School of Informatics

Informatics Research Review
A Survey of Machine Learning Based Website Fingerprinting Attacks and Defenses

[redacted]

January 2021

Abstract

A website fingerprinting attack allows an adversary to detect which websites a user is visiting by analysing the network traffic. Website fingerprinting is inherently a classification task which makes it particularly well suited to apply machine learning methods to. In this survey we explore website fingerprinting attacks that employ machine learning techniques and how effective they are. We explore the defenses that have been developed to evade these attacks and their effectiveness. Finally we suggest future research directions to improve website fingerprinting attacks and defenses, and highlight other domains where website fingerprinting technologies could be applied.

Date: Friday 22nd January, 2021

Supervisor: [redacted]

1 Introduction

The ability to use services on the internet privately is something that every internet user relies on, whether it is to prevent companies and governments from tracking your activity, connect to a corporate network to work from home, to circumvent censorship, or send private messages. Privacy on the internet is achieved by encrypting the network communications sent between devices, and many protocols and services have been designed and built to provide this. An adversary may be able to intercept the network traffic and obtain the encrypted data, but without knowledge of the encryption keys they would be unable to decipher it. Anonymous browsing networks such as ToR (The Onion Router) [1] allows users to surf the internet while hiding their activities from a local observer with the capability to intercept traffic at any point in the network. As of January 2021 there are over 2 million active ToR users every day[1]. As a result of the Covid-19 pandemic there has been a large increase in the number of people working from home and connecting to their corporate infrastructure using virtual private networks (VPNs). Given our increasing dependence on the integrity of these services it is important to discover and fix any weaknesses to prevent adversaries getting hold of private information.

Website fingerprinting is a type of network traffic analysis attack that allows an adversary to determine which websites the victim has visited even if the network communications are encrypted. It's a classification problem: given an encrypted network trace determine which web site was visited. One of the early papers on website fingerprinting, published in 1997, used file sizes as a feature to identify which file was accessed over an encrypted connection [2]. Since then network encryption protocols have become more advanced at obfuscating meta information about the data being transmitted, but at the same time machine learning has transformed the ability for machines to accurately solve classification tasks, making it a good candidate to apply to website fingerprinting and starting a battle between increasingly sophisticated attacks and defenses.

In this paper we survey the research that uses machine learning methods to perform website fingerprinting attacks. To do this we will attempt to answer the following questions:

1. What is a website fingerprinting attack?
2. How are machine learning techniques being used to conduct website fingerprinting attacks and how effective are they?
3. What defenses are there against website fingerprinting attacks and how effective are they against machine learning based attacks?
4. What are the future research directions for using machine learning in fingerprinting attacks?

In Section 2 we review background material required to understand the website fingerprinting literature. In Section 3 we review website fingerprinting attacks that use machine learning and deep learning (a subfield of machine learning) techniques, followed by a review of defenses in Section 4. Finally in Section 5 we explore further research directions followed by the summary and conclusion in Section 6.

[1] See https://metrics.torproject.org/

2 Background

Researchers consider various setups when evaluating website fingerprinting attacks. Figure 1 shows a breakdown of the different setups and their relationships.

When constructing a machine learning model to perform a website fingerprinting attack the adversary must decide which websites they would like to detect, called the *monitor sites* or the *foreground set*. The complement of the foreground set, i.e. set of websites that are not included in the adversaries model, are called the *background set*. To evaluate the performance of a website fingerprinting attack there are two scenarios that are considered, the *closed world* and the *open world*. In the closed world scenario it is assumed the victim only navigates to websites in the foreground set, whereas in the open world scenario the victim may visit any website including those not known to the attacker. The closed world scenario is unrealistic, as the attacker is unlikely to know in advance all the possible websites the victim will visit, but it is a useful simplification to compare the performance of different attack models. When evaluating whether a given website fingerprinting attack is effective we will pay more attention to the open world performance, as this best describes the scenario in which an attack would be deployed.

The dataset for training a website fingerprinting classifier consists of dumps (or traces) of network traffic, typically obtained by setting up a web browser and navigating to the target web page. The websites used to compile the dataset are usually taken from the list of Alexa top websites[2] which compiles a list of the most popular websites. However web browsers typically cache data to speed up load times, which produces different network traces. To get reproducible results some authors clear the browser cache before each web page is loaded, however this may not be realistic except in cases such as the ToR browser which is non caching by default [3]. Another strategy is to load a web page multiple times, obtaining multiple different traces for the same page, and training the classifier on all of them. This makes the classifier more robust, but also inflates the size of the training set.

There are two main approaches to design the features for the website fingerprinting model. Crafting manual features using packet size, direction, or other statistics requires understanding the details of network protocols and are therefore more challenging to create. Automatic feature selection uses deep learning techniques to extract features from the raw dataset. Although this simplifies feature engineering it often requires a lot more data to train the model to a high accuracy. Attacks using traditional machine learning methods are reviewed in Section 3.1 and attacks using deep learning are review in Section 3.2.

There is an important distinction between the terms *website* and *web page*, best defined by Panchenko et al. [4]: "A website is a collection of web pages, which are typically served from a single web domain". Unfortunately the literature is sloppy with this definition and a 'website fingerprinting attack' is typically only able to detect when a user visits the *home page* (or *index page*) of a website, rather than any web page in the website. In most of the literature the terms *website* and *web page* (referring to the index page) are synonymous, and we will adopt this convention in this review.

[2] https://www.alexa.com/topsites

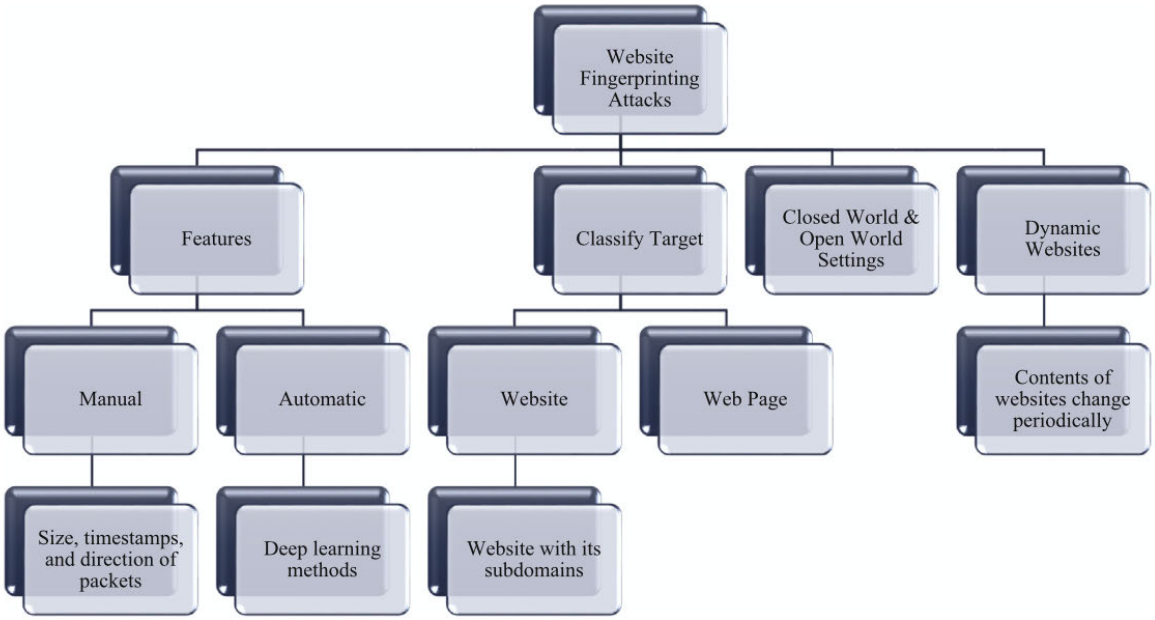

Figure 1: Overview of setups in website fingerprinting attacks. [5]

3 Website Fingerprinting Attacks

3.1 Attacks Using Machine Learning

One of the first website fingerprinting attacks to use machine learning classifiers was developed by Hermann et al. [6]. A Naive Bayes text mining classifier was trained on dumps of network communications using the frequency distribution of IP packet sizes as a feature to identify websites. The model was evaluated in the closed world scenario on a dataset of 775 website network traces obtained using several encryption protocols: CiscoVPN, OpenSSH, Stunnel, OpenVPN, JonDonym, and ToR. The model achieved 95% accuracy on the first four systems, but performed poorly on JonDonym and ToR achieving only 20% and 3% accuracy respectively. The reasons for the poorer performance was because JonDonym sends IP packets in a small range of fixed sizes, and ToR sends packets in a fixed size of 512 bytes. This reduces the noise in IP traffic and makes it difficult for the classifier to distinguish between different websites. Although the accuracies for the other systems look impressive, that they were evaluated in the closed world setting with a small number websites is not a realistic scenario in which an adversary would deploy such a system to fingerprint internet traffic. The authors consider an open world scenario where 78 of the 775 sites are in the monitor set, but only test using OpenSSH, achieving a 1.4% false positive rate, but a drop to 40% accuracy. As a first attempt the paper showed that website fingerprinting using machine classifiers is a viable possibility but that several challenges remained to develop a system that would work in a realistic open world scenario.

Since Hermann et al., there have been several improvements to make more accurate and robust classifiers. The development of improved machine learning methods resulted in applying them to website fingerprinting attacks too. Panchenko et al. used Support Vector Machines (SVM) to train their model using more sophisticated features including the volume, time and direction of IP packets [7]. When evaluated in the closed world scenario on the same dataset of 775

websites used by Hermann the classifier achieved an increase in accuracy on JonDonym from 20% to 80% and on ToR from 3% to 55%, a significant increase in performance. The model was then evaluated in the open world scenario on a much larger dataset consisting of subsets of 5000 monitor sites taken from the list of top 1,000,000 Alexa websites, obtaining a true positive rate of up to 73% and a false positive rate of 0.05%, a substantial improvement over previous classifiers. The success of this model is largely down to the authors attention to detail to devise more sophisticated features that aided classification, rather than just throwing the raw dataset at the classifier. Particular attention was also paid to minimize the false positives in the open world evaluation. This is important as traffic that is incorrectly classified as one of the monitor sites reduces it's effectiveness to the attacker. Despite being evaluated on a large open world dataset, the authors do not describe how an adversary could use such a model in a real world situation, for instance to monitor ToR traffic.

Continuing with the theme of applying improved machine learning methods, Wang et al. developed an attack using a k-Nearest Neighbors (k-NN) classifier [8]. Several features are used to fingerprint network traffic including measuring lengths, ordering, concentration and bursts of IP packets. The k-NN classifier works by identifying whether the feature vector derived from the sample network traces are close to any of the feature vectors from the training data. If the sample is far away from the training data then it can be classified as an unknown data point. This suggests it will perform well in open world scenarios where observed websites may be unknown, that is the classifier wasn't trained on them. In the closed world setting of 100 websites the model achieved 91% accuracy and in the open world scenario achieved a true positive rate of 86% and a false positive rate of 0.6%. While the true positive rate beats Panchenko's SVM classifier the false positive rate is an order of magnitude higher and the authors failed to acknowledge the importance of this to the effectiveness of the attack, explore the reasons for it, or adjust their model to compensate for it. In addition the sample sizes of websites in both the closed and open world evaluation are much smaller compared to Panchenko throwing doubt on how performant the model would be in a real world setup.

To improve the k-Nearest Neighbors approach Hayes and Denezis combined two machine learning methods [3]. They used a Random Forests classifier to generate fingerprints of websites given a set of raw input features based on the number, concentration, order, and arrival time of packets. The generated fingerprints are then fed into a k-Nearest Neighbors classifier which is used to determine if a given fingerprint matches one from the training set. The authors hypothesize that the features generated by the Random Forests classifier are better than the raw input features, and named their approach 'k-fingerprinting'. The classifier was trained on two datasets of website traces, one obtained using the chrome web browser and the other using the ToR browser. For comparison the model was evaluated on the Wang dataset in [8]. In the closed world setting the classifier achieved 91% accuracy and in the open world setting achieved 88% true positive rate and 0.5% false positive rate achieving a minor improvement over Wang's k-NN classifier.

Panchenko et al. also attempted to improve the k-Nearest Neighbors approach using Support Vector Machines and features computed from the cumulative sum of packet lengths, naming their method CUMUL [4]. In the closed world setting the attack achieved 91% accuracy. To evaluate the open world scenario the authors consider two scenarios: multi-class and two-class, where multi-class considers each monitor page as a class in the dataset, and two-class considers each website in the monitor set as one class. Using the dataset from [8] in the multi-class case the model had a 96% true positive rate and 9.61% false positive rate, and in the two class case had a true positive rate of 96% and a false positive rate of 1.9%. This demonstrated that combining all the monitor pages into a single class drastically reduces false positives making

the attack more effective.

3.2 Attacks Using Deep Learning

Rimmer et al. were one of the first to apply deep learning techniques to website fingerprinting attacks using three deep neural networks: Stacked Denoising Autoencoders (SDAE), Convolutional Neural Networks (CNN), and a recurrent Long Short Term Memory network (LSTM) [9]. One of the key benefits of deep learning verses traditional machine learning techniques is that the model can automatically determine features for classifying the training inputs. However in order to train the networks effectively a large training set is required. The data sets used in previous papers were too small, so the researchers compiled a large dataset consisting of 800,000 network traffic traces which the authors claim is a 4-fold increase over datasets used in previous research. In contrast to other papers, the authors optimized their models to reduce the true positive rate (TPR) and false positive rate (FPR). The models were compared in the open world scenario against CUMUL [4]. When optimized for FPR, which as noted before is of greater importance to the attacker, SDAE performed the best beating CUMUL with a TPR of 71.3% and FPR of 3.4%. Although the SDAE classifier had the best performance the CNN model was much faster to train with comparable performance, however it also had a tendency to overfit leading to reduced accuracy in the open world setting. The range of techniques evaluated in this paper together with an evaluation over a huge open world dataset demonstrates the viability of deep learning to website fingerprinting attacks. However the authors do not evaluate their attack against any of the website fingerprinting defenses published in the literature (see Section 4), which is important to determine whether these attacks could work in real world scenarios.

Sirinam et al. developed a deep learning model based on CNNs called Deep Fingerprinting (DF). The authors used a more sophisticated convolutional neural network architecture with the aim of developing a classifier that performed better than the one developed by Rimmer. The model used a different formulation of convolutional block found in modern CNN architectures such as VGG and ResNet combined with the ELU activation function rather than ReLU. In the closed world evaluation consisting of a dataset of 1000 traces for each of 95 websites, DF achieved 98% accuracy compared to 92% for the SDAE Rimmer model. In the open world evaluation the authors used a dataset consisting of 40,716 web traces from 5000 websites taken from the Alexa top 50,000. Rather than performing a single open world experiment, as is typical in other papers, the size of the unmonitored training set was varied to see what effect this has on the TPR and FPR and compared it with values obtained from five other state of the art WF approaches. The results showed that DF outperformed all other state of the art approaches with a 96% TPR and 0.7% FPR. In addition, the performance of DF was evaluated against the best known website fingerprinting defenses in both closed and open world scenarios including BuFLO [10], Tamaraw [11], WTF-PAD [12], and Walkie Talkie [13]. The results showed that DF was able to maintain over 90% accuracy on traffic protected with WTF-PAD. Walkie-Talkie was the only effective defense against DF, but the authors argue that Walkie Talkie is not yet practical to be deployed in a real world scenario due to the bandwidth and latency overhead. Ironically the authors didn't discuss the practicalities of implementing DF in a real world scenario, which is one of the key areas missing in an otherwise comprehensive evaluation of their attack.

One major limitation of website fingerprinting attacks that hadn't been addressed was the short time effectiveness of the classification models. If the content of a webpage changes this will alter the network trace, and hence a classifier trained on older versions of a webpage will fail to classify new new versions. This is particularly true for news and social websites whose content changes frequently. This phenomena is referred to in the literature as *concept drift*.

Training deep learning models is particularly expensive as it requires a large amount of data to achieve good accuracy. To address this Attarian et al. developed a website fingerprinting attack that leverages adaptive streaming algorithms to build a classifier that can be updated with new network traces in real time called AdaWFPA (Adaptive Website Fingerprinting Attack) [5]. The authors compared the performance of their model using the Wang dataset [8] with state of the art classifiers including Wang's k-NN and DF. AdaWFPA performed the best in both closed world scenarios with accuracies over 99% compared to around 90% for k-NN and 98% for DF. In the open world scenario AdaWFPA also outperformed the other classifers with accuracies over 99% and precision and recall over 99%. For comparison k-NN had an accuracy, precision and recall around 90% and DF had an accuracy of 98%, precision of 99% and recall of 94%. The authors also compare their model against the Walkie-Talkie defense [13] and demonstrate that their model can defeat it with an accuracy of over 99% in both open and closed world scenarios.

4 Website Fingerprinting defenses

As website fingerprinting attacks become more accurate and consequently more feasible several defenses were created to thwart them. Dyer et al. attempted to create a 'best case defense' called Buffered Fixed-Length Obfuscation or 'BuFLO' [10]. The defense works by sending dummy packets of a fixed size at constant intervals to pollute the network trace, making it harder for a classifier to distinguish between different websites. However the approach is very inefficient with a significant bandwidth overhead varying from 100% to over 400%. Many encrypted communications protocols such as ToR are low latency and thus the additional overhead would severely impact performance. The authors evaluate the defense in a closed world scenario with a dataset of 128 website traces and conclude from the results of their experiments that BuFLO is unable to prevent website fingerprinting attacks completely as some information that could be used for classification is always leaked depending on the precise configuration of BuFLO used. They then conclude that all countermeasures against traffic analysis attacks will be inadequate, which given the BuFLO was only evaluated in a small closed world scenario, not in any open world scenarios, is far to bold a claim to make. Nevertheless BuFLO was more effective than all the other known defenses at the time and provided a new target for those developing attack models to try and circumvent.

In two papers Cai et al. attempted to address the bandwidth issues of BuFLO by developing defenses called Tamaraw [11] and CS-BuFLO [14]. Tamaraw works by grouping websites by similar size of transmitted data together into 'anonymity sets', and then padding them to the largest size of the group. The authors also suggested using different packet sizes and padding at different rates for both incoming and outgoing traffic to mimic the asymmetry in normal web traffic. CS-BuFLO enhances Tamaraw to add congestion sensitivity and rate adaptation by padding each group up to a power of two or a multiple of the transmitted application data. The approaches were evaluated in both open and closed world scenarios with comparable performance to BuFLO in terms of reducing the accuracy of website fingerprinting attacks, but incurred a bandwidth overhead of over 130%, which while much better is still impractical in a real world deployment.

To improve on the BuFLO family of attacks (BuFLO, Tamaraw and CS-BuFLO), Juarez et al. developed a defense using adaptive padding techniques called Website Traffic Fingerprinting Protection with Adaptive Defense (WTF-PAD) [12]. The defense works by interspersing network traces with 'dummy packets' to disrupt the network trace and make it harder to classify

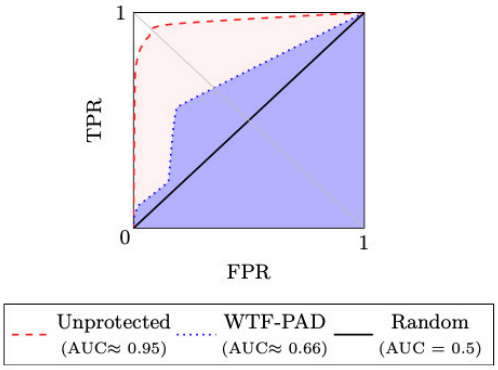

Figure 2: ROC curves of k-NN with and without WTF-PAD defense. Random guessing is also plotted for comparison [12]

a particular trace. Compared to BuFLO type defenses which delay packets to make them appear at constant rates, WTF-PAD does not delay packets, thereby not incurring a latency cost. WTF-PAD was evaluated in the closed world scenario against the k-NN model developed by Wang. The ROC curves for the classifier with and without WTF-PAD are displayed in Figure 2. Without any defense the classifier is almost perfect with AUC (area under curve) 0.95, but with WTF-PAD the AUC is much reduced to 0.66 which is much closer to random guessing. In the open world scenario consisting of a dataset of 5000 websites the defense was also effective, reducing the AUC from 0.79 to 0.27. The bandwidth overhead when using WTF-PAD was under 60% which is much lower than BuFLO attacks, but is still a significant drop in network performance.

Finally we review a novel state of the art defense against website fingerprinting attacks by Wang and Goldberg called Walkie Talkie [13]. The idea is to communicate over the network in half-duplex mode, that is the web browser waits for all responses from the server to be received before sending out the next batch of web requests. Thus traffic only passes between the client and server in one direction at any one time. Once packets are grouped together in the direction of travel Walkie Talkie then inserts dummy packets and delays in a process called 'burst molding' to make sequences of sensitive packets look identical to sequences of non-sensitive ones. By performing communication in half duplex mode, burst molding can be achieved with lower bandwidth and latency than communication in full duplex mode. The bandwidth overhead of Walkie Talkie is 31% with a latency overhead of 34%, much lower than WTF-PAD and BuFLO type defenses. Walkie Talkie was evaluated in the closed world scenario with a dataset of the top 100 Alexa websites against 10 attacks and in each case significantly reduced their accuracy down to less than 50%. In the open world scenario Walkie Talkie was evaluated against the Wang k-NN [8] and CUMUL [4] attacks with a reduction in TPR to 0.68 and 0.2 respectively, and an increase in FPR to 0.62 and 0.35 respectively, demonstrating that the defense is effective against these attacks. However, results by Attarian et al. indicate that their attack AdaWFPA is able to evade Walkie Talkie obtaining an accuracy in the open world setting of over 99% [5]. This suggests that new defenses are required to prevent the state of the art fingerprinting attacks.

5 Further Research

5.1 Improving Website Fingerprinting Attacks

In spite of the advances made with website fingerprinting attacks many limitations still exist that prevent them from being effective in real world scenarios. Juarez et al. compiled a list of assumptions that website fingerprinting papers frequently make [15] including:

- Users browse the web one page at a time and only have a single browser tab open, i.e. no multi-tab browsing.
- The adversary is able to filter out background network traffic produced by other applications.
- The adversary can train their classifier under the same network conditions as the victim. This is particularly important for websites that serve different content depending on the type of device (desktop or mobile) or locality (which could determine the language the web pages are loaded in).

Few papers have considered any of these assumptions, and none in this survey considered all of them. Guaranteeing all of these assumptions is unrealistic, particularly the assumption of effectively filtering background traffic, so it's an important direction for further research to determine whether a classifier can be created that can work around them.

Another important limitation with website fingerprinting methods is that the models are only trained on the home pages of the monitored websites. Depending on the adversaries targets this could be a reasonable assumption, but in general a website fingerprinting attack would also need to cope with sub pages of websites too. A future research direction would be to investigate how feasible it is to use existing attacks to fingerprint websites from sub pages and how effective they are.

A related issue is *concept drift* whereby the content of a website changes over time creating different network traces that the classifiers need to learn. The only paper to address this issue is Attarian et al. who developed AdaWFPA which employed adaptive streaming algorithms to update the model, but the paper is the first to employ this idea and it would be valuable to see if other researchers can improve the performance of the attack even further and evaluate it on other secure communications protocols, such as Wireguard, OpenVPN, and SSH.

Another issue making it difficult to compare the effectiveness of attacks and defenses is the variety of datasets used. Most papers compile their own datasets to evaluate their models, varying in size from 128 websites to hundreds of thousands. Some authors such as Wang et al. publish their datasets which are later used by others to compare performance [8]. However the community would benefit from having a large public dataset of website traces on which researchers can evaluate and compare performance of their attacks and defenses, in the same way that the ImageNet[3] dataset is used by computer vision researches to benchmark and compare their machine learning approaches.

But perhaps the most important question not addressed by any of the papers is whether a website fingerprinting attack could succeed in a real scenario and exactly what an adversary is able to accomplish. To our knowledge there is no published work on devising a detailed threat model and attempting to deploy an attack in a real world setting, such as on ToR or other secure networking protocols.

[3] http://www.image-net.org/

5.2 Improving Website Fingerprinting Defenses

While most publications on website fingerprinting focus on attacks, there is a growing body of research on defenses. Although WTF-PAD and Walkie Talkie provide effective defenses against most attacks they come with a cost of high bandwidth and latency. Additionally Attarian et al. demonstrated that their attack AdaWFPA was able to evade Walkie Talkie with an accuracy in the open world setting of over 99% [5]. If this result is reproducible then this means a new website fingerprinting defense is required that is capable of mitigating the state of the art attacks.

Developing lightweight defenses will also be crucial if fingerprinting attacks become practical for real world deployment. In the case of attacks using deep learning Sirinam et al. suggested crafting adversarial examples to trick a fingerprinter into misclassifying the inputs [16], which given the recent interest in adversarial machine learning it would be a worthwhile line of research to see if such techniques could be used as part of a website fingerprinting defense.

5.3 Applications To Other Domains

The success of website fingerprinting attacks have led to the techniques being applied to other domains too. Encryption of HTTP traffic has been around since the early 2000s but encryption of DNS (Domain Name System) traffic has only recently started to become mainstream with two competing protocols, DNS over HTTPS and DNS over TLS. These protocols are also vulnerable to fingerprinting attacks and the current state of the art was developed by Troncoso et al. who used a Random Forests classifier to obtain classification accuracies above 90% [17]. However to our knowledge deep learning techniques have not been tried on encrypted DNS, and while there are some defenses developed to evade fingerprinting attacks such as *EDNS(0)* padding, they are not at the same level of sophistication or effectiveness as WTF-PAD or Walkie Talkie [17].

Another area of application is fingerprinting smartphone apps using traffic analysis of app network communications. Taylor et al. used machine learning techniques to identify 110 of the most popular apps in the Google Play Store and were able to identify them six months later with up to 96% accuracy [18]. Such attacks allow an adversary to determine which apps a user has installed on their phone, potentially leaking sensitive information about the user, for instance who they bank with, their hobbies, which social networks they use, and leverage that information to launch targeted attacks on that user, for instance spear phishing attacks. It would be a valuable line of research to see if any of the website fingerprinting defenses such as Walkie Talkie are effective at evading such attacks.

6 Summary & Conclusion

In this literature review we have traced the development of machine learning based website fingerprinting attacks and defenses, exploring how effective they are and identified lines of further research.

Website fingerprinting attacks have come a long way from the Naive Bayes classifier developed by Hermann et al. with the state of the art methods employing technologies such as deep learning for automatic feature detection, and adaptive streaming algorithms to automatically update the classification model. However, evaluating the effectiveness of these attacks is challenging and there is still a debate over the best way to evaluate and compare them. Currently researchers evaluate their attacks in two scenarios: 'closed world' where all the websites navigated by the

victim are known to the attacker, and 'open world' where the victim may navigate to sites unknown to the attacker. The open world scenario is more realistic, and while early attacks struggled to get above 50% accuracy the state of the art classifiers are able to attain above 90%. Although these figure sound impressive there are still many assumptions in the open world evaluations, such as the ability for the attacker to filter out background network traffic to obtain a 'clean' network trace that only includes packets from the browser navigating to a single website. Thus while website fingerprinting attacks perform well under laboratory conditions there are many barriers that make them impractical for use in real world situations.

Defenses against website fingerprinting attacks have also matured. The early attempts such as BuFLO were simple to implement but incurred high bandwidth and latency costs of over 400% [10]. Subsequent defenses optimized this with the state of the art 'Walkie Talkie' bringing the bandwidth overhead down to around 30% [13], which while more practical is still a significant reduction in performance that would adversely affect services such as ToR which are already low bandwidth. Exacerbating the situation further, researchers have gradually been able to develop smarter attacks that can evade these defenses such as AdaWFPA which was able to correctly classify websites defended with Walkie Talkie with an accuracy over 99% [5], indicating that research is required to develop new defenses against these state of the art attacks.

Finally we explored further lines of research, perhaps the most valuable being a demonstration of a website fingerprinting attack in a real world scenario together with an analysis of what information the attacker was able to obtain, and whether any countermeasures could have prevented it. The technologies developed for website fingerprinting attacks and defenses have also been applied to other domains including DNS fingerprinting and smartphone app identification, but as far as we can determine, attacks that use deep learning have yet to be applied to these areas, offering a promising line of further research.

References

[1] Paul Syverson, Roger Dingledine, and Nick Mathewson. Tor: The secondgeneration onion router. In *Usenix Security*, pages 303–320, 2004.

[2] Heyning Cheng, , Heyning Cheng, and Ron Avnur. Traffic analysis of ssl encrypted web browsing, 1998.

[3] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1187–1203, Austin, TX, August 2016. USENIX Association.

[4] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, Jan Pennekamp, Klaus Wehrle, and Thomas Engel. Website fingerprinting at internet scale. *Ndss*, 2016.

[5] Reyhane Attarian, Lida Abdi, and Sattar Hashemi. Adawfpa: Adaptive online website fingerprinting attack for tor anonymous network: A stream-wise paradigm. *Computer Communications*, 148:74 – 85, 2019.

[6] Dominik Herrmann, Rolf Wendolsky, and Hannes Federrath. Website fingerprinting: attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, pages 31–42, 2009.

[7] Andriy Panchenko, Lukas Niessen, Andreas Zinnen, and Thomas Engel. Website fingerprinting in onion routing based anonymization networks. In *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, pages 103–114, 2011.

[8] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. Effective attacks and provable defenses for website fingerprinting. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 143–157, 2014.

[9] Vera Rimmer, Davy Preuveneers, Marc Juarez, Tom Van Goethem, and Wouter Joosen. Automated website fingerprinting through deep learning. *Proceedings 2018 Network and Distributed System Security Symposium*, 2018.

[10] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. Peek-a-boo, i still see you: Why efficient traffic analysis countermeasures fail. In *2012 IEEE Symposium on Security and Privacy*, pages 332–346, 2012.

[11] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. A systematic approach to developing and evaluating website fingerprinting defenses. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 227–238, 2014.

[12] Marc Juarez, Mohsen Imani, Mike Perry, Claudia Diaz, and Matthew Wright. Toward an efficient website fingerprinting defense. In Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows, editors, *Computer Security – ESORICS 2016*, pages 27–46, Cham, 2016. Springer International Publishing.

[13] Tao Wang and Ian Goldberg. Walkie-talkie: An efficient defense against passive website fingerprinting attacks. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1375–1390, Vancouver, BC, August 2017. USENIX Association.

[14] Xiang Cai, Rishab Nithyanand, and Rob Johnson. Cs-buflo: A congestion sensitive website fingerprinting defense. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 121–130, 2014.

[15] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. A critical evaluation of website fingerprinting attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 263–274, New York, NY, USA, 2014. Association for Computing Machinery.

[16] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 1928–1943, New York, NY, USA, 2018. Association for Computing Machinery.

[17] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. Encrypted dns - privacy? a traffic analysis perspective, 2020.

[18] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic. Robust smartphone app identification via encrypted network traffic analysis. *IEEE Transactions on Information Forensics and Security*, 13(1):63–78, 2018.