,000. The traffic traces are then inputted into the *feature generator*, which closely corresponds to the CNN model developed in DF [45]. The feature generator includes eight convolutional layers, followed by batch normalization and activation functions to prevent overfitting. It also includes four max-pooling layers for input data reduction and four dropout layers for overfitting mitigation during the reduction. Instead of using a fixed learning rate drop-off scheme with a fixed number of epochs, as done in [45], we employ a learning rate that varies and adjusts during training based on the performance of the validation set. Finally, we replace the fully connected layers and output prediction previously developed in DF with a single softmax layer, permitting intermediate evaluation of the data in reduced dimension.

Once the dimensions of the initial traffic traces have been reduced, our method proceeds to *bag creation*. During this stage, the traces are grouped into multiple bags and each bag is labeled with the corresponding website class to which the pages belong. We ensure that each bag contains traces of different pages. If the initial website class has multiple traces from a single page, they are distributed to multiple bags. The resulting bags are fed into the final stage of our method, called *weights learner*. To develop the weights learner, we first introduce a custom layer that computes and optimizes the weights derived from traces in a single bag. Similar to [27], this layer implements the hyperbolic tangent function to consider incoming and outgoing packets in separate traces. The activation function softmax is utilized to ensure all page weights in a bag sum to one, avoiding the influence of different bag sizes. The computed weights are then employed in the following layer for the multiplication with the initial traces in a bag and the output is flattened. Before making predictions for the final website label, additional measures are taken to prevent overfitting in this stage. We use batch normalization and dropout techniques and also employ a variable learning rate that is adjusted during training. Further details regarding the architecture of our MIL-based WFP attack, data dimensions, and selection of hyperparameters are provided in Appendix A.

4.3 Use of Hidden Markov Model

While our preceding set-aware methods account for successive visits to multiple pages within a website, they do not consider their exact visiting order. To check whether this additional leakage can further improve the accuracy of the attack, we create another set-aware method based on an HMM. Although Cai et al. [7] previously used an HMM to model certain user behavior, their study was limited in scope, failing to analyze the effect of the number of pages within a website accessed by the user or the impact of different user behaviors. While we rely on different sources to obtain potential page visiting orders, it is worth noting that such browsing patterns are very user-specific and we argue that it is unrealistic to fully exploit such leakage. As shown in Section 6.2, high accuracies can be obtained by using merely sets of pages without regard to their visiting order.

We create an individual HMM model for each website, where the pages correspond to different states and the state transition probabilities indicate the likelihood of a user moving from one page to another. As most websites comprise a large number of pages, implementing a separate state for each individual page does not scale. Thus, we use clustering to group multiple pages that share similar appearances and link connectivity with other pages on the same website into a single HMM state. To account for varying number of clusters across different websites, we implement the DBSCAN clustering method [18], which does not require any prior knowledge of the number of clusters. The resulting clusters represent hidden states in our HMM model. A separate webpage classifier is then trained on each of these clusters.

Beside the set of hidden states, we also need to define the set of observations, the set of transition probabilities indicating the likelihood of generating a given observation upon transitioning to a certain hidden state, the set of initial probabilities, and the set of observation probabilities to complete the HMM model for each website. The set of observations corresponds to the set of cluster labels predicted for each testing page. To derive the set of transition probabilities, we use two sources of data: a sitemap graph of each website containing available pages with the link relationships between them, and automatically-generated user browsing sessions describing sequences of pages (see Appendix B). The set of start probabilities is the frequency of clusters, counted for a set of training sessions, containing the first page in a session. The set of observation probabilities contains the probability values obtained by the webpage classifier and indicating the likelihood that a given testing page belongs to each of the possible classes. Then, we apply the Viterbi algorithm to find the most likely sequence of hidden states given the sequence

of observations in the HMM for each website. Finally, we sum the predicted probability values of each page for each website and multiply the aggregated probabilities of the pages in the sequence. The website class with the highest likelihood is our final prediction.

4.4 Strengthening Webpage Classifiers

While increasing the number of pages per website and traces per page improves the accuracy of traditional ML-based WFP attacks (see Section 6.1), they only achieve moderate accuracy when fingerprinting websites instead of individual pages. This may be because the manual feature sets used by these methods are not suitable for detecting websites, or since there are too many page types within a single website class, leading to divergent traffic patterns interfering with the classifiers.

To verify our first assumption, we extended our feature generator, presented in Section 4.2, to enable the independent storage of extracted feature vectors in files. We have opted for the set of 100 features per traffic trace as a good choice for our evaluation. Although we explored the use of the autoencoder suggested by Oh et al. [38] and a variational autoencoder, we were unable to achieve any improvement with the features extracted from these methods in our evaluation settings.

To validate our second assumption, we suggest the use of clustering. We cluster training traces of pages that belong to a single website prior to inputting them into a webpage classifier. Intuitively, traces assigned to a single cluster share a high level of similarity. Each of the clusters then forms a separate class for training the classifier. During testing, if a test trace is predicted to belong to a particular cluster, it is assigned to the website associated with that cluster. We tested several clustering methods, including DBSCAN [18], OPTICS [3], ShiftMean [28], Gaussian Mixture Model [53], k-Means [23]. Based on our empirical analysis, we identify k-Means as the most suitable clustering method for our evaluation.

5 Experimental Setup

We present our evaluation setup to allow for verifiable results.

Datasets. Our analysis requires a set of multiple websites, each represented by its respective pages. As there is no such existing dataset, we compiled a new dataset including 100 websites from the Alexa Top list of the most popular websites [16]. To confirm the representativeness of our Alexa list and the soundness of our findings, we created a second validation dataset comprising 100 websites[1] from the latest Tranco Top list of the most popular websites [33]. Both samples contain websites from various categories (e.g., news, social media, online shops) with diverse layouts and content from many regions worldwide. For each website, we selected 80 unique URLs of accessible pages by following links from the index page and using the method presented in [39]: We choose a link with a probability of 0.5 from the first, 0.25 from the second, 0.125 from the third, and 0.0625 from the fourth and fifth result pages of that website. As websites can be accessed from different sources, e.g., visting a browser-cached URL or performing a search engine query [11, 32], we used the same method from [39] to query Google—one of the most popular search engines—and further retrieved ten unique URLs of accessible pages per website. In total, we gathered the URLs of index pages from 100 unique websites, and the URLs of 90 unique non-index pages for each of these websites. We refer to the datasets as `ALEXA-WSC-FG` and `TRANCO-WSC-FG`.

We also visited the 5,000 most popular Alexa websites, excluding the 100 websites in `ALEXA-WSC-FG`, and collected nine unique pages for each website using the same method from [39]. We call this dataset `ALEXA-WSC-BG` and mainly use it for our open-world analysis. The last dataset we created for evaluating our HMM-based method is called `ALEXA-WSC-HMM`. While it includes the same websites as those in `ALEXA-WSC-FG`, `ALEXA-WSC-HMM` comprises URLs of 50 unique non-index pages per website and uses a sitemap graph for each website and automatically-generated user browsing sessions to record the exact page visiting order. Due to space constraints, we refer the reader to Appendix B for details regarding the creation of sitemap graphs and user sessions.

Network Traffic Collection. As in related work [39,42,45], we excluded page loads denying requests coming from Tor, showing a CAPTCHA, having no content, or pointing to a client or server error. We also omitted pages that require user authentication to access their content, as they are usually not compatible with Tor's default configuration[2]. The attacker is not interested in fingerprinting such page loads as none of them can be considered successful in accessing the content. We used the automated approach presented in [39] to gather metadata, such as TCP packet sizes, their direction and timing, for all pages in our datasets. Then, we reconstructed the Tor cells using a data extraction method from [39]. We used the Tor Browser 7.5.6 for all `ALEXA-*` datasets and the latest Tor Browser 12.5.3 for gathering `TRANCO-WSC-FG`. The validation of our results across different Tor Browser versions is essential to establish their soundness (see Section 6.4). In total, we collected 90 traffic traces for each non-index page and 20 traffic traces for each index page in `ALEXA-WSC-FG`, and one traffic trace for each page in `ALEXA-WSC-BG` and `TRANCO-WSC-FG`. We further gathered 10 user sessions per website from `ALEXA-WSC-HMM`, with each session consisting of 10 pages and 20 traffic traces per page.

Evaluation Setup. We use four modern webpage classifiers: CUMUL [39], k-FP [24], DF [45], and Var-CNN [4]. We refer the reader to the original papers for more details on them. For all experiments, unless otherwise stated, we conduct 10-fold cross-validation (CV) based on either the number

[1] https://tranco-list.eu/download/X53GN/500

[2] As Tor often changes the exit node, and, hence, the IP address visible for these pages, this causes the pages to activate their security mechanisms.

of traces per page or the number of pages per website.

6 Evaluation and Discussion

This section analyzes the efficacy of our novel WFP methods. We consider two main evaluation strategies. In the first strategy, called *website with known pages*, we assume that the attacker can obtain fingerprints of all pages belonging to a target website, i.e., traces from all pages of a website are used for training a classifier. In this strategy, we conduct the CV based on the number of available traces per page for each website. While this strategy is suitable for websites with a small number of pages that seldom change, fingerprinting each page of a large website with many pages that frequently change content is practically unfeasible. Thus, we consider a second strategy, called *website with unknown pages*, in which the attacker can only use a portion of the pages available on a website to train a classifier, while the testing is conducted on pages that the classifier has not yet encountered. Here, we conduct the CV based on the number of available pages per website. We computed the *accuracy*, i.e., the probability of a correct prediction (either a true positive or a true negative), for our closed-world analysis and the *F1-score*, i.e., the harmonic mean of precision and recall, for our open-world experiments.

Section 6.1 begins with identifying a suitable training strategy and amount of data for state-of-the-art webpage classifiers for our threat model to enable a fair comparison with our methods. Section 6.2 provides insights into the efficiency of our novel fingerprinting methods in both closed- and open-world scenarios. Section 6.3 presents the level of security provided by state-of-the-art WFP defenses when our novel WFP methods are employed. In Section 6.4, we summarize our main takeaways and discuss potential limitations and future work.

6.1 Webpage versus Website Fingerprinting

Webpage Classifiers for Website Fingerprinting. First, we examine the efficacy of existing webpage classifiers in detecting websites in a closed world, using our dataset `ALEXA-WSC-FG`. Similar to previous work [4,39,45], we chose to use 90 traces for each website class. Beside the classical webpage detection done in previous research that we call *individual page*, we also study another evaluation scenario, where traces of a different page, not seen by the classifier before, are used for testing. We further distinguish between two training strategies. While the first strategy, called *website with single page training*, entails a classifier learning a single page per website, the second strategy, called *website with milti-page training*, involves a classifier being trained on multiple pages of a single website (with one trace per page)[3]. As shown in Table 1 (rows one and two), the accuracy of all methods decreases by around 50% when using a single page for training and aiming to detect a website. This drop indicates that existing attacks lack the robustness needed to make them viable. The use of only one page per website for training is particularly highly insufficient for identifying websites. On the other hand, training the classifiers on multiple pages per website, while keeping the same number of traces per website class (row three in Table 1), leads to an accuracy increase from about 20% (for CUMUL and Var-CNN) to nearly 30% (for k-FP and DF). Overall, the classification results of webpage classifiers, obtained in our more realistic evaluation setting, remain notably lower than expected, demanding dedicated methods to fingerprint websites.

[3] Both cases correspond to our strategy *website with unknown pages*.

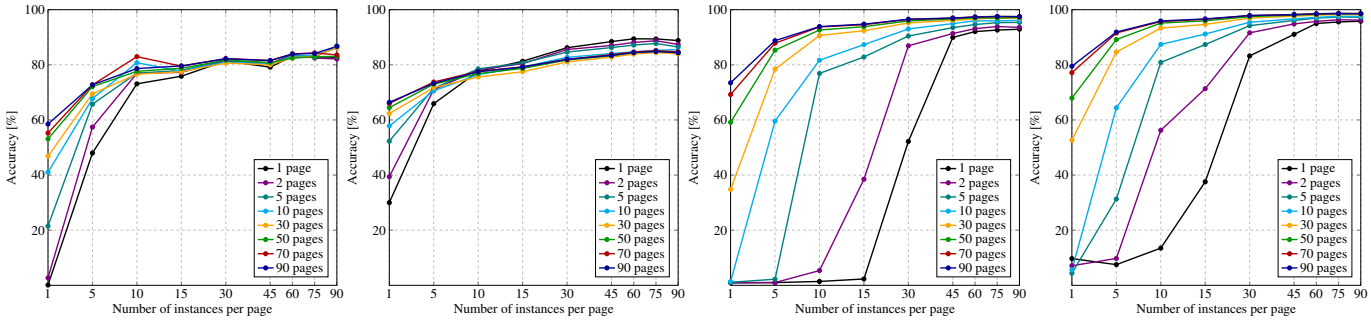
Table 1: Accuracy (in %) of webpage classifiers for fingerprinting individual pages vs. websites.

Scenario	CUMUL	k-FP	DF	Var-CNN
Individual page	82.75	88.81	92.91	95.80
Website with single page training	39.20	33.06	44.10	54.00
Website with multi-page training	56.14	62.98	70.57	76.36

Number of Available Pages for Website Fingerprinting. Next, we examine how the number of available pages for each site impacts the accuracy of the webpage classifiers. For our *website-with-known-pages* strategy, Figure 2 shows that all methods attain higher accuracy with a larger number of pages per site and traces per page in a closed world. Both DL-based attacks can learn a website and achieve over 90% accuracy already with 30 pages, each containing 10 traces. When using even more traces for these pages, the accuracy improves to over 96% for DF and 98% for Var-CNN. Contrariwise, the traditional ML-based methods CUMUL and k-FP achieve around 10% less accuracy than DF and Var-CNN when training on a larger number of pages and traces per page.

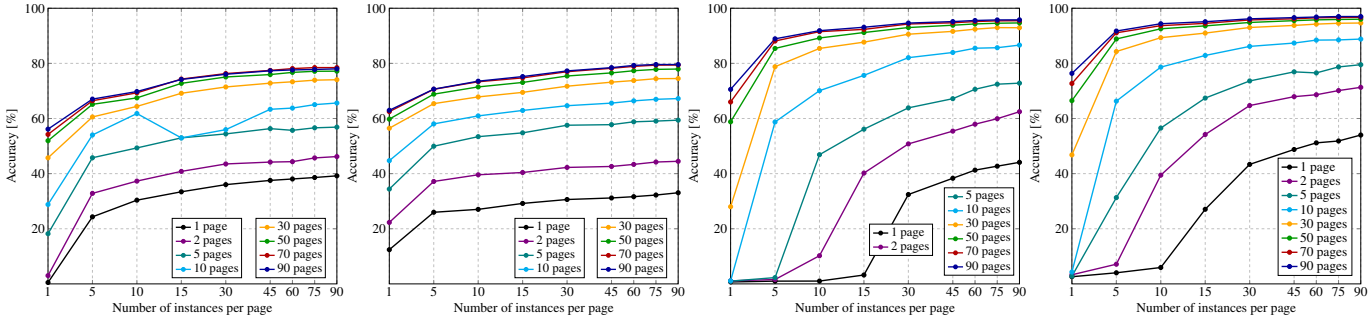
For our *website-with-unknown-pages* strategy, Figure 3 shows an overall lower accuracy for all attacks, compared to the first strategy. It is especially significant when the classifiers learn fewer than 10 pages per website. Although increasing the number of traces for this set of pages improves accuracy by roughly up to 40% for DF and Var-CNN and up to 20% for CUMUL and k-FP, the highest achieved detection rate for the attacks remains below 60% for CUMUL and k-FP, around 70% for DF, and less than 80% for Var-CNN. All webpage classifiers require a minimum of 50 pages per website, each containing at least 10 traces, to effectively learn websites. This indicates a substantial rise in the amount of data needed to train these classifiers, requiring about 500 or more traces per class instead of 90. While the DL-based methods DF and Var-CNN can achieve over 95% accuracy with a large number of pages per site and traces for them, both traditional ML-based methods CUMUL and k-FP attain an accuracy of less than 80% for all sets of pages and traces per page. Still, none of these attacks scales when used to detect websites in real-world settings (see Section 6.2.3).

More Training Pages vs. More Training Traces per Page. We assess the importance of the number of pages vs. the num-



(a) CUMUL (b) k-FP (c) DF (d) Var-CNN

Figure 2: Existing WFP methods with varying number of pages per website and traces per page for *website with known pages*.



(a) CUMUL (b) k-FP (c) DF (d) Var-CNN

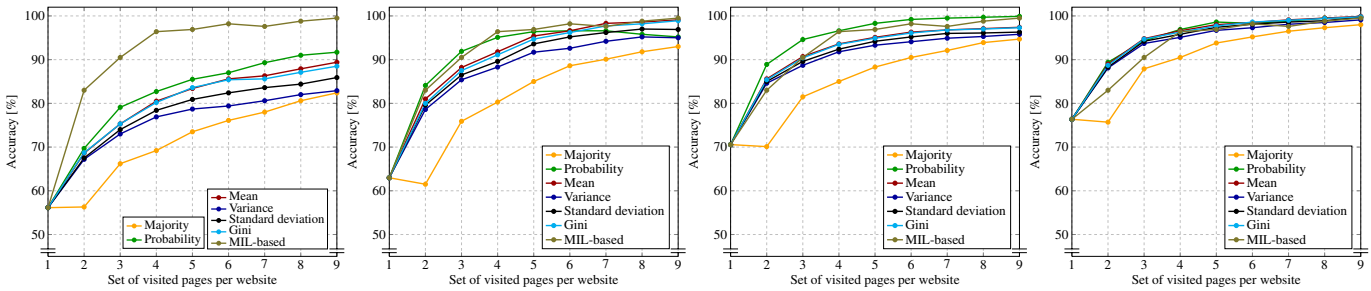
Figure 3: Existing WFP methods with varying number of pages per website and traces per page for *website with unknown pages*.

ber of traces per page to achieve higher accuracies. We only consider the results from Figures 2 and 3 that are obtained when the classifiers cumulatively rely on the same number of training samples, but with varying numbers of pages and traces per page, e.g., 10 pages per site with 90 traces per page vs. 30 pages per site with 30 traces per page. The trend of improved accuracy with the use of larger number of pages noted in Table 1 is less evident, when the size of the training data increases. Thus, we conclude that a variety of training pages per site is crucial when dealing with smaller datasets.

Index Pages for Training. We evaluated whether the use of the index pages is particularly important to detect websites, as doubted in [39]. Based on our analysis (see Appendix C), we conclude that the inclusion of index pages in training does not affect the accuracy of the attacks in either evaluation strategy.

Timing Information for Website Fingerprinting. Two of the evaluated classifiers, k-FP and Var-CNN, rely not only on packet order and size, but also on timing information. As shown in Figures 2 and 3, both classifiers achieve higher accuracy for very small datasets, i.e., less than 10 pages per site and less than 10 traces per page. The increase in accuracy is particularly evident in the results for k-FP. The k-FP's accuracy surpasses 40% for our *website-with-known-pages* strategy and is more than 20% higher than the accuracies of the other attacks when the *website-with-unknown-pages* strategy is used. Although Var-CNN has only a slightly increased accuracy for very small datasets, it still achieves over a 30% higher accuracy when using 5 pages per site and 5 traces per page than its DL-based competitor DF. As the number of pages per site and the number of traces per page increase, the effect of additional timing information on accuracy disappears.

Strengthening Traditional ML-based Classifiers. As stated in Section 4.4, the moderate accuracy of both traditional ML-based attacks can be attributed to suboptimal feature sets or too dissimilar traffic patterns across distinct pages of a website. We analyze whether the use of new feature sets or applying clustering methods in these attacks to account for diverse website structures can enhance their effectiveness. We conducted multiple experiments using our strategy *website with unknown pages* and a dataset of 90 pages per website with 15 traces per page. The dataset size is suitable for this evaluation as the attacker can already achieve high accuracy in detecting websites using both DL-based competitors with such a dataset. While the first experiment involved learning all training pages of a given site in a single class (as previously done), the second experiment included grouping the training pages into multiple clusters and learning each cluster separately (see Section 4.4). Both experiments are performed using the original features of the corresponding webpage classifiers. Then, we replicated these experiments, but this time with the feature set produced by our feature generator (see Section 4.4). Table 2 shows the results obtained. Regardless of the feature set used, the use of clustering does not improve the accuracy of both classifiers. However, the newly generated



(a) Voting (CUMUL) vs. MIL-based (b) Voting (k-FP) vs. MIL-based (c) Voting (DF) vs. MIL-based (d) Voting (Var-CNN) vs. MIL-based

Figure 4: Accuracy achieved by our voting-based strategies with state-of-the-art webpage classifiers and our MIL-based method.

Table 2: Accuracy (in %) of traditional ML-based classifiers in combination with clustering and new feature sets.

		Without clustering		With clustering			
		Original feature set	New feature set	Original feature set	New feature set	DF	Var-CNN
Website with known pages	CUMUL	79.59	95.73	77.35	95.78	94.74	96.65
	k-FP	79.18	93.62	78.56	93.50		
Website with unknown pages	CUMUL	74.17	93.83	70.43	93.86	93.11	95.11
	k-FP	75.21	91.69	74.50	91.46		

features increase their accuracy by almost 20%, making them as efficient as the DL-based methods. This suggests that the original feature sets may not be suitable for detecting websites as opposed to individual pages, and thus need to be revised.

6.2 Analysis of Our Fingerprinting Methods

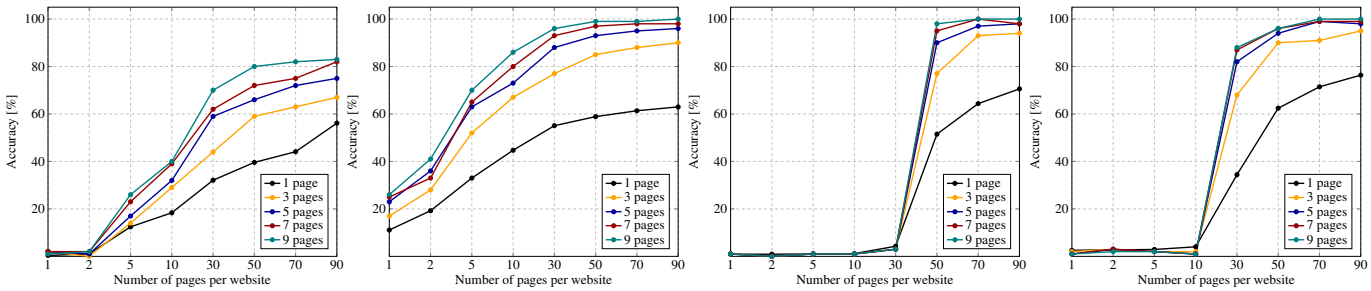
This section evaluates the efficiency of our novel fingerprinting methods, proposed in Section 4.1 and 4.2. We focus on our more realistic and, at the same time, more difficult evaluation strategy *website with unknown pages*. We assume that the attacker uses multiple pages per website to train a classifier. We consider the results from Table 1, row three, as the baseline for our closed-world analysis and use the same dataset consisting of 90 pages per website and one trace per page.

6.2.1 Use of Voting-based Strategies

Efficiency of Different Voting-based Strategies. Figure 4 shows the accuracies achieved by combining each webpage classifier with each of our voting-based strategies. Regardless of the chosen strategy, we observe an overall significant increase in the detection rates of all webpage classifiers as the number of successively visited pages belonging to a single website increases. We further notice that the majority voting—our simplest strategy—already provides a significant increase in accuracy (up to 90%) for three of the classifiers (k-FP, DF, and Var-CNN) for six consecutive page visits. Still, for short sets of visited pages we observe only a slow increase in accuracy, while for two consecutive pages the accuracy of three of the classifiers (k-FP, DF, and Var-CNN) even slightly decreases. The main reason is that more often there is no predominant website class in a set of two pages, and the final decision should practically be guessed in controversial cases.

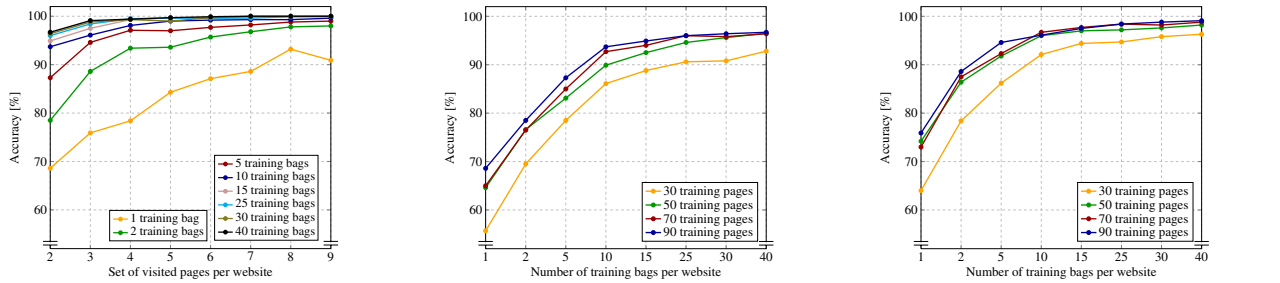
When the mean voting is used, all classifiers experience a boost in accuracy of roughly 10% to 20% compared to majority voting as the number of visited pages increases. For k-FP, this increase is notably evident for a set of two page visits where we see a detection rate over 20% higher compared to the one obtained with majority voting. The accuracies attained by the variance, standard deviation, and Gini-weighted mean strategies closely approximate those achieved by mean voting. The Gini-weighted mean strategy performs better than the other two strategies on most sets of pages, but the differences in accuracy compared to mean voting are negligible. When employing the probability voting, a notable boost in accuracy is evident among all classifiers in comparison to the other strategies discussed above. The probability voting differs from majority voting in that it only requires two clicks on a website for nearly 90% accuracy using DF and Var-CNN. With one more click, both DL-based classifiers improve to a detection rate of 95% accuracy and k-FP achieves more than 90%. Additionally, DF and Var-CNN achieve an accuracy of almost 100% for sets of more than seven page visits. Although probability voting has shown higher detection rates than the other voting-based strategies when used with CUMUL, DF, and Var-CNN, accuracy degradation occurs with k-FP when test sets contain more than five pages. This may arise because k-FP produces zero probabilities for website classes, which can harm the computation of probability voting. Therefore, we selected probability voting as the most effective voting-based strategy for CUMUL, DF, and Var-CNN in all subsequent experiments. Mean voting was selected for k-FP.

Impact of Different Training Tactics. We also examine how the number of training pages per website impacts the accuracy of our best voting strategy employed with every webpage classifier. We conduct experiments where we change the number of training pages, ranging from one to 90 pages per website, while maintaining the same testing set as mentioned above. As shown in Figure 5, achieving a higher accuracy requires more user clicks per website when the number of training pages decreases. When there are 90 training pages available, the user can be deanonymized with nearly 95% accuracy when



(a) CUMUL (b) k-FP (c) DF (d) Var-CNN

Figure 5: Accuracy achieved by our best voting strategy per classifier, when the number of training pages increases.



(a) Increasing number of training bags for a set of 90 pages per website.

(b) Test bag of two pages with increasing training pages and bags per website.

(c) Test bag of three pages with increasing training pages and bags per website.

Figure 6: Accuracy achieved by our MIL-based attack in different evaluation settings.

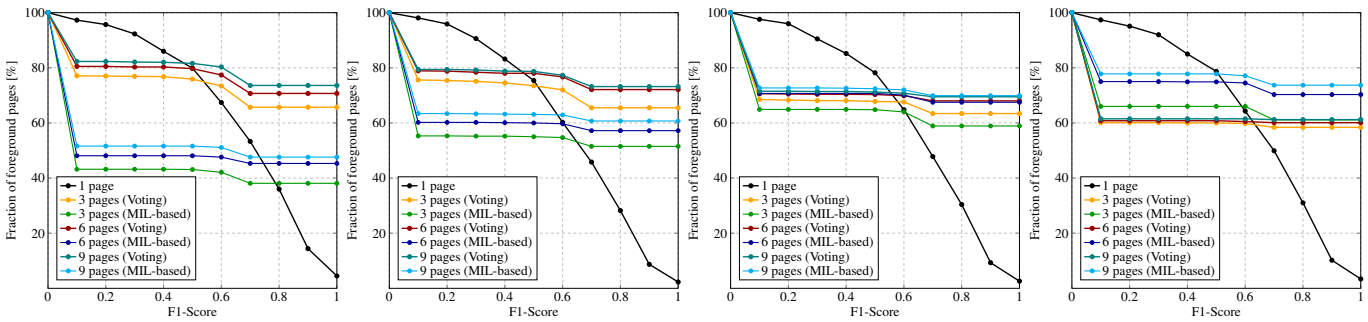
visiting only three pages on a website using k-FP, DF, and Var-CNN. However, this number increases for k-FP and Var-CNN to five pages, and for DF to seven pages, when relying on 50 training pages. When using only 30 training pages, the attacker needs up to nine pages for k-FP and Var-CNN to achieve a similar detection rate. It can be also seen that 70 training pages are enough to obtain the same accuracies for all the classifiers as those obtained with 90 training pages. Further training tactics are discussed in Appendix D.

6.2.2 Our MIL-based Attack

We proceed with the evaluation of our MIL-based method. Figure 4 shows the accuracy achieved by it compared to our voting-based strategies. Similar to the voting-based methods, we see a significant increase in the accuracy of our MIL-based attack as the number of sequentially accessed pages within a website increases. In particular, it achieves an efficiency that is comparable to that of our voting-based strategies when four or more pages per website are accessed. Still, this increase is smaller for a set of two or three pages per website. While the number of training bags used by our MIL-based method is always the same in all experiments, we have fewer traces per bag due to the smaller number of pages accessed per website, which reduces the accuracy for smaller numbers of user clicks.

Number of Training Bags per Website. In response, we examine the influence of the number of training bags on the accuracy of our MIL-based attack. For this analysis, we use a variable number of training bags for our weight learner while maintaining the same testing set as in the previous experiment. In addition, our feature generator is trained with all possible traces that can be found in the training bags, eliminating any potential bias in our results. As shown in Figure 6a, we see that one training bag is already sufficient to obtain a detection rate of over 90% when accessing eight or more pages within a website. As the number of training bags increases, the number of pages that need to be consecutively visited to achieve high accuracy decreases. In particular, just one additional training bag is necessary to reduce the set of observed user clicks to four and to still reach over 90% accuracy. Overall, the use of two training bags increases the detection rate by nearly 20% for sets of up to four pages, and by about 10% for larger sets of five or more pages. While the accuracy of smaller sets, up to four pages, is boosted by almost 10% when using five training bags, this boost begins to degrade gradually for larger numbers of training bags. Concurrently, our MIL-based attack achieves 100% accuracy for bigger sets consisting of seven, eight, or nine pages when using 15 training bags or more.

Number of Training Pages per Website. Finally, we analyze how the number of training pages per website affects the accuracy of our MIL-based attack. In particular, we examine how many training bags are needed for different number of training pages per website to achieve a high accuracy. Similar to the experiments above, we use an increasing number of training bags for each set of training pages, while keeping the test set the same. We also focus on the detection rates achieved when the user accesses two and three pages consecutively (in Section 6.2.1, we discovered that two, at most three,



(a) 500 background websites (b) 1000 background websites (c) 2000 background websites (d) 5000 background websites

Figure 7: F1-score for our MIL-based attack and out best voting stategy compared to the F1-score for webpage fingerprinting.

user clicks are sufficient to achieve more than 90% accuracy with our voting-based strategies). As shown in Figure 6b, the attacker can deanonymize a user with nearly 70% accuracy by observing only two consecutive pages that a user visits and solely two training bags for a training set of 30 pages per website. Figure 6c also shows that an additional user click can boost this accuracy by around 10%. While this result is similar to the one achieved by using a combination of k-FP with our voting-based methods and an equal number of training pages per website and testing user clicks, it is about 10% and 20% greater than the detection rates obtained for Var-CNN and CUMUL, respectively. Moreover, it is up to sixteen times higher than the accuracy obtained for DF. In particular, we see that for a set of 30 training pages per website, the attacker can achieve almost up to 93% and 97% accuracy when using 25 or more training bags. Furthermore, we observe that the detection rate of our MIL-based attack improves by about 7% to 10% for all sets of training bags when the attacker uses a set of 50 training pages per website. For larger numbers of training pages per website, the accuracy increase is negligible. Overall, it is evident from Figures 6b and 6c that the accuracy of our MIL-based attack is primarily affected by the number of training bags, rather than the number of different training pages per website.

6.2.3 Open-World Evaluation

This section assesses the efficiency of both our best voting-based strategy and our MIL-based method in an open world. For our experiments, we train a separate classifier for each foreground website (taken from `ALEXA-WSC-FG` as in the previous experiments) that the attacker aims to detect and use an increasing number of websites from `ALEXA-WSC-BG` as background, representing the universe. As described in Section 5, each background website comprises nine pages, resulting in a universe size of up to 45,000 traffic traces—a magnitude commonly used in related work [43, 45]. For the baseline and our voting-based strategy, we report only those classification results obtained with the best performing webpage classifier due to space constraints. Based on our analysis in Section 6.2.2, we further choose to use 40 training bags for each foreground website for our MIL-based method. Figure 7 illustrates the obtained results. We see that for the largest number of background websites, only three clicks are sufficient to deanonymize a user with a F1-score of 1.0 for roughly 60% of the foreground websites, while traditional webpage fingerprinting cannot detect almost any website. As the number of consecutive page visits increases, the percentage of foreground websites detected with a F1-score of 1.0 increases to almost 75%. This huge gain in the detection rate through our methods remains also for smaller values of F1-score. Using our novel fingerprint methods, the percentage of websites detected with a 90% or higher F1-score increases by over 50%. Overall, despite the increasing background sizes, both our best voting-based strategy and our MIL-based method achieve a steady and constant F1-score.

6.2.4 Evaluation of our HMM-based Strategy

Finally, we analyze our HMM-based method, proposed in Section 4.3. To do this, we use the dataset `ALEXA-WSC-HMM` and combine the worst-performing classifier CUMUL (cf. Section 6.1) with our HMM-based method. As our evaluation shows, such classifiers benefit most from observing the sequence of visited pages. In the first evaluation strategy, we assume that the attacker can learn all user sessions, i.e., the HMM contains transition information for all possible user sessions. This strategy is an adapted version of *website with known pages*, where we test WFP attacks using different traces of the known pages per website. As shown in Figure 8, the accuracy increases significantly for all test sets of pages of length of two or higher. In the second evaluation strategy, we assume that user sessions used for testing are unknown to the attacker. In most of the cases, this scenario leads to testing on pages per website that are also unknown to both CUMUL and HMM. The accuracy decreases slightly compared to the scenario when the sessions are known. Still, we see a similar positive trend from the use of sequence of pages. The negative effect of unknown sessions is prominent for short sets of pages and decreases when using longer sessions. For

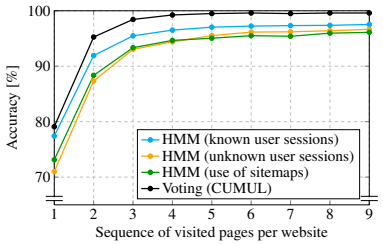


Figure 8: Accuracy achieved by our HMM-based strategy.

a session of nine pages, the accuracy is 96.8% compared to 97.8% when all sessions are known. Finally, instead of learning transition probabilities from user sessions for HMM, we use the sitemap graphs of the target websites. As shown in Figure 8, the use of sitemap graphs to compute the transition probabilities is beneficial for user sessions of at most four pages. Still, the differences in accuracy across the different use cases are negligible. Despite the improvements in accuracy, our HMM-based method is not able to surpass our best voting strategy, leading to the conclusion that knowing the exact order of the visited pages does not primarily affect the classifier's accuracy.

6.3 Effectiveness against WFP Defenses

This section analyzes the robustness of our novel fingerprinting attacks against WFP defenses. We consider six state-of-the-art methods: Tamaraw [6], CS-Buflo [5], WTF-PAD [30], TrafficSliver-Net [13], FRONT [20], and RegulaTor [26]. While TrafficSliver-Net currently stands as one of the most effective and prominent practical defenses [35], CS-Buflo and Tamaraw are among the state-of-the-art methods that result in one of the lowest accuracies for WFP attacks, despite being developed only as a simulation. WTF-PAD refers to a group of WFP defenses that offer easy practical deployment while having a lower level of security. FRONT and RegulaTor are newer WFP defenses that promise both increased security and low implementation costs. For our analysis, we created defended traces for each defense using their public implementations with the optimal performing parameters reported in the original papers. We also used the `ALEXA-WSC-FG` dataset, which consists of 90 pages per website, with 15 traces per page and considered the evaluation strategy *website with unknown pages*. We train the webpage classifiers with a sufficiently large number of traces, ensuring that we achieve high accuracy for our baseline. As shown in Table 3, the accuracy of our novel methods increases when the user sequentially browses multiple pages on a website despite the use of a WFP defense. Although the detection rate for the strong defense Tamaraw remains under 20%, the increase is still concerning, having risen almost fivefold (i.e., from 4.61% to nearly 20%). CS-BUFLO shows a similar increase in accuracy as Tamaraw. Even worse, here the attacker's accuracy is over 50% for a set of nine sequentially accessed pages. For all sets of visited pages, the attacker achieves about a 10% higher detection rate with RegulaTor compared to CS-BUFLO. Although the accuracy achieved for the different sets of pages in the case of TrafficSliver-Net is still below 50%, it has approximately doubled. Both defenses WTF-PAD and FRONT become practically useless when the number of user clicks within a website increases. These results indicate that our attacks are not only more realistic, but also more dangerous.

Table 3: Accuracy (in %) of our attacks against WFP defenses.

Defense	Classifier	Set of pages								
		1	2	3	4	5	6	7	8	9
Tamaraw	Voting	4.61	7.20	9.93	12.47	12.67	14.07	16.27	17.80	18.93
	MIL-based	–	5.37	7.25	8.81	10.79	12.19	13.77	14.73	16.36
CS-Buflo	Voting	10.89	18.13	23.33	33.27	37.40	43.93	46.93	52.47	56.00
	MIL-based	–	12.89	19.25	24.77	29.62	34.19	37.18	40.21	43.17
TrafficSliver-Net	Voting	19.92	29.93	34.48	38.73	40.45	42.79	43.80	44.85	46.55
	MIL-based	–	10.40	14.48	18.62	22.06	25.67	28.69	32.18	35.21
WTF-PAD	Voting	90.72	99.20	99.73	100.00	100.00	100.00	100.00	100.00	100.00
	MIL-based	–	98.28	99.61	99.89	99.99	99.99	99.99	100.00	100.00
RegulaTor	Voting	17.17	27.67	38.27	44.20	50.20	56.20	61.53	63.60	64.87
	MIL-based	–	16.11	22.83	27.77	31.89	36.19	40.29	43.44	46.48
FRONT	Voting	67.00	88.60	96.87	98.73	99.40	99.67	99.87	99.93	100.00
	MIL-based	–	86.41	94.82	97.70	98.85	99.38	99.55	99.77	99.86

6.4 Discussion and Limitations

Unlike previous research that aim at detecting only individual pages via isolated page loads, our methods are able to (i) account for sequential visits to pages of a single website and (ii) detect the website a user visits, regardless of the page accessed within that website. If further desired, existing works such as [36, 44] can be used to conduct webpage detection in a secondary step, once our methods have detected the website. We present a more realistic assessment of the degree to which Tor users are vulnerable to WFP, in contrast to previous work that downplays the threat of WFP in Tor and gives an overestimated impression of the level of security provided.

Limitations. The main limitation of our work is the assumption that the attacker can distinguish and segregate consecutive visits to multiple pages associated with a given website. We argue that this is the worst-case scenario for a Tor user who is fixated on the content of a single website, and the segmentation of the visited pages from said website is also possible (e.g., using [50]). We further evaluate what happens if this assumption does not hold, i.e., how noise pages impact the accuracy of our methods. Noise pages refer to one or more pages in the set of visited pages that do not belong to the target website. Similar to Section 6.2, we examine the *website-with-unknown-pages* strategy, use multiple pages per website (excluding noise pages) to train a classifier, and use the dataset comprising 90 pages per site and one trace per page. We compute the accuracy for a set of nine accessed (i.e., testing) pages in a closed world, while the number of randomly selected noise pages in this set ranged from one to four. As shown in Table 4 (rows one and two), the more noise pages are present, the lower accuracy our methods achieve. Still, the accuracy of our best voting-based method is only minimally

Table 4: Accuracy (in %) of our MIL-based method and our voting-based stategies for increasing number of noise pages.

Classifier	Pages from a website + noise pages				Pages from a website			
	8+1	7+2	6+3	5+4	8	7	6	5
MIL-based	96.30	92.90	77.50	68.80	98.80	97.60	98.20	96.90
Our best voting	99.50	98.30	93.20	80.80	99.70	99.50	99.20	98.30
Gini-weighted mean voting	99.40	99.10	98.40	97.00	99.40	99.00	98.60	97.70

impacted when up to three noise pages are present in the set, resulting in no more than a 6% decrease. On the other hand, our MIL-based method is slightly more affected with a 5% reduction in accuracy for two noisy pages and almost a 20% decrease for three noisy pages. In case of four noise pages, the accuracies of both methods notably decrease, i.e., by around 20% for our best voting-based method and almost 30% for our MIL-based method. However, it should be noted that in the latter case the four noise pages are already almost half of the whole set of visited pages, which is a significant amount of noise. It is also noteworthy that according to research on user browsing behavior [11], on average users navigate to the first noise page after clicking five times within the target website, which is already enough for our methods to detect that target website without any confusion from noise pages (see Figure 4). We also check whether one of our three voting-based strategies using weighting methods are the better choice for this evaluation setting. As shown in row three in Table 4, the use of the Gini-weighted mean voting is already sufficient to filter out all possible noise pages and, thus, not negatively affect the accuracy of the attack. We believe that—in future work—the performance our MIL-based method can be further optimized.

Our work further focused on a scenario where a user visits websites in a single tab—the worst case from user's perspective. However, there are user groups that browse the web in multiple tabs. As our voting-based strategies are classifier-independent, they can be combined with existing webpage classifiers [21, 51, 52] that focus on multi-tab detection of individual pages, thus boosting their otherwise limited accuracy for website fingerprinting. Based on the lessons learned from [14], our MIL-based method can be further improved to account for more appropriate patterns for detecting websites in a multi-tab environment, which we leave for future work.

Different Datasets & Tor Browser Versions. To validate the soundness of our evaluation results, we repeated some of our key experiments using the dataset `TRANCO-WSC-FG` collected with the Tor Browser 12.5.3. Our verification shows no significant difference in the evaluation results, which again confirms our findings. Worse, we observe higher accuracies achieved with `TRANCO-WSC-FG` compared to `ALEXA-WSC-FG`. We refer the reader to Appendix E for more details.

Takeaways. In a closed world, existing state-of-the-art DL-based webpage classifiers can detect a website with high accuracy without considering the set of visited pages of that website at the cost of using a huge amount of training data. In contrast to our novel fingerprinting methods, they require for website fingerprinting five times or more traces for training—often unavailable in practice—than reported in the original papers for webpage fingerprinting. Further, existing state-of-the-art traditional ML-based webpage classifiers can only achieve moderate classification results, regardless of the number of training traces, when applied to detecting websites. While the use of clustering did not improve the accuracy of the original implementations, the newly created features increase the accuracy of the classifiers by about 20%, making them competitive with the state-of-the-art DL-based WFP attacks. Despite these closed-world results, all existing webpage classifiers fail to detect almost any website in real-world settings. Contrary to that, we have shown that both our voting-based strategies and our MIL-based attack are effective measures to enhance WFP and increase its feasibility in the real world. Our novel methods achieve F1-scores of 1.0 for more than 60% of the target websites. Although previous work [10, 39] showed limited scalability of WFP attacks in the real world, our results suggest the need to reconsider these evaluations in the context of our revisited evaluation settings.

We have shown that the worse the performance of the webpage classifier is, the more its performance can be improved by considering a set of visited pages and using our voting-based strategies. Hence, webpage classifiers trained with little data (or not so efficient ones) benefit significantly from our methods. The use of training traces corresponding to different pages of a website is also critical when dealing with small datasets, regardless of the fingerprinting methods used. Finally, the effectiveness of state-of-the-art defenses is drastically hampered by our improved attacks. Defenses with lower deployment costs, which make them attractive candidates for adoption in Tor, are particularly affected by our methods.

7 Conclusion

Understanding the feasibility and limitations of WFP attacks is crucial to assess the degree of protection offered to Tor users. Prior work mainly focused on detecting individual pages via isolated page loads, rather than websites, and ignored information regarding consecutive user visits to pages within a website. In response, we proposed both a novel WFP attack and effective strategies that leverage information on consecutive page visits within a website and aim at detecting websites, regardless of the specific page accessed. With our methods, we showed that merely two, at most three, clicks within a website are sufficient to successfully deanonymize a Tor user with nearly 100% accuracy in a closed world. In real-world settings, our methods achieve F1-scores of 1.0 for over half of the target websites, whereas existing classifiers fail to detect nearly any website. Finally, we showed that our novel methods remain robust against WFP defenses, achieving 2.5

to 5 times the accuracy of prior work, and in some cases even rendering the defenses useless. Overall, our work showed that WFP presents a much greater risk to the privacy of Tor users who visit multiple pages within a website and the need for further research to provide adequate protection. The source code of our methods is available online [1].

Acknowledgments

We thank our anonymous shepherd and all reviewers for their valuable feedback. Parts of this work have been funded by the German Federal Office for Information Security (BSI) under the project medCS.5, and the German Federal Ministry of Education and Research (BMBF) under the projects Energy Innovation Center (EIZ), KISS_KI, and WAIKIKI.

References

[1] https://www.informatik.tu-cottbus.de/~andriy/zwiebelfreunde/methods-usenix-sec-2024/.

[2] Kota Abe and Shigeki Goto. Fingerprinting Attack on Tor Anonymity using Deep Learning. In *Asia Pacific Advanced Network Workshop*, APAN, 2016.

[3] Mihael Ankerst, Hans-Peter Breunig, Markus M. and Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. *SIGMOD Record*, 28(2):49–60, June 1999.

[4] Sanjit Bhat, David Lu, Albert Kwon, and Srinivas Devadas. Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning. *Proceedings on Privacy Enhancing Technologies*, 2019(4):292–310, 2019.

[5] Xiang Cai, Rishab Nithyanand, and Rob Johnson. CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense. In *13th Workshop on Privacy in the Electronic Society*, WPES, Scottsdale, AZ, USA, 2014. ACM.

[6] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A Systematic Approach to Developing and Evaluating Website Fingerprinting Defenses. In *21st Conference on Computer and Communications Security*, CCS, pages 227–238, Scottsdale, AZ, USA, November 2014. ACM.

[7] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a Distance: Website Fingerprinting Attacks and Defenses. In *19th Conference on Computer and communications security*, CCS, pages 605–616, Raleigh, NC, USA, October 2012. ACM.

[8] Marc-André Carbonneau, Veronika Cheplygina, Eric Granger, and Ghyslain Gagnon. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition*, 2016(77):329–353, 2018.

[9] Lidia Ceriani and Paolo Verme. The origins of the gini index: Extracts from variabilità e mutabilità (1912) by corrado gini. *The Journal of Economic Inequality volume*, 10(3):421–443, 2012.

[10] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World. In *USENIX Security Symposium*, Boston, MA, USA, August 2022. USENIX Association.

[11] Kyle Crichton, Nicolas Christin, and Lorrie Faith Cranor. How Do Home Computer Users Browse the Web? *ACM Transactions on the Web*, 16(1), September 2021.

[12] Weiqi Cui, Tao Chen, Christian Fields, Julianna Chen, Anthony Sierra, and Eric Chan-Tin. Revisiting Assumptions for Website Fingerprinting Attacks. In *Asia Conference on Computer and Communications Security*, AsiaCCS, pages 328–339, Auckland, New Zealand, 2019. ACM.

[13] Wladimir De la Cadena, Asya Mitseva, Jens Hiller, Jan Pennekamp, Sebastian Reuter, Julian Filter, Thomas Engel, Klaus Wehrle, and Andriy Panchenko. Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS, pages 1971–1985, Virtual Event, USA, 2020. ACM.

[14] X. Deng, Q. Yin, Z. Liu, X. Zhao, Q. Li, M. Xu, K. Xu, and J. Wu. Robust multi-tab website fingerprinting attacks in the wild. In *Symposium on Security and Privacy*, S&P, pages 1005–1022, San Francisco, CA, USA, may 2023. IEEE Computer Society.

[15] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *13th Conference on USENIX Security Symposium*, pages 303–320, San Diego, CA, USA, August 2004. USENIX Association.

[16] Expired Domains. Alexa Top 100 Most Popular Websites. https://www.expireddomains.net/alexa-top-websites/, 2023. (Lastly Accessed: October 2023).

[17] Kevin Dyer, Scott Coull, Thomas Ristenpart, and Thomas Shrimpton. Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail. In *33rd IEEE Symposium on Security and Privacy*, S&P, San Francisco, CA, USA, May 2012. IEEE.

[18] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Second International Conference on Knowledge Discovery and Data Mining*, KDD, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[19] Rafael Gálvez, Marc Juarez, and Claudia Diaz. Profiling Tor Users with Unsupervised Learning Techniques. In *International Workshop on Inference and Privacy in a Hyperconnected World*, INFER, Darmstadt, Germany, July 2016. Springer.

[20] Jiajun Gong and Tao Wang. Zero-delay Lightweight Defenses against Website Fingerprinting. In *29th USENIX Security Symposium*, Boston, MA, USA, August 2020. USENIX Association.

[21] Xiaodan Gu, Ming Yang, and Junzhou Luo. A Novel Website Fingerprinting Attack against Multi-tab Browsing Behavior. In *19th International Conference on Computer Supported Cooperative Work in Design*, CSCWD, pages 234–239, Calabria, Italy, 2015. IEEE.

[22] Zhong Guan, Gang Xiong, Gaopeng Gou, Zhen Li, Mingxin Cui, and Chang Liu. BAPM: Block Attention Profiling Model for Multi-tab Website Fingerprinting Attacks on Tor. In *Annual Computer Security Applications Conference*, ACSAC, pages 248–259, Virtual Event, USA, December 2021. ACM.

[23] John A. Hartigan and Manchek A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[24] Jamie Hayes and George Danezis. k-fingerprinting: A Robust Scalable Website Fingerprinting Technique. In *25th USENIX Conference on Security Symposium*, Austin, TX, USA, August 2016. USENIX Association.

[25] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier. In *Workshop on Cloud Computing Security*, CCSW, Chicago, IL, USA, November 2009. ACM.

[26] Nicholas Holland, James K. and Hopper. RegulaTor: A Straightforward Website Fingerprinting Defense. *Proceedings on Privacy Enhancing Technologies*, 2022(2):344–362, 2022.

[27] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning, 2018.

[28] Xin Jin and Jiawei Han. *Mean Shift*, pages 806–808. Springer, Boston, MA, USA, 2017.

[29] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A Critical Evaluation of Website Fingerprinting Attacks. In *21st Conference on Computer and Communications Security*, CCS, pages 263–274, Scottsdale, AZ, USA, November 2014. ACM.

[30] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an Efficient Website Fingerprinting Defense. In *21st European Symposium on Research in Computer Security*, ESORICS, Heraklion, Greece, 2016. Springer.

[31] Sheharbano Khattak, Taria Elahi, Laurent Simon, Colleen M. Swanson, Steven J. Murdoch, and Ian Goldberg. SoK: Making Sense of Censorship Resistance Systems. In *16th Privacy Enhancing Technologies Symposium*, PETS, pages 37–61, Darmstadt, Germany, July 2016. DE GRUYTER.

[32] Ravi Kumar and Andrew Tomkins. A Characterization of Online Browsing Behavior. In *19th International Conference on World Wide Web*, WWW, pages 561–570, Raleigh, NC, USA, April 2010. ACM.

[33] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *26th Annual Network and Distributed System Security Symposium*, NDSS, San Diego, CA, USA, February 2019. Internet Society.

[34] Robert I. Lerman and Shlomo Yitzhaki. A note on the calculation and interpretation of the gini index. *Economics Letters*, 15(3–4):363–368, 1984.

[35] N. Mathews, J. K. Holland, S. Oh, M. Rahman, N. Hopper, and M. Wright. Sok: A critical evaluation of efficient website fingerprinting defenses. In *Symposium on Security and Privacy*, S&P, pages 969–986, San Francisco, CA, USA, May 2023. IEEE.

[36] Brad Miller, Ling Huang, A. D. Joseph, and J. D. Tygar. I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis. In *Privacy Enhancing Technologies*, PETS, pages 143–163, Amsterdam, The Netherlands, July 2014. Springer International Publishing.

[37] Se Eun Oh, Nate Mathews, Mohammad Saidur Rahman, Matthew Wright, and Nicholas Hopper. Pgandalf: Gan for data-limited fingerprinting.

[38] Se Eun Oh, Saikrishna Sunkam, and Nicholas Hopper. p^1-FP: Extraction, Classification, and Prediction of Website Fingerprints with Deep Learning. *Proceedings on Privacy Enhancing Technologies*, 2019(3):191–209, 2019.

[39] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. Website Fingerprinting at Internet Scale. In *23rd Annual Network and Distributed System Security Symposium*, NDSS, San Diego, CA, USA, February 2016. Internet Society.

[40] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website Fingerprinting in Onion Routing Based Anonymization Networks. In *10th Annual ACM Workshop on Privacy in the Electronic Society*, WPES. ACM, October 2011.

[41] Mohammad Saidur Rahman, Payap Sirinam, Nate Mathews, Kantha Girish Gangadhara, and Matthew Wright. Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks. *Proceedings on Privacy Enhancing Technologies*, 2020(3):5–24, 2020.

[42] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated Website Fingerprinting through Deep Learning. In *25th Annual Network and Distributed System Security Symposium*, NDSS, San Diego, CA, USA, February 2018. Internet Society.

[43] Meng Shen, Kexin Ji, Zhenbo Gao, Qi Li, Liehuang Zhu, and Ke Xu. Subverting website fingerprinting defenses with robust traffic representation. In *USENIX Security Symposium*, USENIX Security. USENIX Association, August 2023.

[44] Meng Shen, Yiting Liu, Liehuang Zhu, Xiaojiang Du, and Jiankun Hu. Fine-Grained Webpage Fingerprinting Using Only Packet Length Information of Encrypted Traffic. *IEEE Transactions on Information Forensics and Security*, 16:2046–2059, 2021.

[45] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *25th Conference on Computer and Communications Security*, CCS, Toronto, ON, Canada, October 2018. ACM.

[46] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-Shot Learning. In *26th Conference on Computer and Communications Security*, CCS, London, United Kingdom, November 2019. ACM.

[47] Michael Carl Tschantz, Sadia Afroz, Anonymous, and Vern Paxson. SoK: Towards Grounding Censorship Circumvention in Empiricism. In *Symposium on Security and Privacy*, S&P, pages 914–933, San Jose, CA, USA, May 2016. IEEE.

[48] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective Attacks and Provable Defenses for Website Fingerprinting. In *24th USENIX Conference on Security Symposium*, San Diego, CA, USA, August 2014. USENIX Association.

[49] Tao Wang and Ian Goldberg. Improved Website Fingerprinting on Tor. In *12th Workshop on Privacy in the Electronic Society*, WPES, Berlin, Germany, November 2013. ACM.

[50] Tao Wang and Ian Goldberg. On Realistically Attacking Tor with Website Fingerprinting. *Proceedings on Privacy Enhancing Technologies*, 2016(4):21–36, 2016.

[51] Yixiao Xu, Tao Wang, Qi Li, Qingyuan Gong, Yang Chen, and Yong Jiang. A Multi-Tab Website Fingerprinting Attack. In *34th Annual Computer Security Applications Conference*, ACSAC, pages 327–341, San Juan, PR, USA, 2018. ACM.

[52] Qilei Yin, Zhuotao Liu, Qi Li, Tao Wang, Qian Wang, Chao Shen, and Yixiao Xu. An Automated Multi-Tab Website Fingerprinting Attack. *IEEE Transactions on Dependable and Secure Computing*, 19(6):3656–3670, November–December 2022.

[53] Yi Zhang, Miaomiao Li, Siwei Wang, Sisi Dai, Lei Luo, En Zhu, Huiying Xu, Xinzhong Zhu, Chaoyun Yao, and Haoran Zhou. Gaussian mixture model clustering with incomplete data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 17(1s):1–14, 2021.

[54] Zhongliu Zhuo, Yang Zhang, Zhi-li Zhang, Xiaosong Zhang, and Jingzhong Zhang. Website Fingerprinting Attack on Anonymity Networks Based on Profile Hidden Markov Model. *IEEE Transactions on Information Forensics and Security*, 3(5):1081–1095, 2018.

A Selection of Hyperparameters and Implementation of our MIL-based Method

We used Tensorflow[4] to implement our MIL-based method. Similar to [43,45], we relied on the extended candidate search method to identify the optimal hyperparameters for our model. For the hyperparameter tuning, we divided the dataset into training, validation, and testing with an 8:1:1 ratio using 10-fold CV and executed multiple experiments in a closed-world scenario utilizing the validation accuracy (i.e., the probability of a correct prediction in the validation set) as the performance metric. Table 5 summarizes the search ranges for the different hyperparamers and the values selected for them.

[4] https://www.tensorflow.org/

Table 5: Hyperparameter selection for our MIL-based method.

Hyperparameters	Search Range	Selected Value
Dimension of input traces	[5000]	5000
Optimizer	[Adam, Adamax, SGD]	Adamax
Training epochs	[10, ..., 70]	60
Learning rate	[0.001, ..., 0.01]	[0.002, 0.003]
Weight decay	[0.0001, ..., 0.01]	0.0001
Batch size	[100, ..., 300]	128
Activation Functions	[Tanh, ReLU, ELU]	Tanh, ELU, ReLU
Dropout	[0, ..., 0.5]	[0.1, 0.5]
Regularizer	[L1, L2, L1L2]	L2

Both the feature generator and the weights learner were trained using a weight decay of 0.0001, a batch size of 128, and initial learning rates of 0.002 and 0.003, respectively. We further identified the use of up to 60 training epochs as a good trade-off between classification accuracy and training time. For the convolutional layers in our feature generator we used 32, 64, 128, and 256 filters, respectively, and applied a dropout rate of 0.1. For our weights learner, we utilized a dropout rate of 0.5 and the regularizer L2. Finally, we used the Adam optimizer to minimize the loss function in our experiments.

B Creating Sitemap Graphs and User Sessions

For each website in `ALEXA-WSC-HMM`, we create a sitemap graph to collect automatically-generated user sessions.

Generating Sitemap Graphs. We create a sitemap graph per website containing data about available pages with the link relationship between them. Although some websites offer a hierarchical overview documents of their pages, these documents do not always provide data on page linkability and, thus, cannot be used to create our sitemap graphs. In response, we use a different strategy to collect the sitemap graphs. First, for each website, we gather the URLs of its index page and four additional, popular pages of it that were found from Google, i.e., to simulate that users access a website not only through its index page but also through an already known link, using a bookmark or by querying a search engine [32]. Starting from one of these five pages, we then extract all URLs from that page referring to the same website. We group the collected URLs based on their position on the page, i.e., whether they are located in the navigation section or in the footer, and exclude groups of URLs that are typically less visited by users, e.g., privacy policy and legal notice pages. From the remaining groups, we randomly select ten groups of URLs, fetch one random URL from each of these groups to simulate a user click, and repeat the procedure described above to decide on the next click. The crawling of URLs terminates once we reach a depth of ten pages for each website and have gathered at least 2000 unique pages. Finally, we build a directed graph where each node represents a URL and an edge between two nodes corresponds to a link between these URLs. For this graph, we consider all seen URLs regardless whether they were selected by the sampling for further steps or not.

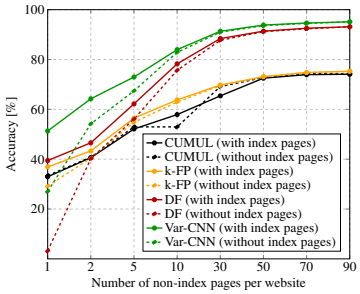
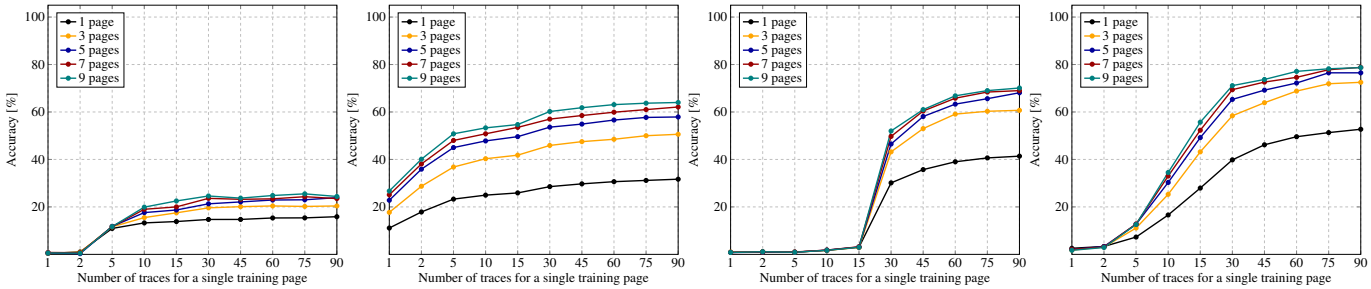


Figure 9: Accuracy achieved by the webpage classifiers with and without the use of index pages for training.

Generating User Sessions. Although a stored browser history would be a reliable source of real user sessions, it can reveal confidential data about users and usually is kept private. Thus, we use the gathered sitemap graphs to synthetically create a set of user sessions while ensuring that they exhibit realistic characteristics, as described by Kumar et al. [32]. As users can access a website in different ways, we use either the index page or one of the four additional pages of a site to start a user session. As Miller et al. [36], we execute a random walk over the sitemap graph of that website to sample the rest of the user session, whereas we prefer pages that have been visited neither in the current nor in any previously generated session. The latter increases the diversity between different sessions (and, thus, complicates the WFP attack).

C Use of Index Pages for Training

We evaluate whether the use of the index pages is particularly important to detect websites, as doubted in [39]. To achieve this, we conducted several experiments using our *website-with-unknown-pages* strategy in a closed world where each website was represented (i) by either 20 traces of its index page and multiple traces of other non-index pages belonging to the same website, (ii) or by its non-index pages only. In the first case, index pages are used solely for training and never for testing. Figure 9 illustrates the classification results obtained for a dataset containing 15 traffic traces per non-index page and a varying number of non-index pages, ranging from 1 to 90. As we can see, the difference in classification accuracy achieved when using index pages for training versus not using them is negligible for a set of 10 or more non-index pages for all classifiers. Although the difference is higher for smaller sets (i.e., less than 10) of non-index pages, particularly for both DL-based classifiers DF and Var-CNN, this is mainly due to the fact that the classifiers receive more training data through the use of index pages, which improves their accuracy. Similar results were obtained for different numbers of traces per non-index page (i.e., other than 15), which were omitted due to space constraints. Thus, we conclude that the use of index pages for training does not affect the accuracy of the



(a) CUMUL (b) k-FP (c) DF (d) Var-CNN

Figure 10: Accuracy achieved by our best voting strategy per classifier, when the number of traces of the training page increases.

attacks in either evaluation strategy and do not distinguish between using index and non-index pages in our evaluation.

D Additional Training Tactics

We also assess to what extent our voting strategies enhance the accuracy of webpage classifiers trained with a single page per website that is represented by an increasing number of traces. To accomplish this, we use the same test set as in Section 6.2.1, but adjusting the number of traces available per training page that represent a particular website. Figure 10 shows the obtained results. Unlike the results in Figure 5, where k-FP accuracy can increase by 40%, using a single page for training reduces this increase by about 30%. However, the opposite scenario is observed for DF. If the classifier is trained on a single page per website, probability voting achieves about 30% higher detection rate compared to the approximately 25% increase when using multiple pages per website. Nevertheless, the overall accuracy of all classifiers remains significantly lower than that of multiple training pages.

E Different Datasets & Tor Browser Versions

All evaluation results presented in Section 6 were obtained using a single set of websites and an old version of the Tor Browser. To verify that our results are generalizable to different sets of websites and newer Tor Browser versions, we repeated the experiments from Figure 4 using the `TRANCO-WSC-FG` dataset and computing only the best voting-based strategy for all webpage classifiers. Table 6 summarizes the results obtained. As we can see, the trend towards improved accuracy is evident with an increase in the number of sequentially visited pages per website. Overall, we achieve higher detection rates for both our voting-based methods and our MIL-based attack with all sets of pages when using the `TRANCO-WSC-FG` dataset in conjunction with the latest Tor Browser 12.5.3. Thus, we confirm the soundness of our findings.

Table 6: Accuracy (in %) of our fingerprinting methods for different datasets and Tor Browser versions.

Classifier	Set of pages								
`TRANCO-WSC-FG` & Tor Browser 12.5.3									
	1	2	3	4	5	6	7	8	9
Voting (with CUMUL)	71.76	85.20	91.50	93.70	95.10	96.00	95.80	96.40	96.80
Voting (with k-FP)	66.56	83.60	90.10	93.60	94.80	96.30	97.40	98.50	98.80
Voting (with DF)	87.36	98.00	99.40	99.60	99.80	99.90	99.90	99.90	100.00
Voting (with Var-CNN)	87.42	97.90	99.60	99.80	99.80	100.00	100.00	100.00	100.00
MIL-based	–	94.60	96.90	98.20	98.70	98.90	99.10	99.60	99.20
`ALEXA-WSC-FG` & Tor Browser 7.5.6									
Voting (with CUMUL)	56.14	69.70	79.10	82.70	85.50	87.00	89.30	91.00	91.70
Voting (with k-FP)	62.98	81.00	88.20	91.80	95.40	96.60	98.30	98.60	99.10
Voting (with DF)	70.57	88.90	94.60	96.60	98.30	99.20	99.50	99.70	99.90
Voting (with Var-CNN)	76.36	89.40	94.30	96.90	98.60	98.30	98.80	99.50	99.60
MIL-based	–	83.00	90.50	96.40	96.90	98.20	97.60	98.80	99.50