IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 February 2023, accepted 28 February 2023, date of publication 6 March 2023, date of current version 17 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3253559

SURVEY

A Survey on Deep Learning for Website Fingerprinting Attacks and Defenses

PEIDONG LIU[1], LONGTAO HE[2], AND ZHOUJUN LI[1,3]
[1]State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China
[2]National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
[3]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding authors: Zhoujun Li (lizj@buaa.edu.cn), Longtao He (hlt@cert.org.cn), and Peidong Liu (pdliu@buaa.edu.cn)

This work was supported in part by the National Key Research and Development Plan Project under Grant 2021YFB3101400; in part by the National Natural Science Foundation of China under Grant 62276017, Grant U1636211, and Grant 61672081; in part by the Fund of the Key Laboratory of Power Grid Automation of China Southern Power Grid Company Ltd. under Grant GDDKY2021KF03; and in part by the Fund of the State Key Laboratory of Software Development Environment under Grant SKLSDE-2021ZX-18.

ABSTRACT The attacks and defenses on the information of which website pages are visited by users are important research subjects in the field of privacy enhancing technologies, they are termed as website fingerprinting (WF) attacks and defenses. Nowadays, deep learning is an important tool in many research areas, including WF attacks and defenses. In this paper, we offer a comprehensive survey on deep learning for WF attacks and defenses. After a brief introduction, we first summarize deep learning, WF attacks, and WF defenses. For deep learning, we review the common paradigms, architectures, and performance metrics. For WF attacks, we review the approaches, challenges and solutions. The approaches include deep learning, traditional machine learning, and other methods. Challenges and solutions cover multi-tab browsing, concept drift, and the base rate fallacy. For WF defenses, we review the strategies and approaches. Then, we survey deep learning for WF attacks, and deep learning for WF defenses. In deep learning for WF attacks, we survey in detail the deep learning paradigms, architectures of WF attack models, and the performance of several representative WF attack models, and look into the future. In deep learning for WF defenses, we survey the architecture, efficacy and overhead of deep learning models in WF defenses, and look into the future. In the end, we summarize this paper.

INDEX TERMS Deep learning, website fingerprinting, WF attack, WF defense.

I. INTRODUCTION

Internet users and web service providers are employing more and more privacy enhancing technologies (PETs) for security reasons and privacy concerns. Among the various technologies, The Onion Router (Tor) [1] provides anonymous communication between users and hidden service providers through layered encryption, and the privacy protection it provides is regarded as outstanding. However, even Tor still suffers from traffic analysis attacks like website fingerprinting (WF), which can be carried out to reveal the website pages visited by the user, thus compromising her privacy. WF adopts deep learning, traditional machine learning or other methods to achieve its goals. Powered by newly developed neural network architectures and emerging learning paradigms, deep learning achieves impressive results on WF tasks.

WF attacks pose a great threat to those people who want to hide their web activities. Inevitably, great efforts are put into developing defensive measures against these attacks. These measures alter contents and behavior of the communication to mislead the attackers. Deep learning and traditional machine learning methods are vulnerable to adversarial examples, which incorporate small perturbations enough to change the expected results. Applying adversarial perturbations to traffic instances is a practical approach to WF defenses [2]. Adversarial examples can be generated through deep learning or other methods.

Deep learning emerged and achieved stunning results on ImageNet classification tasks [3], and had since become a

The associate editor coordinating the review of this manuscript and approving it for publication was Shu Xiao.

powerful tool widely adopted in various domains. The ability to learn representations of data through multiple levels of abstraction lays foundation for the success of deep learning [4]. As mentioned above, deep learning underpins most state-of-the-art WF attacks, and serves as an indispensable tool for some WF defenses.

This paper surveys deep learning for WF attacks and defenses. First, We briefly summarize deep learning, WF attacks and WF defenses. Then, we give detailed reviews on deep learning for WF attacks and defenses separately and look into the future. In the end, we summarize the survey.

II. DEEP LEARNING

Fueled by the advances in computing hardware, the deep learning community nowadays constantly comes up with new architectures. With the changes in the characteristics and scale of available data for training, new deep learning paradigms also emerge.

A. PARADIGMS

Common machine learning paradigms include supervised learning which requires labeled data only, semi-supervised learning which leverages both labeled and unlabeled data, unsupervised learning which requires unlabeled data only, and reinforcement learning which utilizes data generated via interactions of the agent with an environment.

Transfer learning deals with situations like where there is not enough training data in the target task/domain but sufficient data in the source task/domain which shares commonality with the target task/domain, it extracts knowledge from one or more source application scenarios to help improve the learning performance in a target scenario. Multi-task learning also exploits commonality across different tasks, but treats all tasks with equal importance [5].

Self-supervised learning takes unlabeled data as input, leverages the inherent co-occurrence relationships of the data as self-supervision, learns to predict or reconstruct masked or corrupted portions of the data [6], [7], [8]. Because there is no manual label involved, self-supervised learning can be viewed as a branch of unsupervised learning. While unsupervised learning focuses on discovering data patterns, self-supervised learning concentrates on recovering [7].

Pre-training acts as the first stage of transfer learning, extracts knowledge from one or more source tasks, and is often followed by fine-tuning as the second stage which transfers the obtained knowledge to target tasks. If the source tasks require labeled data only, then the pre-training is supervised. With supervised pre-training, the computer vision community benefits a lot from models like ResNet50 [9], explorations on language tasks with models like CoVe [10] also achieve promising results. If the source tasks require unlabeled data only, then the pre-training is unsupervised. If an unsupervised pre-training stage utilizes the training data itself to recover portions of the data, then it is self-supervised [6]. Self-supervised pre-training brings about great progress on language tasks with models like BERT [11] and GPT-3 [12], a new paradigm of ''pre-train, prompt, predict'' also arises, in which the downstream tasks are reformulated to look more like those solved during the pre-training stage with the help of a textual prompt, replacing the ''pre-train, fine-tune'' paradigm [13].

Metric learning [14] algorithms produce distance metrics that capture the important relationships among data. Meta-learning, known as learning to learn, is a process in which previous knowledge and experiences are used to guide the model in the learning of a new task. There are mainly three types of meta-learning methods: metric-based, model-based, and optimization-based [15].

Few-shot learning is proposed to mimic the capacity of human beings learning a novel concept with only a few examples or no examples. When no examples are needed, it is called zero-shot learning. Transfer learning can pre-train all previous experiences into a model for few-shot learning to fine-tune [5]. Meta-learning is also an effective way to solve the problem of few-shot learning [15].

Contrastive learning learns representations by contrasting positive pairs against negative pairs [16]. Self-supervised learning based on contrastive learning enables the CLIP model [17] to achieve impressive results at zero-shot transfer learning, with massive amounts of unlabeled language and vision data in the multi-modal fashion [18].

Other paradigms include federated learning, etc.

B. ARCHITECTURES

Transformers [19], recurrent neural networks (RNNs), convolutional neural networks (CNNs), generative adversarial networks (GANs) are among the most common deep learning architectures.

Currently, Transformer-based models [11], [20] outperform other neural networks on language and vision tasks, due to the global representations learned by entirely depending on attention mechanisms. RNNs perform well on tasks with sequential inputs, like speech and language. Conventional RNNs have difficulties in learning long-term dependencies because it is hard to learn to store information for long, while long short-term memory (LSTM) networks introduce special hidden units to remember inputs and are more effective. CNN layers are inspired by visual neuroscience architecture, and CNNs are suitable for vision tasks [4]. GANs are used to estimate generative models [21], the ideas of adversarial training adopted by GANs lead to successful applications in image synthesis, content creation, domain adaptation, domain or style transfer [8]. Siamese networks and triplet networks are commonly used in metric-based meta-learning.

Other deep learning architectures include graph neural networks (GNNs), variational auto-encoders (VAEs), etc.

C. PERFORMANCE METRICS

Common performance metrics for deep learning-based classifiers include [22], [23]: accuracy, error rate, precision, recall, F-score, true positive rate (TPR), false positive rate (FPR), etc. Accuracy refers to the proportion of examples

for which the model produces the correct output. Error rate refers to the proportion of examples for which the model produces an incorrect output. There are four types of all the results of classifiers: true positive (TP), true negative (TN), false positive (FP), false negative(FN). Precision is the ratio of true positives over the sum of true positives and false positives, i.e., TP / (TP + FP). Recall is the ratio of true positives over the sum of true positives and false negatives, i.e., TP / (TP+FN). F-score takes both precision and recall into the consideration, is their harmonic mean, i.e., 2 × precision × recall / (precision + recall). TPR is equal to recall. FPR is the ratio of false positives over the sum of true negatives and false positives, i.e., FP / (TN + FP).

III. WF ATTACKS

WF attacks aim to identify website pages from corresponding encrypted traffic or related data acquired through passively monitoring the communication process. The website pages to be identified can be the index pages (homepages) of different websites, or different webpages including index page and non-index pages (inner pages) of the same website, or some other combinations. The acquired traffic or other data may include the raw IP packets, the metadata of these packets (including size, direction, inter-arrival time, etc.), and side-channel information such as power consumption, etc.

Consider each target website page as a label, and the corresponding traffic or other data as an instance, WF can be viewed as a classification problem. When all the class labels are monitored webpages, the attack scenario is called closed world, in which the user never visits non-monitored webpages. When each monitored webpage is considered as a separate class and all the non-monitored webpages are considered as a single class, the attack scenario is called open world, in which the user may visit monitored or non-monitored pages. The closed-world setting is not realistic and commonly used for theoretical purposes like comparing classifiers, while the open-world setting is more realistic [24].

A. APPROACHES

The approaches to WF attacks include deep learning, traditional machine learning, and other methods.

Various deep learning architectures and paradigms have been explored in realizing WF attacks. Abe and Goto [25] proposed a new method for fingerprinting attacks on Tor anonymity using Stacked Denoising Autoencoder (SDAE). Rimmer et al. [26] proposed an automated feature learning-based WF (AWF) attack based on deep learning, they designed, tuned and evaluated SDAE, CNN and LSTM models to automatically learn traffic features for website recognition. Sirinam et al. [27] presented deep fingerprinting (DF), a new website fingerprinting attack against Tor that leveraged CNNs with a sophisticated architectural design. He et al. [28] proposed a WF attack which used two-layer GRU network to extract the time feature and the 50-layer ResNet to extract the spatial features of the website fingerprint. Oh et al. [29] studied the suitability of Multi-layer Perceptrons (MLPs) and CNNs as perceptron-fingerprinting (p-FP) classifiers in a wide range of settings including identifying top Alexa websites in the closed-world and open-world settings, open-world multi-class classification, search query (keyword) fingerprinting, Onion Service fingerprinting, TLS-encrypted WF, etc. Shusterman et al. [30], [31] designed and implemented the cache occupancy side-channel attack, they evaluated the use of CNN and LSTM for fingerprinting websites based on the cache activity traces collected while loaded by the browsers. Bhat et al. [32] proposed Var-CNN, a website fingerprinting attack that leveraged ResNets along with novel insights specific to packet sequence classification. Sirinam et al. [33] proposed a new WF attack called Triplet Fingerprinting (TF) that used triplet networks for N-shot learning (NSL). Rahman et al. [34] proposed the *Tik-Tok* attack, which used directional timing representation generated by simply multiplying the timestamp of each packet by its directional representation, for the DF classifier, and achieved modest accuracy improvement over direction-only information in several settings. Wang et al. [35] proposed a novel website fingerprinting attack based on a two-channel Temporal Convolutional Networks (2ch-TCN) model that extracted features from both the packet sequences and packet timing information. Dahanayaka et al. [36], [37] analyzed CNNs for WF attacks, and found that they focused mainly on transitions between uploads and downloads in trace fronts, exhibited few-shot learning capabilities, and outperformed RNNs due to their resilience to random shifts in data. Wang et al. [38] presented a cross-platform website fingerprinting (CPWF) attack based on Multi-Similarity Loss which was introduced by deep metric learning to guide the deep learning model to extract effective feature sets. Ramezani et al. [39] developed a multi-label classifier based on LSTM, that can predict the websites visited by a user in a certain period, the classifier utilized the server names appearing in chronological order in the TLSv1.2 and TLSv1.3 Client Hello packets as features. Oh et al. [40] introduced Generative Adversarial Networks for Data-Limited Fingerprinting (GANDaLF), a new deep-learning-based technique to perform WF attack on Tor traffic, GANDaLF worked with few training samples, used GAN to generate a large set of fake data that helped to train a deep neural network in distinguishing between classes of actual training data. Wang et al. [41] analyzed the application of CNN and LSTM models in WF attacks using side-channel data. Wang et al. [42] proposed a new method, named Adaptive Fingerprinting (AF), which can achieve high WF attack accuracy over few encrypted traffic by leveraging adversarial domain adaption, i.e., a domain adversarial network, to learn a DNN-based Feature Extractor over one or multiple source datasets by formulating a minimax game between a Feature Extractor and a Domain Discriminator. The learned Feature Extractor was extracted and attached with a traditional machine learning classifier (e.g., kNN) to carry out the classification over a target dataset. Shen et al. [43] proposed BurNet, a fine-grained WF method using CNNs which took unidirectional burst sequences as input, sophisticated architecture was designed to

improve classification accuracy and reduce time complexity in training. Guo et al. [44] proposed a deep nearest neighbor small-sample website fingerprinting (DNNF) attack, first deep local fingerprinting features of websites were extracted via CNN, then webpage prediction was carried out by the k-nearest neighbor (kNN) classifier. Lu et al. [45] proposed the Graph Attention Pooling Network for fine-grained website fingerprinting (GAP-WF), introduced the trace graph to describe the contextual relationship between flows in webpage loading, utilized the Graph Neural Networks (GNNs) to learn the intra-flow and inter-flow features. Dani et al. [46] proposed a cross-trace WF attack based on the DF model under the closed-world setting, which leveraged the semantic correlation of the content of webpages across traffic traces generated by the same user to improve attack accuracy when existing defenses were enabled, four semantic similarity evaluation methods were investigated, including TF-IDF with cosine similarity, BERT embeddings with cosine similarity, Word Movers Distance (WMD), and GLoVe embeddings with cosine similarity. Nasr et al. [2] performed adversarial training to increase the robustness of deep learning models like DF to adversarial examples, and used adversarial perturbations as a regularizer to train robust traffic analysis models. To generate a set of adversarial examples in the training process, they randomly chose a number of packets and flipped their directions from -1 to +1 and vice versa. Similarly, for the packet timings and sizes they enforced all of application constraints for generating adversarial examples. Zhang et al. [47] proposed Tripod, a novel data augmentation method for WF attacks, which applied three packet manipulations (Injecting, Removing, and Losing) on one collected traffic trace to generate several augmented traces, experimental results on ResNet-18, ResNet-34, VGG-16, VGG-19, DF, Var-CNN showed that Tripod had good universality because it had enhanced these six WF attacks and may work with more WF attacks. Chen et al. [48] introduced a model-agnostic, efficient, and harmonious data augmentation (HDA) method that can improve deep WF attacks significantly, the method augmented the original training data by rotating and masking out randomly individual samples and mixing (linearly combining) sample pairs in arbitrary proportions, experimental results showed that Var-CNN with HDA achieved the best results. Gulmezoglu [49] focused on explaining traditional machine learning and deep learning models in the context of microarchitecture-based website fingerprinting attacks, performance counters and cache occupancy side-channels were implemented on Google Chrome and Tor browsers, LIME and saliency map eXplainable Artificial Intelligence (XAI) methods were applied to examine the leakage points in the side-channel data after the models were trained. Guo et al. [50] studied and proposed a homology analysis-based few-shot WF attack, relying on a Convolutional Neural Network-Bidirectional Long Short-Term Memory (CNN-BiLSTM) model. Chen et al. [51] studied few-shot website fingerprinting attack where only a few training samples per website were available, introduced a novel Transfer Learning Fingerprinting Attack (TLFA) that can transfer knowledge from the labeled training data of websites disjoint and independent to the target websites, TLFA employed embedding CNN model in the pre-training stage, and explored multivariate logistic regression (LR), support vector machine (SVM) with linear kernel, multilayer perceptron (MLP) in the fine-tune stage. Sun et al. [52] proposed a WF attack to identify the websites visited by Tor users through frequency domain fingerprinting (FDF) of network traffic, they extracted the direction and length features of circuit sequences in access traffic, combined and transformd them into frequency domain data, and classifed the data using a model combining CNN, FC, and self-attention. Yanbin Wang et al. [53] presented snWF, a novel WF attack based on a simple and effective neural network snapshot ensemble, which used the newly designed CNN model as the base classifier, and can reduce the variance of neural networks and improve the robustness of the attack. Li et al. [54] proposed more robust DNN models for WF attacks using adversarial training. Yongxin Chen et al. [55] proposed a data augmentation method which can improve the performance of deep learning-based WF attacks using the generated bionic traces. Cherubin et al. [56] adapted TF attack to an online setting and trained the WF models on data safely collected on a Tor exit relay. Chen et al. [57] introduced a novel Meta-Bias Learning (MBL) method for few-shot WF attack.

Traditional machine learning methods have long been employed in WF attacks. Sun et al. [58] chose the number and length of objects requested as part of a webpage as the traffic trace signature, and Jaccard coefficient as the metric for measuring the similarity between two signatures, to fingerprint website pages. Bissias et al. [59] presented a WF attack which measured the similarity of webpages by computing the cross correlation of two sequences of values of the statistical characteristics of web requests from interesting sites, including distributions of packet sizes and inter-arrival times. Liberatore et al. [60] examined the effectiveness of two traffic analysis techniques for identifying encrypted HyperText Transfer Protocol (HTTP) streams, one based on the naïve Bayes classifier and one on the Jaccard coefficient, on the basis of similarities to features like packet lengths in a library of known profiles. Herrmann et al. [61] presented a novel WF method based on a Multinomial Naïve-Bayes classifier, which applied common text mining techniques to the normalised frequency distribution of observable IP packet sizes. Gong et al. [62] developed a remote WF attack on home broadband users, which used the full time series data contained in the observation, and performed kNN classification using dynamic time warping (DTW) distance metric. Lu et al. [63] selected sequence of HTTP request sizes and sequence of HTTP response sizes (except MTU packets) as features for the WF attack, and used Levenshtein distance to measure the similarity between two website fingerprints. Panchenko et al. [64] applied support vector machines (SVM)

to features based on volume, time, and direction of the traffic in the WF attack on Tor and JAP. Cai et al. [65] presented a new webpage fingerprinting attack based on SVM classifier with distance-based kernel, and a novel WF attack based on Hidden Markov Model (HMM). Dyer et al. [66] built four WF classifiers based on naïve Bayes classifier: time (TIME), bandwidth (BW), the variable n-gram (VNG), and VNG++, respectively using the coarse features: total transmission time, total per-direction bandwidth, traffic ''burstiness'', and these three combined. Wang et al. [67] employed SVM to perform WF attack on Tor cell sequences, the SVM was trained by directly computing the kernel matrix from these sequences using distance-based metrics like the Damerau-Levenshtein distance and the proposed fast Levenshtein-like distance. Wang et al. [68] proposed a WF attack based on a kNN classifier applied on a large feature set with weight adjustment. Shi et al. [69] generalized the Bayesian network (BayesNet) based WF attack on static webpages to dynamic webpages, by introducing a new feature called traffic surge period, and adapting the first n Components of Haar Wavelet Transformation. He et al. [70] proposed a novel active WF attack based on SVM with one-against-rest multi-class model, which actively delayed HTTP requests originated from users for a certain period to isolate responding traffic segments containing different web objects. Kwon et al. [71] evaluated CART, C4.5, and kNN classifiers for WF attacks on Tor hidden service clients and servers, and found that kNN worked best. Al-Naami et al. [72] proposed the packet to vector (P2V) approach based on the naïve Bayes classifier, they constructed a corpus from network packets and represented these packets as real-valued vectors, and modeled WF attack using the Global Vector space representation (GloVe). Hayes et al. [73] presented k-fingerprinting, a new WF technique based on random decision forests, which achieved better performance than previous attacks over standard web pages as well as Tor hidden services even against WF defenses. Panchenko et al. [74] proposed CUMUL, a novel WF attack on Tor based on SVM, which sampled features from a cumulative representation of a trace, involving packet size, direction and ordering. Jahani et al. [75], [76] introduced a new WF attack based on Fast Fourier Transform (FFT) to calculate the similarity distance between two instances. Spreitzer et al. [77] provided a WF attack utilizing Android data-usage statistics collected by an unprivileged application, and used the Jaccard index as a metric to determine the similarity between two websites. Al-Naami et al. [78] studied WF attacks using BI-directioNal Dependence (BIND) features (Bi-burst size and time; uni-burst size, time and count; packet size), they evaluated SVM under the closed-world setting, weighted kNN and random forest under the open-world setting, with and without defenses on HTTPS and Tor datasets. Panchenko et al. [79], [80] employed a two-phase SVM-based CUMUL WF attack on Tor hidden services (HSs), which utilized the sum of packet sizes transmitted between the client and each of the entry nodes as additional features, first detected the connection to an HS, and then determined the visited HS within the HS universe. Ejeta et al. [81] used kNN for WF attack on Psiphon traffic. Zhuo et al. [82] proposed a WF attack based on profile hidden Markov model (PHMM), which explicitly considered possible hyperlink transitions when fingerprinting a target website. Qin et al. [83] designed a power estimation based side-channel attack to perform WF, which was carried out by SVM classifier with average amplitudes of 6 equal-sized frequency range bins of FFT-transformed power trace as features. Matyunin et al. [84] provided a WF attack on mobile devices, which uses the RF classifier and features based on the reaction of magnetometer sensors to CPU activity. Zeng et al. [85] proposed a WF attack on malicious websites with traffic aggregated by SNI, which employed RF classifier and took advantage of burst. Zhang et al. [86] utilized deep forest to fingerprint different webpages in the same website, they proposed to use the local request and response sequence (LRRS) as features of Internet traffic in HTTP/1.1 or HTTP/2, the raw features were slid by several different sizes of convolutional layers to generate more features in the multi-grained scanning process and the resulting features were fed into cascade forests which had a multi-layer structure, each layer consisted of the Random Forest classifiers and the Completely-Random Forest classifiers. Meng et al. [87] proposed a novel Website Response Fingerprinting (WRFP) Attack based on response time feature and extremely randomized tree algorithm. Ghiette et al. [88] proposed a two-stage algorithm using MinHash and locality sensitive hashing in combination with the Jaccard similarity to improve the scalability of WF attacks. Ma et al. [89] proposed a context-aware WF attack on encrypted proxies, which employed RF classifier and systematically tackled the training-testing asymmetry problem using a two-stage spatial-temporal flow correlation approach. Kim et al. [90] did a pilot study on real-time WF attacks on Tor hidden services, the classification was conducted with XGBoost, Decision Tree, and Random Forest, results showed enough accuracy in classifying fewer websites. Mitseva et al. [91] proposed two novel WF methods, voting-based and HMM-based, which can take advantage of the consecutive visits of multiple pages of a single website to detect websites. Shen et al. [92], [93] proposed FineWP, a novel fine-grained webpage fingerprinting method based on kNN, RF, Decision Tree classifiers, which utilized length information of packets in bidirectional client-server interactions as distinctive features. Kailong Wang et al. [94] proposed the novel intra-domain WF attack that aimed to differentiate the webpages within the same social media website, which employed RF classifier and utilized temporal and volumetric features including CDN bursts. Mei et al. [95] evaluated WF attacks on Tor using statistical features or package sequence features, based on C4.5, RF, kNN, or Quadratic Discriminant Analysis (QDA) classifiers, separately. Okazaki et al. [96] proposed a WF attack using SVM classifier, and Virtual Set Size

fluctuation of a specific process regarding website browsing as a feature. Li et al. [97] constructed a resource loading tree (RLTree) to represent a website, based on the multiple initial TCP sessions generated by visiting the website, and proposed a novel WF attack based on RLTree similarity. Hongcheng Zou et al. [98] presented Probabilistic Fingerprinting (PF), a new WF attack based on kNN, using topic probability vectors of traffic instances as features. Kexin Zou et al. [99] proposed a novel lightweight WF attack on Bitcoin hidden service, using a random decision forest classifier with features from TLS packet size and direction.

Early research adopted simple and direct methods to fingerprint websites. Cheng et al. [100] utilized the HyperText Markup Language (HTML) file size, total object size, total number of objects and a link structure algorithm to identify the webpage. Hintz [101] presented a WF attack which simply observed the amount of encrypted data that was transferred.

Deep learning approaches usually do better than traditional machine learning. DF [27] reported 98% accuracy, TF [33] reported 95% accuracy, while CUMUL [74] reported 92.03% accuracy.

B. CHALLENGES AND SOLUTIONS

In the real world, WF attacks face many challenges presented by all parties directly or indirectly involved, including Internet users, web service providers, browser developers, operating system (OS) developers, Internet service providers, computing and routing hardware manufacturers, etc. These stakeholders can affect the contents and/or behavior of the target traffic within their power. Among the challenges, significant ones include overlapping traffic of different webpages, concept drift, the base rate fallacy, etc., other ones include browser versions, OS versions, network locations, routing policies, etc.

Multi-tab browsing may produce overlapping traffic traces of two or more webpages, if the user opens a new tab of the browser requesting a new webpage when one or more old webpages are still loading elements. WF attacks on such overlapping traffic traces face the challenge of distinguishing between the webpages and identifying them. Juarez et al. [102] observed a dramatic drop in the accuracy of classifiers trained on single tab traces when tested on the overlapping traces of two webpages, they also observed a drop in the accuracy when the size of the world increased, their experiment implied that feature selection might be more important than learning models and shorter delay between the loading of the two webpages did not mean smaller distance between the observed overlapping trace and either of the two webpage traces. Wang et al. [103] splitted Tor cell sequences by distinguishing between different web pages that may occur sequentially or even in parallel, they demonstrated the effectiveness of time-based splitting and classification-based splitting. Gu et al. [104] identified the overlapping webpages separately, they utilized the delay between the loading of the two webpages which they called think time, analyzed the traffic during the think time and selected fine-grained features to identify the first page, employed coarse features to identify the second page from the remaining traffic. Xu et al. [105] presented a new BalanceCascade-XGBoost scheme to identify the start point of the second page, they also developed a new classifier based on random forests which can accurately classify webpages given only the small chunk of packets between the start time of the two webpages. Yin et al. [106] built a WF classifier based on XGBoost algorithm, replacing the original random forest classifier in [105]. Cui et al. [107] proposed a splitting algorithm based on Hidden Markov Model to identify two continuous network traces and a sectioning algorithm to identify overlapping network traces, which divided the trace into multiple sections and performed website prediction on each section independently, based on the hypothesis that if two traces overlap, the beginning of the first trace and the end of the second trace would be unaffected. Gong et al. [108] improved known solutions to splitting with a new framework called Coarse-Decided Score-Based (CDSB), they used a random forest classifier with 511 features extracted by expert knowledge to decide how many splits there are, extended the XGBoost scheme in [105] to score each outgoing packet in the trace, chose the highest-scoring packet as a split in each round, and eliminated nearby packets from consideration as splits for future rounds. Cui et al. [109] proposed a CNN-based classifier to distinguish between one-page and multiple-page traces, evaluated the DL model on partial traces and improved the performance on traces missing the head part by adding the head detection, constructed two DL models for webpage prediction on two-page overlapping traces, training on the first N and last N packets to predict the first and second webpages in traces respectively. Guan et al. [110] proposed a Block Attention Profiling Model (BAPM) which fully utilized the whole multi-tab packet trace including the overlapping area to generate a tab-aware representation from direction sequences, divided the trace into blocks and attention-based profiling was used to group blocks belonging to the same webpage tab, and identified each website page under a global view. Chen et al. [111] proposed an end-to-end Website Fingerprint Detection (WFD) method, based on the idea of considering each monitored trace of interest in traffic traces as a specific object in an image, and adapted object detection methods in computer vision to the multi-tab browsing scenario.

Concept drift in WF attacks refers to the phenomenon that the contents of webpages are constantly changing over time, which results in changes in the patterns of traffic traces and may affect the accuracy of WF attacks. Resilent WF classifiers can grasp salient and stable features, and remain effective over time. Juarez et al. [102] studied the effect of staleness on WF and observed the extremely fast drop in accuracy over time, modeled the updating cost of a WF system with respect to the webpage changes. Wang et al. [103] learned from experience that a small amount of data was enough for training an effective WF classifier and therefore keeping the data fresh was easy, they provided several

practical schemes for updating the training set. Al-Naami et al. [78] addressed the challenge through regularly updating the model, they studied the effect of fixed update in which fixed updates were applied to re-train the model periodically, and dynamic update in which model was re-trained whenever there was a drift between the current data and previously seen training data. Attarian et al. [112], [113] proposed AdaWFPA which avoided concept drift by updating its model over time. Zhu et al. [114] proposed a novel WF attack framework, Persistent Attack of Student (PAS), which integrated self-training mechanism with DL, trained new DL models using concept drift datasets with pseudo labels to alleviate the impact of concept drift. Yanbin Wang et al. [53] revealed that under concept drift WF attacks suffered more severe performance degradation in a open-world setting than in a closed-world setting.

The base rate fallacy refers to the bias in the evaluation of the WF attack introduced by the base rate or prior, i.e., the probability of a user visiting a monitored webpage a priori, under the open-world scenario. When the base rate is low, even if the WF classifier reports high true positive rate (TPR) and low false positive rate (FPR), the rate of successful attack still can be much lower [102], [115]. Several methods have been proposed to mitigate the effects of the base rate fallacy, including the Classify-Verify approach, Precision Optimizers (POs), etc. The Classify-Verify approach rejects predictions of the classifier when estimated probabilities are lower than a threshold determined by training [102]. POs adopt strategies similar to the Classify-Verify approach, and the presented confidence-based POs, distance-based POs, and ensemble POs are inspired by clustering and ensemble learning techniques [116].

IV. WF DEFENSES

WF defenses aim to defend against WF attacks, thus adopting various methods to modify the content and behavior of traffic, conceal the webpages visited and protect the privacy of both parties of the communication. WF defenses evolve along with WF attacks, like in an arms race, in which both sides keep coming up with new strategies and approaches.

A. STRATEGIES

The strategies of WF defenses include noise, mimicry, regularization, adversarial examples, etc. The noise strategy randomly adds dummy packets to target traffic traces to disrupt the classification. The mimicry strategy modifies the traffic of a webpage to look like another to confuse the classifier. The regularization strategy defines fixed rules and patterns for all webpage traffic to follow to reduce information leakage [117]. The adversarial examples strategy generates adversarial traffic traces against the classifier.

B. APPROACHES

Various approaches to WF defenses have been proposed, following different strategies. Deep learning underpins several latest and effective WF defenses, other approaches rely on traditional methods.

Many deep learning architectures have been employed in realizing WF defenses. Jiang et al. [118] proposed a novel WF defense called PST, which predicted subsequent fuzzy bursts with a neural network, given a few past bursts of a trace as input, then searched small but effective adversarial perturbation directions based on observed and predicted bursts, finally transfered the perturbation directions to the remaining bursts. Rahman et al. [141], [142] proposed Mockingbird, a technique for generating traces that resisted adversarial training by moving randomly in the space of viable traces and not following more predictable gradients. Mockingbird gradually changed the defended source sample to get closer to a randomly selected target sample, until a trained deep learning-based WF classifier called the detector predicted that the class of the sample had changed. Hou et al. [119] proposed a novel WF defense based on adversarial examples, generated by WF-GAN, a GAN with an additional WF classifier component. Sadeghzadeh et al. [120] proposed Adversarial Website Adaptation (AWA), a new defense against WF attack using adversarial deep learning approaches. Gong et al. [121] proposed Surakav, a tunable and practical WF defense with reasonable overhead, which made use of a GAN to generate realistic sending patterns and regulated buffered data according to the sampled patterns. Sun et al. [122] proposed WF-UAP, a WF defense which employed GAN to generate Universal Adversarial Perturbations (UAPs) to add to the defended traffic traces.

Traditional methods have been employed since the earliest WF defenses came into being. Sun et al. [58] described three traffic-shaping mechanisms, i.e., padding, mimicking, and morphing, which can be used to defend against WF attacks. Levine et al. [123] introduced defensive dropping, a variation of cover traffic that better defended against timing attacks. Shmatikov et al. [124] proposed adaptive padding, a defense against timing analysis, in which intermediate mixes inject dummy packets into statistically unlikely gaps in the packet flow, destroying timing fingerprints without adding any latency to application traffic. Wright et al. [125] proposed a novel method for thwarting statistical traffic analysis algorithms by optimally morphing one class of traffic to look like another class. Dyer et al. [66] proposed Buffered Fixed-Length Obfuscator (BuFLO), which operated by sending fixed-length packets at a fixed interval for at least a fixed amount of time. Cai et al. [65], [126] proposed Congestion-Sensitive BuFLO (CSBuFLO), which optimized BuFLO to make the protocol congestion sensitive, rate adaptive, and efficient at hiding macroscopic website features, such as total size and the size of the last object. Cai et al. [115] proposed Tamaraw, based on an extension of the concept of optimal partitioning and feature hiding, which extended and tuned BuFLO to hide the most significant traffic features, the packet size was set at 750 bytes rather than the MTU, outgoing traffic was fixed at a higher packet interval than incoming traffic to reduce the overhead in both bandwidth and time. Wang et al. [68] constructed a principled and provably private defense using an approximation of the smallest

common supersequence problem and clustering techniques, based on the finding that bandwidth-optimal simulatable, deterministic defense was to transmit packets using supersequences over anonymity sets. Nithyanand et al. [127] proposed Glove, an SSH-based highly-tunable defense, which used existing knowledge of website traces to add cover traffic conservatively, extended the traffic morphing principles to cover all features, and provided information-theoretic security guarantees. Juarez et al. [128] proposed Website Traffic Fingerprinting Protection with Adaptive Defense (WTF-PAD), which adapted Adaptive Padding (AP) to combat WF attacks on Tor, and included a number of link-padding primitives that enable more sophisticated padding strategies than basic AP. Wang et al. [129] proposed Walkie-Talkie (WT), which molded burst sequences so that sensitive and non-sensitive pages look the same. WT modified the browser to communicate in half-duplex mode, and produced easily moldable burst sequences to leak less information and cost little overhead. Cherubin et al. [130] proposed two application-level defenses, one was Application Layer Padding Concerns Adversaries (ALPaCA) for the server, the other was client-side Lightweight application-Layer Masquerading Add-on (LLaMA) for the client. ALPaCA altered the size of each content type, e.g., PNG, HTML, CSS of an index.html page to conform to the size distribution of a significant fraction of the total Tor .onion site pages. LLaMA added extra delays to the HTTP requests, which altered the order of the requests in a similar way to randomized pipelining (RP). Zhuo et al. [82] presented two WF defenses, one was Probabilistic MTU padding, which selectively padded packets to MTU size with an equal probability for each packet, the other was Probabilistic Dummy packet, which inserted dummy packets of random direction and size into the testing packet sequence, each packet had an equal probability to place the to-be-inserted packet ahead of it. Lu et al. [131] introduced DynaFlow, a highly tunable WF defense based on dynamically-adjusting flows, which used fixed burst patterns with dynamically-changing intervals between packets to hide the websites a user visited. Chan-Tin et al. [132] proposed a WF defense which first grouped websites with similar number and size of packets into clusters, then performed traffic morphing within each cluster to make all other websites have the same packet sizes as the website with the biggest packet size. Cui et al. [133], [134] introduced a WF defense which generated cover traffic that looked like historical network traffic of user visiting websites. Matyunin et al. [84] presented WF defense measures against information leakage through magnetometer sensors, including physical shielding with ferromagnetic materials, placing magnetometer sensors far away from CPUs when designing smartphone motherboards, lowering sampling rate, limiting user permission to access magnetometers, etc. Liu et al. [135] proposed a WF defense based on genetic algorithm to generate adversarial samples. Cadena et al. [136], [137] proposed TrafficSliver, network-layer and application-layer WF defenses for Tor, which distributed traffic between the user and the middle OR over multiple entry ORs to limit the information available to an attacker. The network-layer TrafficSliver realized the concept of multipathing entirely within Tor, while the client-side application-layer TrafficSliver distributed single HTTP requests for different web objects over distinct Tor entry nodes. Gong et al. [108] proposed two novel zero-delay lightweight defenses, FRONT and GLUE. FRONT focused on obfuscating the trace front with dummy packets, and also randomized the number and distribution of dummy packets for trace-to-trace randomness to impede the attacker's learning process. GLUE added dummy packets between separate traces so that they appeared to the attacker as a long consecutive trace, rendering the attacker unable to find their start or end points. Abusnaina et al. [138] proposed a novel defense mechanism using a per-burst injection technique, called Deep Fingerprinting Defender (DFD), against deep learning-based WF attacks, which had two operation modes, one-way and two-way injection. DFD was designed to break the inherent patterns preserved in Tor traffic traces by carefully injecting dummy packets within every burst. Henri et al. [139] proposed a multihoming-based WF defense, in which a user can split traffic among the networks, and designed a novel multipath scheduler called HyWF, which can be combined with other defenses like adaptive padding and Walkie-Talkie. Al-Naami et al. [140] proposed BiMorphing, a WF defense that obfuscated original website traffic patterns through the use of double sampling and mathematical optimization techniques to deform packet sequences and destroy traffic flow dependency characteristics used by attackers to identify websites. Nasr et al. [2] proposed a WF defense to apply blind adversarial perturbations on the patterns of live network traffic, which were generated through solving specific optimization problems tailored to WF attacks, and included changing the timings and sizes of packets, as well as inserting dummy network packets. Shusterman et al. [30], [31] proposed a defense against cache-based WF attack, which created spurious network activity to introduced noise in the cache, thus masking the website rendering activity. Wang [117] created the first defense that was strong enough for the one-page setting by augmenting Tamaraw with greater randomization overhead so that its anonymity sets were more evenly dispersed. Gulmezoglu [49] proposed an XAI-based obfuscation defense technique as a countermeasure against microarchitecture-based WF attacks. Hou et al. [143] presented a novel effective defense, which generated universal perturbation that can transform original examples to adversarial examples tailored to specific WF attack models. Hou et al. [144] proposed a WF defense named Attack to Attack (A2A) that leveraged adversarial examples to attack the WF attacker's classifier. A2A manipulated traffic iteratively according to the output of a substitute model which was an elaborate model intentionally learning a

similar classification boundary with the attacker's model. Huang et al. [145] proposed two effective and efficient algorithms, PadS and PadI, to defend Shadowsocks against WF attacks, which were based on the size and the true distribution of time series of traffic data packets, respectively. Luo et al. [146] proposed Random Bidirectional Padding (RBP), a novel website fingerprinting defense technology based on time sampling and random bidirectional packets padding, which can covert the real packets distribution to destroy the Inter-Arrival Time (IAT) features in the traffic sequence and increase the difference between the datasets with random bidirectional virtual packets padding. Shan et al. [147] proposed Dolos, which injected dummy packets into traffic traces by computing input-agnostic adversarial patches that disrupt deep learning classifiers used in WF attacks. Patches are then applied to alter and protect user traffic in real time. Importantly, these patches are parameterized by a user-side secret, ensuring that attackers cannot use adversarial training to defeat Dolos. Dahanayaka et al. [37] proposed FRONT-U, which defended website visits by obfuscating transitions between uploads and downloads in trace fronts and provided similar privacy as the defense FRONT, with half the data overhead. Li et al. [148] proposed cache shaping, a defense against cache-based WF attacks, which produced dummy cache activities by introducing dummy I/O operations and implementing with multiple processes, hiding fingerprints when a user visited websites. Holland et al. [149] presented a realistic and novel defense, RegulaTor, which took advantage of common patterns in web browsing traffic to reduce defense overhead, worked by regularizing the size and shape of packet 'surges' that frequently occur in download traffic, masking potentially revealing features, 'surge' was broadly defined as a large number of packets sent over a short period of time. Tang et al. [150] proposed segmented adversary defense (SAD) for deep learning-based WF attacks, in which sequence data were divided into multiple segments to ensure that SAD was feasible in real scenarios, then the adversarial examples for each segment of data can be generated by SAD. Liang et al. [151] observed that Walkie-Talkie significantly increased the page loading time (time overhead) although the bandwidth overhead was not high and analyzed the cause of the increased page loading time, and presented a defending approach called Tail Time (TT), which addressed the problem by limiting the maximum time for which a pending request can block subsequent requests. Zhang et al. [152] proposed RAP, an application layer WF defense based on randomizing the request order and location of website resources. Li et al. [54] proposed a new WF defense called Minipatch based on adversarial patches, which injected extremely few dummy packets in real-time traffic. Ling et al. [153] proposed a genetic-programming-based variant cover traffic search technique to generate WF defense strategies for effectively injecting dummy Tor cells into the raw Tor traffic. Smith et al. [154] designed and implemented the QCSD framework, which leveraged QUIC and HTTP/3 to emulate existing WF defenses by bidirectionally adding cover traffic and reshaping connections solely from the client.

V. DEEP LEARNING FOR WF ATTACKS

In WF attacks, deep learning mainly exercises its power through the role of classifiers. We focus on the learning paradigm, architecture, performance, and model update policy to survey deep learning for WF attacks, and look into the future. We summarized the paradigms and architectures of deep learning for WF attacks in existing works in Table 1.

A. PARADIGMS

As shown in Table 1, the paradigms of deep learning for WF attacks in existing works mainly cover supervised learning (26 models), semi-supervised learning (2 models), transfer learning (2 models), metric-learning (4 models), and meta-learning (1 model). The choice of deep learning paradigms for WF attacks is greatly influenced by the amount of available labeled and unlabeled training examples, i.e., traffic traces.

In supervised learning, when the labeled traffic traces for training are insufficient, data augmentation can help generate more data with lower cost to help improve WF attacks, such methods include HDA, Tripod, and Bionic data augmentation. The HDA method [48] can be used in a harmonious manner to expand a tiny training dataset to an arbitrarily large collection, which involved both intrasample and intersample data transformations. Experiments showed that HDA can boost deep learning WF attack models like Var-CNN in both closed-world and open-world settings, at the absense and presence of strong defense. Tripod [47] used three manipulations of Injecting, Removing, and Losing on one collected traffic trace to generate several augmented traces, reflecting the changes or exceptions of the Internet. The Injecting manipulation introduced background traffic. The Removing and Losing manipulations simulated the packet loss and packet retransmissions due to network congestion or transmission errors. Experimental results showed that Tripod had enhanced six deep learning WF attack models and may work with more. Bionic traces [55] can be generated based on the rearranged send-and-receive pairs (SRPs), expensive experiments showed that bionic traces successfully simulated the website traffic and relieved the data-hungry problem of deep learning-based WF attacks.

B. ARCHITECTURES

As shown in Table 1, the architectures of deep learning for WF attacks in existing works include common basic neural networks like FC, Self-attention, GRU, MLP, SDAE, CNN (including ResNet, VGG), RNN (including LSTM), GNN, GAN, Siamese networks, Triplet networks and their variants, and popular advanced neural networks like AWF, DF, Var-CNN, TF, etc. Together with other necessary components, they form the classifiers for WF attacks.

C. PERFORMANCE

Existing works on deep learning for WF attacks evaluated the performance of their models on various datasets, under

TABLE 1. Deep learning paradigms and architectures for WF attacks.

Paradigms	Models	Architectures
Supervised learning	SDAE [25]	SDAE
	AWF [26]	SDAE, CNN, LSTM
	DF [27]	CNN
	GRU and ResNet [28]	GRU, ResNet-50
	p-FP [29]	MLP, CNN
	Cache-based WF [30], [31]	CNN, LSTM
	Var-CNN [32]	ResNet-18
	Tik-Tok [34]	DF
	2ch-TCN [35]	CNN
	Realistic WF [109]	CNN, LSTM
	Multi-session WF [39]	LSTM
	Side-channel information-based WF [41]	CNN, LSTM
	BurNet [43]	CNN
	DNNF [44]	CNN
	GAP-WF [45]	GNN
	Cross-trace WF [46]	DF
	DNN with Blind adversarial training [2]	DF
	DNN with Tripod data augmentation [47]	DF, Var-CNN, ResNet-18, ResNet-34, VGG-16, VGG-19
	DNN with HDA data augmentation [48]	Var-CNN, ResNet-34
	Microarchitecture-based WF [49]	1-D CNN
	BAPM [110]	CNN, Self-attention
	FDF [52]	CNN, FC, Self-attention
	snWF [53]	CNN
	WFD [111]	1-D ResNets
	DNN with Minipatch adversarial training [54]	DF
	DNN with Bionic data augmentation [55]	Var-CNN
Semi-supervised learning	GANDaLF [40]	GAN
	PAS [114]	DCNN, DF, AWF
Transfer learning	AF [42]	Domain adversarial network
	TLFA [51]	CNN, MLP
Metric-learning	TF [33]	Triplet networks
	CPWF [38]	CNN
	CNN-BiLSTM-based Siamese networks [50]	Siamese networks, CNN, LSTM
	Online WF [56]	TF
Meta-learning	MBL [57]	CNN

the scenarios of the closed-world setting, or the open-world setting. It is not easy to compare their performance directly, thus we choose several representative models to do the survey, including DF, TF, snWF, Online WF, as shown in Table 2.

The DF [27] model achieved over 98% accuracy on Tor traffic with no defenses. DF remained effective in the open-world setting, with 99% precision and 94% recall on traffic without defense, and can still get 96% precision and 68% recall on traffic with the WTF-PAD defense.

The TF [33] model used triplet networks for NSL, achieved up to 95% accuracy using only 20 examples per website, and near 85% accuracy when using only five examples per class and the feature extractor was pre-trained with a three-year-old dataset. TF remained effective in a small open-world setting, achieving approximately 90% precision and 80% recall when tuned for precision. When the size of the world was significantly increased, its performance degraded significantly to 30% precision and 70% recall.

The snWF [53] model managed to determine whether a user is visiting a monitored website, with a true positive rate of 98.1% and a false positive rate of 5.7%, in a large open-world setting with 400,000 websites. In the case of only 100 training samples per monitored website, snWF achieved 90.4% balanced accuracy and 84.2% TPR. A more realistic attack scenario, termed as wide-world, was also evaluated. In the face of concept drift, snWF was found to be more resilient than any other attacks.

TABLE 2. Performance of deep learning models.

Models	Accuracy
DF [27]	98% on non-defended dataset
TF [33]	95% with 20 examples per website
snWF [53]	90.4% with 100 examples per website
Online WF [56]	95% with 5 websites

The Online WF [56] model evaluated WF using genuine Tor traffic as ground truth, and under a true open world setting achieved by adapting TF attack to an online setting and training the WF models on data safely collected on a Tor exit relay. They achieved a WF accuracy of above 95% when monitoring a small set of 5 popular websites, but that accuracy quickly degraded to less than 80% when monitoring as few as 25 websites.

D. THE FUTURE

In the future, new deep learning architectures and paradigms will come up, offering new choices for constructing novel WF classifiers. Currently unexplored deep learning classifiers will be tested for WF classification tasks, and the ones with better performance will be chosen. Few-shot learning, foundation models, and adversarial training are becoming popular, and are likely to be widely adopted in WF attacks.

VI. DEEP LEARNING FOR WF DEFENSES

In WF defenses, deep learning mainly takes effect through its role in realizing the defense strategy, e.g., generating sending patterns or adversarial examples, etc. We focus on the architecture, efficacy, and overhead to survey deep learning for WF defenses, and look into the future.

A. ARCHITECTURE, EFFICACY AND OVERHEAD

Common deep learning architectures for WF defenses include RNN, GAN, transformer, etc.

In PST [118], an encoder-decoder RNN served the purpose of predicting subsequent parts of a trace after observing the first several parts. The neural network included three components, the embedding layer, the encoder layer, and the decoder layer. In the embedding layer, each burst size in the observed network trace was embedded into a vector. The encoder layer adopted an LSTM network, took the embedded vectors as input, and generated compressed features. The decoder layer adopted another LSTM network with the attention mechanism, took the compressed feature vectors as input, used a fully-connected neural network to scan and calculate a joint score vector for each hidden state of the encoder, thus predicting the subsequent burst sizes of the network trace. Experimental results over a public closed-world dataset demonstrated that PST can reduce the accuracy of

DF by 87.6%, under 31% bandwidth overhead, observing only the first 10 bursts of the network trace. Moreover, PST adapted to WF attacks dynamically, which could be retrained or updated.

WF-GAN [119] adopted a GAN which followed the paradigm of AdvGAN [155], which adopted similar architectures for generator and discriminator with pix2pix [156] and CycleGAN [157]. Pix2pix used a U-Net-based architecture for the generator, and a convolutional PatchGAN classifier for the discriminator, both generator and discriminator used modules of the form convolution-BatchNorm-ReLU. CycleGAN used PatchGAN for the discriminator. WF-GAN was evaluated on DF, it achieved 90% success rate with at most 15% overhead for untargeted defense, and over 90% targeted defense success rate when the size of target website set was twice as many as that of the source website set.

In Mockingbird [141], [142], AWF and DF played the role of detector models, which were trained to detect whether the predicted class of the sample had changed. Mockingbird dropped the accuracy of DF and Var-CNN hardened with adversarial training from 98% to 42–58% while incurring only 58% bandwidth overhead.

AWA [120] used adversarial deep learning approaches to create a transformer set in each run, so that each website had a unique transformer, each transformer generated adversarial traces to evade the adversary's classifier. They accommodated secret random elements in the training phase of transformers in order for AWA to generate various sets of transformers in each run. They run AWA several times and created multiple sets of transformers. There were two versions of AWA, including Universal AWA (UAWA) and Non-Universal AWA (NUAWA). NUAWA needed to access the entire trace of a website in order to generate an adversarial trace, while there was no such need for UAWA. If an adversary and a target user selected different sets of transformers, the accuracy of adversary's classifier was almost 19.52% and 31.94% with almost 22.28% and 26.28% bandwidth overhead in UAWA and NUAWA, respectively. If a more powerful adversary generated adversarial traces through multiple sets of transformers and trained a classifier on them, the accuracy of adversary's classifier was almost 49.10% and 25.93% with almost 62.52% and 64.33% bandwidth overhead in UAWA and NUAWA, respectively.

WF-UAP [122] had three components: the generator, the discriminator, and the target WF classifier. An encoder and decoder based architecture was specified to realize the functions of the discriminator and the generator respectively. A random sample from a normal distribution was fed into the generator to produce a perturbation, which was then scaled and added to the traffic trace, the perturbed instance was then fed into the discriminator and the target classifier. Experimental results over a public dataset demonstrated that WF-UAP reduced the accuracy of Var-CNN from 98% to 15% with at most 20% bandwidth overhead.

Surakav [121] made use of a generator that can output infinite non-repeated sending patterns, which was achieved by training a well-designed GAN to mimic realistic traffic patterns of different webpages. Surakav reduced overhead while maintaining effectiveness by tunneling packets through different sending patterns rather than a constant pattern. Experiments showed that Surakav was able to reduce the attacker's true positive rate by 57% with 55% data overhead and 16% time overhead.

B. THE FUTURE

In the future, new deep learning paradigms, architectures, and models will arise, new WF defense strategies and approaches may also come up. Unexplored combinations of deep learning tools and WF defense requirements will be tested continually, and the ones with better efficacy and smaller overhead will stand out.

VII. SUMMARY

We briefly surveyed deep learning for WF attacks and defenses in this paper. First, we introduced the common usages of deep learning in WF attacks and WF defenses. Second, we surveyed deep learning, WF attacks, and WF defenses separately in detail. For deep learning, we surveyed the paradigms, architectures, and performance metrics. For WF attacks, we surveyed the approaches, challenges and solutions. The approaches included deep learning, traditional machine learning, and other methods. Challenges and solutions covered multi-tab browsing, concept drift, and the base rate fallacy. For WF defenses, we surveyed the strategies and approaches. Third, we surveyed deep learning for WF attacks, and deep learning for WF defenses. In deep learning for WF attacks, we surveyed in detail the deep learning paradigms, architectures of WF attack models and the performance of several representative WF attack models, and looked into the future. In deep learning for WF defenses, we surveyed the architecture, efficacy and overhead of deep learning models in WF defenses, and looked into the future.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proc. 13th USENIX Secur. Symp.* San Diego, CA, USA, Aug. 2004, pp. 303–320.

[2] M. Nasr, A. Bahramali, and A. Houmansadr, "Defeating DNN-based traffic analysis systems in real-time with blind adversarial perturbations," in *Proc. 30th USENIX Secur. Symp.*, Aug. 2021, pp. 2705–2722. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/nasr

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, vol. 25, F. Pereira, C. Burges, L. Bottou, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2012.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[5] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, *Transfer Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2020, pp. 3–13.

[6] X. Han et al., "Pre-trained models: Past, present and future," *AI Open*, vol. 2, pp. 225–250, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666651021000231

[7] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 857–876, Jan. 2023.

[8] Y. Bengio, Y. LeCun, and G. E. Hinton, "Deep learning for AI," *Commun. ACM*, vol. 64, pp. 58–65, Jun. 2021, doi: 10.1145/3448250.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[10] B. McCann, J. Bradbury, C. Xiong, and R. Socher, "Learned in translation: Contextualized word vectors," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017.
[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA, Jun. 2019, pp. 4171–4186. [Online]. Available: https://www.aclweb.org/anthology/N19-1423
[12] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020, pp. 1877–1901.
[13] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," 2021, *arXiv:2107.13586*.
[14] C.-K. Hsieh, L. Yang, Y. Cui, T.-Y. Lin, S. Belongie, and D. Estrin, "Collaborative metric learning," in *Proc. 26th Int. Conf. World Wide Web*, Geneva, Switzerland, Apr. 2017, pp. 193–201, doi: 10.1145/3038912.3052639.
[15] Y. Tian, X. Zhao, and W. Huang, "Meta-learning approaches for learning-to-learn in deep learning: A survey," *Neurocomputing*, vol. 494, pp. 203–223, Jan. 2022, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231222004684
[16] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn. (PMLR)*, in Proceedings of Machine Learning Research, vol. 119, A. Singh, Ed., Jul. 2020, pp. 1597–1607. [Online]. Available: http://proceedings.mlr.press/v119/chen20j.html
[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. 38th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 139, M. Meila and T. Zhang, Eds., Jul. 2021, pp. 8748–8763. [Online]. Available: https://proceedings.mlr.press/v139/radford21a.html
[18] S. Hochreiter, "Toward a broad AI," *Commun. ACM*, vol. 65, no. 4, pp. 56–57, Mar. 2022, doi: 10.1145/3512715.
[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2017.
[20] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–16. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy
[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2014.
[22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[23] Y.-S. Lim, H.-C. Kim, J. Jeong, C.-K. Kim, T. Kwon, and Y. Choi, "Internet traffic classification demystified: On the sources of the discriminative power," in *Proc. 6th Int. Conf.*, New York, NY, USA, Nov. 2010, pp. 1–12, doi: 10.1145/1921168.1921180.
[24] T. Wang, "Website fingerprinting: Attacks and defenses," Ph.D. dissertation, Univ. Waterloo, Waterloo, ON, Canada, 2015.
[25] K. Abe and S. Goto, "Fingerprinting attack on Tor anonymity using deep learning," in *Proc. Asia–Pacific Adv. Netw.*, 2016, pp. 15–20.
[26] V. Rimmer, D. Preuveneers, M. Juarez, T. V. Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.
[27] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2018, pp. 1928–1943, doi: 10.1145/3243734.3243768.
[28] X. He, J. Wang, Y. He, and Y. Shi, "A deep learning approach for website fingerprinting attack," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2018, pp. 1419–1423.
[29] S. E. Oh, S. Sunkam, and N. Hopper, "FP: Extraction, classification, and prediction of website fingerprints with deep learning," *Privacy Enhancing Technol.*, vol. 2019, no. 3, pp. 191–209, Jul. 2019, doi: 10.2478/popets-2019-0043.
[30] A. Shusterman, L. Kang, Y. Haskal, Y. Meltser, P. Mittal, Y. Oren, and Y. Yarom, "Robust website fingerprinting through the cache occupancy channel," in *Proc. 28th USENIX Secur. Symp.*, Santa Clara, CA, USA, Aug. 2019, pp. 639–656.
[31] A. Shusterman, Z. Avraham, E. Croitoru, Y. Haskal, L. Kang, D. Levi, Y. Meltser, P. Mittal, Y. Oren, and Y. Yarom, "Website fingerprinting through the cache occupancy channel and its real world practicality," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2042–2060, Oct. 2021.
[32] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: A data-efficient website fingerprinting attack based on deep learning," in *Proc. Privacy Enhancing Technol. Symp. (PETS)*, 2019, pp. 292–310, doi: 10.2478/popets-2019-0070.
[33] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet fingerprinting: More practical and portable website fingerprinting with N-shot learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2019, pp. 1131–1148, doi: 10.1145/3319535.3354217.
[34] M. S. Rahman, P. Sirinam, N. Mathews, K. G. Gangadhara, and M. Wright, "Tik-Tok: The utility of packet timing in website fingerprinting attacks," in *Proc. Privacy Enhancing Technol. Symp. (PETS)*, 2020, pp. 5–24, doi: 10.2478/popets-2020-0043.
[35] M. Wang, Y. Li, X. Wang, T. Liu, J. Shi, and M. Chen, "2ch-TCN: A website fingerprinting attack over Tor using 2-channel temporal convolutional networks," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–7.
[36] T. Dahanayaka, G. Jourjon, and S. Seneviratne, "Understanding traffic fingerprinting CNNs," in *Proc. IEEE 45th Conf. Local Comput. Netw. (LCN)*, Nov. 2020, pp. 65–76.
[37] T. Dahanayaka, G. Jourjon, and S. Seneviratne, "Dissecting traffic fingerprinting CNNs with filter activations," *Comput. Netw.*, vol. 206, Apr. 2022, Art. no. 108770. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128622000068
[38] S. Wang, L. Wang, S. Yin, H. Zhao, and H. Shentu, "CPWF: Cross-platform website fingerprinting based on multi-similarity loss," in *Proc. Int. Conf. Netw. Netw. Appl. (NaNA)*, Dec. 2020, pp. 73–80.
[39] A. Ramezani, A. Khajehpour, and M. J. Siavoshani, "On multi-session website fingerprinting over TLS handshake," in *Proc. 10th Int. Symp. Telecommun. (IST)*, Dec. 2020, pp. 211–216.
[40] S. E. Oh, N. Mathews, M. S. Rahman, M. Wright, and N. Hopper, "GANDaLF: GAN for data-limited fingerprinting," *Privacy Enhancing Technol.*, vol. 2021, no. 2, pp. 305–322, Apr. 2021, doi: 10.2478/popets-2021-0029.
[41] H. Wang, H. Sayadi, A. Sasan, P. D. S. Manoj, S. Rafatirad, and H. Homayoun, "Machine learning-assisted website fingerprinting attacks with side-channel information: A comprehensive analysis and characterization," in *Proc. 22nd Int. Symp. Quality Electron. Design (ISQED)*, Apr. 2021, pp. 79–84.
[42] C. Wang, J. Dani, X. Li, X. Jia, and B. Wang, "Adaptive fingerprinting: Website fingerprinting over few encrypted traffic," in *Proc. 11th ACM Conf. Data Appl. Secur. Privacy*, New York, NY, USA, Apr. 2021, pp. 149–160, doi: 10.1145/3422337.3447835.
[43] M. Shen, Z. Gao, L. Zhu, and K. Xu, "Efficient fine-grained website fingerprinting via encrypted traffic analysis with deep learning," in *Proc. IEEE/ACM 29th Int. Symp. Quality Service (IWQOS)*, Jun. 2021, pp. 1–10.
[44] M. Guo, J. Fei, and Y. Meng, "Deep nearest neighbor website fingerprinting attack technology," *Secur. Commun. Netw.*, vol. 2021, pp. 5399816:1–5399816:14, Jan. 2021, doi: 10.1155/2021/5399816.
[45] J. Lu, G. Gou, M. Su, D. Song, C. Liu, C. Yang, and Y. Guan, "GAP-WF: Graph attention pooling network for fine-grained SSL/TLS website fingerprinting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
[46] J. Dani and B. Wang, "HiddenText: Cross-trace website fingerprinting over encrypted traffic," in *Proc. IEEE 22nd Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Aug. 2021, pp. 274–281.

[47] Y. Zhang, X. Sun, X. Qin, C. Li, S. Wang, and Y. Xie, "Tripod: Use data augmentation to enhance website fingerprinting," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Sep. 2021, pp. 1–7.

[48] M. Chen, Y. Wang, Z. Qin, and X. Zhu, "Few-shot website fingerprinting attack with data augmentation," *Secur. Commun. Netw.*, vol. 2021, p. 2840289:1–2840289:13, Jan. 2021, doi: 10.1155/2021/2840289.

[49] B. Gulmezoglu, "XAI-based microarchitectural side-channel analysis for website fingerprinting attacks and defenses," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 4039–4051, Nov. 2022.

[50] M. Guo and J. Fei, "Website fingerprinting attacks based on homology analysis," *Secur. Commun. Netw.*, vol. 2021, Oct. 2021, Art. no. 6070451, doi: 10.1155/2021/6070451.

[51] M. Chen, Y. Wang, H. Xu, and X. Zhu, "Few-shot website fingerprinting attack," *Comput. Netw.*, vol. 198, Jan. 2021, Art. no. 108298. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621003108

[52] Y. Sun, X. Luo, H. Wang, and Z. Ma, "A method for identifying Tor users visiting websites based on frequency domain fingerprinting of network traffic," *Secur. Commun. Netw.*, vol. 2022, Jan. 2022, Art. no. 3306098, doi: 10.1155/2022/3306098.

[53] Y. Wang, H. Xu, Z. Guo, Z. Qin, and K. Ren, "SnWF: Website fingerprinting attack by ensembling the snapshot of deep learning," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1214–1226, 2022.

[54] D. Li, Y. Zhu, M. Chen, and J. Wang, "Minipatch: Undermining DNN-based website fingerprinting with adversarial patches," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2437–2451, 2022.

[55] Y. Chen, Y. Wang, and L. Yang, "SRP: A microscopic look at the composition mechanism of website fingerprinting," *Appl. Sci.*, vol. 12, no. 15, p. 7937, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/15/7937

[56] G. Cherubin, R. Jansen, and C. Troncoso, "Online website fingerprinting: Evaluating website fingerprinting attacks on Tor in the real world," in *Proc. 31st USENIX Secur. Symp.* Boston, MA, USA, Aug. 2022, pp. 753–770.

[57] M. Chen, Y. Wang, and X. Zhu, "Few-shot website fingerprinting attack with meta-bias learning," *Pattern Recognit.*, vol. 130, Oct. 2022, Art. no. 108739. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320322002205

[58] Q. Sun, D. R. Simon, Y.-M. Wang, W. Russell, V. N. Padmanabhan, and L. Qiu, "Statistical identification of encrypted web browsing traffic," in *Proc. IEEE Symp. Secur. Privacy*, May 2002, pp. 19–30.

[59] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy vulnerabilities in encrypted HTTP streams," in *Privacy Enhancing Technologies*, G. Danezis and D. Martin, Eds. Berlin, Germany: Springer, 2006, pp. 1–11.

[60] M. Liberatore and B. N. Levine, "Inferring the source of encrypted HTTP connections," in *Proc. 13th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2006, pp. 255–263, doi: 10.1145/1180405.1180437.

[61] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial Naïve–Bayes classifier," in *Proc. ACM Workshop Cloud Comput. Secur.*, New York, NY, USA, Nov. 2009, pp. 31–42, doi: 10.1145/1655008.1655013.

[62] X. Gong, N. Kiyavash, and N. Borisov, "Fingerprinting websites using remote traffic analysis," in *Proc. 17th ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2010, pp. 684–686, doi: 10.1145/1866307.1866397.

[63] L. Lu, E.-C. Chang, and M. C. Chan, "Website fingerprinting and identification using ordered feature sequences," in *Computer Security—ESORICS 2010*, D. Gritzalis, B. Preneel, and M. Theoharidou, Eds. Berlin, Germany: Springer, 2010, pp. 199–214.

[64] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," in *Proc. 10th Annu. ACM workshop Privacy Electron. Soc.*, New York, NY, USA, Oct. 2011, pp. 103–114, doi: 10.1145/2046556.2046570.

[65] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, "Touching from a distance: Website fingerprinting attacks and defenses," in *Proc. ACM Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2012, pp. 605–616, doi: 10.1145/2382196.2382260.

[66] 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 *Proc. IEEE Symp. Secur. Privacy*, May 2012, pp. 332–346.

[67] T. Wang and I. Goldberg, "Improved website fingerprinting on Tor," in *Proc. 12th ACM Workshop Workshop Privacy Electron. Soc.*, New York, NY, USA, Nov. 2013, pp. 201–212, doi: 10.1145/2517840.2517851.

[68] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *Proc. 23rd USENIX Secur. Symp.*, San Diego, CA, USA, Aug. 2014, pp. 143–157.

[69] Y. Shi and S. Biswas, "Website fingerprinting using traffic analysis of dynamic webpages," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 557–563.

[70] G. He, M. Yang, X. Gu, J. Luo, and Y. Ma, "A novel active website fingerprinting attack against tor anonymous system," in *Proc. IEEE 18th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2014, pp. 112–117.

[71] A. Kwon, M. AlSabah, D. Lazar, M. Dacier, and S. Devadas, "Circuit fingerprinting attacks: Passive deanonymization of Tor hidden services," in *Proc. 24th USENIX Secur. Symp.*, Washington, DC, USA, Aug. 2015, pp. 287–302.

[72] K. Alnaami, G. Ayoade, A. Siddiqui, N. Ruozzi, L. Khan, and B. Thuraisingham, "P2V: Effective website fingerprinting using vector space representations," in *Proc. IEEE Symp. Ser. Comput. Intell.*, Dec. 2015, pp. 59–66.

[73] J. Hayes and G. Danezis, "*k*-fingerprinting: A robust scalable website fingerprinting technique," in *Proc. 25th USENIX Secur. Symp.*, Austin, TX, USA, Aug. 2016, pp. 1187–1203.

[74] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, J. Pennekamp, K. Wehrle, and T. Engel, "Website fingerprinting at internet scale," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, 2016, pp. 1–15.

[75] H. Jahani and S. Jalili, "A novel passive website fingerprinting attack on Tor using fast Fourier transform," *Comput. Commun.*, vol. 96, pp. 43–51, Dec. 2016.

[76] H. Jahani and S. Jalili, "Online Tor privacy breach through website fingerprinting attack," *J. Netw. Syst. Manage.*, vol. 27, no. 2, pp. 289–326, Apr. 2019, doi: 10.1007/s10922-018-9466-z.

[77] R. Spreitzer, S. Griesmayr, T. Korak, and S. Mangard, "Exploiting data-usage statistics for website fingerprinting attacks on android," in *Proc. 9th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, New York, NY, USA, Jul. 2016, pp. 49–60, doi: 10.1145/2939918.2939922.

[78] K. Al-Naami, S. Chandra, A. Mustafa, L. Khan, Z. Lin, K. Hamlen, and B. Thuraisingham, "Adaptive encrypted traffic fingerprinting with bi-directional dependence," in *Proc. 32nd Annu. Conf. Comput. Secur. Appl.*, New York, NY, USA, Dec. 2016, pp. 177–188, doi: 10.1145/2991079.2991123.

[79] A. Panchenko, A. Mitseva, M. Henze, F. Lanze, K. Wehrle, and T. Engel, "Analysis of fingerprinting techniques for Tor hidden services," in *Proc. Workshop Privacy Electron. Soc.*, New York, NY, USA, Oct. 2017, pp. 165–175, doi: 10.1145/3139550.3139564.

[80] A. Mitseva, A. Panchenko, F. Lanze, M. Henze, K. Wehrle, and T. Engel, "POSTER: Fingerprinting Tor hidden services," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2016, pp. 1766–1768, doi: 10.1145/2976749.2989054.

[81] T. G. Ejeta and H. J. Kim, "Website fingerprinting attack on Psiphon and its forensic analysis," in *Digital Forensics and Watermarking*, C. Kraetzer, Y.-Q. Shi, J. Dittmann, and H. J. Kim, Eds. Cham, Switzerland: Springer, 2017, pp. 42–51.

[82] Z. Zhuo, Y. Zhang, Z.-L. Zhang, X. Zhang, and J. Zhang, "Website fingerprinting attack on anonymity networks based on profile hidden Markov model," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1081–1095, May 2018.

[83] Y. Qin and C. Yue, "Website fingerprinting by power estimation based side-channel attacks on Android 7," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng.*, Aug. 2018, pp. 1030–1039.

[84] N. Matyunin, Y. Wang, T. Arul, K. Kullmann, J. Szefer, and S. Katzenbeisser, "MagneticSpy: Exploiting magnetometer in mobile devices for website and application fingerprinting," in *Proc. 18th ACM Workshop Privacy Electron. Soc.*, New York, NY, USA, Nov. 2019, pp. 135–149, doi: 10.1145/3338498.3358650.

[85] X. Zeng, C. Kang, J. Shi, Z. Li, and G. Xiong, "A novel website fingerprinting method for malicious websites detection," in *Information and Communication Technology for Intelligent Systems*, S. C. Satapathy and A. Joshi, Eds. Singapore: Springer, 2019, pp. 723–730.

[86] Z. Zhang, C. Kang, G. Xiong, and Z. Li, "Deep forest with LRRS feature for fine-grained website fingerprinting with encrypted SSL/TLS," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, Nov. 2019, pp. 851–860, doi: 10.1145/3357384.3357993.

[87] Y. Meng and J. Fei, "Hidden service website response fingerprinting attacks based on response time feature," *Secur. Commun. Netw.*, vol. 2020, Dec. 2020, Art. no. 8850472, doi: 10.1155/2020/8850472.

[88] V. Ghiëtte and C. Doerr, "Scaling website fingerprinting," in *Proc. IFIP Netw. Conf., Netw.*, 2020, pp. 199–207.

[89] X. Ma, M. Shi, B. An, J. Li, D. X. Luo, J. Zhang, and X. Guan, "Context-aware website fingerprinting over encrypted proxies," in *Proc. IEEE Conf. Comput. Commun.*, May 2021, pp. 1–10.

[90] D. Kim, L. Ho, Y.-H. Kim, W.-G. Kim, and D. Hwang, "Poster: A pilot study on real-time fingerprinting for Tor onion services," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2021, pp. 1–3.

[91] A. Mitseva, J. Pennekamp, J. Lohmöller, T. Ziemann, C. Hoerchner, K. Wehrle, and A. Panchenko, "POSTER: How dangerous is my click? Boosting website fingerprinting by considering sequences of webpages," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2021, pp. 2411–2413, doi: 10.1145/3460120.3485347.

[92] M. Shen, Y. Liu, L. Zhu, X. Du, and J. Hu, "Fine-grained webpage fingerprinting using only packet length information of encrypted traffic," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2046–2059, 2021.

[93] M. Shen, Y. Liu, S. Chen, L. Zhu, and Y. Zhang, "Webpage fingerprinting using only packet length information," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[94] K. Wang, J. Zhang, G. Bai, R. Ko, and J. S. Dong, "It's not just the site, it's the contents: Intra-domain fingerprinting social media websites through CDN bursts," in *Proc. Web Conf.*, New York, NY, USA, Apr. 2021, pp. 2142–2153, doi: 10.1145/3442381.3450008.

[95] H. Mei, G. Cheng, W. Gao, and J. Chen, "Website fingerprinting on access network and core gateway," in *Proc. 17th Int. Conf. Mobility, Sens. Netw. (MSN)*, Dec. 2021, pp. 671–678.

[96] T. Okazaki, H. Kato, S. Haruta, and I. Sasase, "A website fingerprinting attack based on the virtual memory of the process on Android devices," in *Proc. 26th IEEE Asia–Pacific Conf. Commun. (APCC)*, Oct. 2021, pp. 7–12.

[97] C. Li, L. Nie, and L. Zhao, "RLTree: Website fingerprinting through resource loading tree," in *Network and System Security*, M. Yang, C. Chen, and Y. Liu, Eds. Cham, Switzerland: Springer, 2021, pp. 3–16.

[98] H. Zou, Z. Wei, J. Su, B. Zhao, Y. Xia, and N. Zhao, "PF: Website fingerprinting attack using probabilistic topic model," *Secur. Commun. Netw.*, vol. 2021, Oct. 2021, Art. no. 3265300, doi: 10.1155/2021/3265300.

[99] K. Zou, J. Shi, Y. Gao, X. Wang, M. Wang, Z. Li, and M. Su, "Bit-FP: A traffic fingerprinting approach for Bitcoin hidden service detection," in *Proc. IEEE 6th Int. Conf. Data Sci. Cyberspace (DSC)*, Oct. 2021, pp. 99–105.

[100] H. Cheng and R. Avnur, "Traffic analysis of SSL encrypted web browsing," Tech. Rep., 1998.

[101] A. Hintz, "Fingerprinting websites using traffic analysis," in *Privacy Enhancing Technologies*, R. Dingledine and P. Syverson, Eds. Berlin, Germany: Springer, 2003, pp. 171–178.

[102] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2014, pp. 263–274, doi: 10.1145/2660267.2660368.

[103] T. Wang and I. Goldberg, "On realistically attacking Tor with website fingerprinting," *Proc. Privacy Enhancing Technol.*, vol. 2016, no. 4, pp. 21–36, Oct. 2016, doi: 10.1515/popets-2016-0027.

[104] X. Gu, M. Yang, and J. Luo, "A novel website fingerprinting attack against multi-tab browsing behavior," in *Proc. IEEE 19th Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2015, pp. 234–239.

[105] Y. Xu, T. Wang, Q. Li, Q. Gong, Y. Chen, and Y. Jiang, "A multi-tab website fingerprinting attack," in *Proc. 34th Annu. Comput. Secur. Appl. Conf.*, New York, NY, USA, Dec. 2018, pp. 327–341, doi: 10.1145/3274694.3274697.

[106] Q. Yin, Z. Liu, Q. Li, T. Wang, Q. Wang, C. Shen, and Y. Xu, "An automated multi-tab website fingerprinting attack," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 6, pp. 3656–3670, Nov. 2022.

[107] W. Cui, T. Chen, C. Fields, J. Chen, A. Sierra, and E. Chan-Tin, "Revisiting assumptions for website fingerprinting attacks," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, New York, NY, USA, Jul. 2019, pp. 328–339, doi: 10.1145/3321705.3329802.

[108] J. Gong and T. Wang, "Zero-delay lightweight defenses against website fingerprinting," in *Proc. 29th USENIX Secur. Symp.*, Aug. 2020, pp. 717–734. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/gong

[109] W. Cui, T. Chen, and E. Chan-Tin, "More realistic website fingerprinting using deep learning," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 333–343.

[110] Z. Guan, G. Xiong, G. Gou, Z. Li, M. Cui, and C. Liu, "BAPM: Block attention profiling model for multi-tab website fingerprinting attacks on Tor," in *Proc. Annu. Comput. Secur. Appl. Conf.*, New York, NY, USA, Dec. 2021, pp. 248–259, doi: 10.1145/3485832.3485891.

[111] M. Chen, Y. Chen, Y. Wang, P. Xie, S. Fu, and X. Zhu, "End-to-end multi-tab website fingerprinting attack: A detection perspective," 2022, *arXiv:2203.06376*.

[112] R. Attarian, L. Abdi, and S. Hashemi, "AdaWFPA: Adaptive online website fingerprinting attack for Tor anonymous network: A stream-wise paradigm," *Comput. Commun.*, vol. 148, pp. 74–85, Dec. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419300763

[113] R. Attarian and S. Hashemi, "Investigating the streaming algorithms usage in website fingerprinting attack against Tor privacy enhancing technology," in *Proc. 16th Int. Iranian Soc. Cryptol. Conf. Inf. Secur. Cryptol. (ISCISC)*, Aug. 2019, pp. 33–38.

[114] Z. Zhu, G. Chen, Z. Zhang, M. Fang, Q. Song, and B. Mao, "Website fingerprinting attack through persistent attack of student," in *Proc. IEEE 7th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, Nov. 2021, pp. 78–82.

[115] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2014, pp. 227–238, doi: 10.1145/2660267.2660362.

[116] T. Wang, "High precision open-world website fingerprinting," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 152–167.

[117] T. Wang, "The one-page setting: A higher standard for evaluating website fingerprinting defenses," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2021, pp. 2794–2806, doi: 10.1145/3460120.3484790.

[118] M. Jiang, Y. Wang, G. Gou, W. Cai, G. Xiong, and J. Shi, "PST: A more practical adversarial learning-based defense against website fingerprinting," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[119] C. Hou, G. Gou, J. Shi, P. Fu, and G. Xiong, "WF-GAN: Fighting back against website fingerprinting attack using adversarial learning," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–7.

[120] A. M. Sadeghzadeh, B. Tajali, and R. Jalili, "AWA: Adversarial website adaptation," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3109–3122, 2021.

[121] J. Gong, W. Zhang, C. Zhang, and T. Wang, "Surakav: Generating realistic traces for a strong website fingerprinting defense," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 1558–1573.

[122] B. Sun, W. Yang, M. Yan, Y. Zhu, and Z. Bai, "A practical website fingerprinting defense approach with universal adversarial perturbations," in *Proc. 7th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Apr. 2022, pp. 752–760.

[123] B. N. Levine, M. K. Reiter, C. Wang, and M. Wright, "Timing attacks in low-latency mix systems," in *Financial Cryptography*, A. Juels, Ed. Berlin, Germany: Springer, 2004, pp. 251–265.

[124] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: Attacks and defenses," in *Computer Security—ESORICS 2006*, D. Gollmann, J. Meier, and A. Sabelfeld, Eds. Berlin, Germany: Springer, 2006, pp. 18–33.

[125] C. V. Wright, S. E. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis," in *Proc. Netw. Distrib. Syst. Secur. Symp.* 2009, San Diego, CA, USA, Feb. 2009, pp. 1–14.

[126] X. Cai, R. Nithyanand, and R. Johnson, "CS-BuFLO: A congestion sensitive website fingerprinting defense," in *Proc. 13th Workshop Privacy Electron. Soc.*, New York, NY, USA, Nov. 2014, pp. 121–130, doi: 10.1145/2665943.2665949.

[127] R. Nithyanand, X. Cai, and R. Johnson, "Glove: A bespoke website fingerprinting defense," in *Proc. 13th Workshop Privacy Electron. Soc.*, New York, NY, USA, Nov. 2014, pp. 131–134, doi: 10.1145/2665943.2665950.

[128] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," in *Computer Security—ESORICS 2016*, I. Askoxylakis, S. Ioannidis, S. Katsikas, and C. Meadows, Eds. Cham, Switzerland: Springer, 2016, pp. 27–46.

[129] T. Wang and I. Goldberg, "Walkie-talkie: An efficient defense against passive website fingerprinting attacks," in *Proc. 26th USENIX Secur. Symp.*, Vancouver, BC, USA, Aug. 2017, pp. 1375–1390.

[130] G. Cherubin, J. Hayes, and M. Juarez, "Website fingerprinting defenses at the application layer," in *Proc. Privacy Enhancing Technol. Symp. (PETS)*, no. 2, 2017, pp. 186–203, doi: 10.1515/popets-2017-0023.

[131] D. Lu, S. Bhat, A. Kwon, and S. Devadas, "DynaFlow: An efficient website fingerprinting defense based on dynamically-adjusting flows," in *Proc. Workshop Privacy Electron. Soc.*, New York, NY, USA, Jan. 2018, pp. 109–113, doi: 10.1145/3267323.3268960.

[132] E. Chan-Tin, T. Kim, and J. Kim, "Website fingerprinting attack mitigation using traffic morphing," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1575–1578.

[133] W. Cui, J. Yu, Y. Gong, and E. Chan-Tin, "Realistic cover traffic to mitigate website fingerprinting attacks," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1579–1584.

[134] W. Cui, J. Yu, Y. Gong, and E. Chan-Tin, "Efficient, effective, and realistic website fingerprinting mitigation," *ICST Trans. Secur. Saf.*, vol. 6, no. 20, Apr. 2019, Art. no. 161977.

[135] X. Liu, Z. Zhuo, X. Du, X. Zhang, Q. Zhu, and M. Guizani, "Adversarial attacks against profile HMM website fingerprinting detection model," *Cognit. Syst. Res.*, vol. 54, pp. 83–89, May 2019.

[136] W. De la Cadena, A. Mitseva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, and A. Panchenko, "TrafficSliver: Fighting website fingerprinting attacks with traffic splitting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2020, pp. 1971–1985, doi: 10.1145/3372297.3423351.

[137] W. De la Cadena, A. Mitseva, J. Pennekamp, J. Hiller, F. Lanze, T. Engel, K. Wehrle, and A. Panchenko, "POSTER: Traffic splitting to counter website fingerprinting," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Nov. 2019, pp. 2533–2535, doi: 10.1145/3319535.3363249.

[138] A. Abusnaina, R. Jang, A. Khormali, D. Nyang, and D. Mohaisen, "DFD: Adversarial learning-based approach to defend against website fingerprinting," in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 2459–2468.

[139] S. Henri, G. Garcia-Aviles, P. Serrano, A. Banchs, and P. Thiran, "Protecting against website fingerprinting with multihoming," *Proc. Privacy Enhancing Technol.*, vol. 2020, no. 2, pp. 89–110, Apr. 2020, doi: 10.2478/popets-2020-0019.

[140] K. Al-Naami, A. El-Ghamry, M. S. Islam, L. Khan, B. Thuraisingham, K. W. Hamlen, M. Alrahmawy, and M. Z. Rashad, "BiMorphing: A bi-directional bursting defense against website fingerprinting attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 2, pp. 505–517, Mar. 2021.

[141] M. S. Rahman, M. Imani, N. Mathews, and M. Wright, "Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 1594–1609, 2021.

[142] M. Imani, M. S. Rahman, and M. Wright, "Adversarial traces for website fingerprinting defense," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, Oct. 2018, pp. 2225–2227, doi: 10.1145/3243734.3278493.

[143] C. Hou, J. Shi, M. Cui, M. Liu, and J. Yu, "Universal website fingerprinting defense based on adversarial examples," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 99–106.

[144] C. Hou, J. Shi, M. Cui, and Q. Yang, "Attack versus attack: Toward adversarial example defend website fingerprinting attack," in *Proc. IEEE 20th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Oct. 2021, pp. 766–773.

[145] S. Huang, X. Ma, and H. Bian, "Effectively and efficiently defending shadowsocks against website fingerprinting attacks," in *Proc. 8th Int. Conf. Dependable Syst. Their Appl. (DSA)*, Aug. 2021, pp. 251–256.

[146] T. Luo, L. Wang, S. Yin, H. Shentu, and H. Zhao, "RBP: A website fingerprinting obfuscation method against intelligent fingerprinting attacks," *J. Cloud Comput.*, vol. 10, no. 1, p. 29, May 2021, doi: 10.1186/s13677-021-00244-8.

[147] S. Shan, A. N. Bhagoji, H. Zheng, and B. Y. Zhao, "A real-time defense against website fingerprinting attacks," 2021, *arXiv:2102.04291*.

[148] H. Li, N. Niu, and B. Wang, "Cache shaping: An effective defense against cache-based website fingerprinting," in *Proc. 12th ACM Conf. Data Appl. Secur. Privacy*, New York, NY, USA, 2022, pp. 252–263, doi: 10.1145/3508398.3511500.

[149] J. K. Holland and N. Hopper, "RegulaTor: A straightforward website fingerprinting defense," *Proc. Privacy Enhancing Technol.*, vol. 2022, no. 2, pp. 344–362, Apr. 2022, doi: 10.2478/popets-2022-0049.

[150] R. Tang, G. Shen, C. Guo, and Y. Cui, "SAD: Website fingerprinting defense based on adversarial examples," *Secur. Commun. Netw.*, vol. 2022, Apr. 2022, Art. no. 7330465, doi: 10.1155/2022/7330465.

[151] J. Liang, C. Yu, K. Suh, and H. Han, "Tail time defense against website fingerprinting attacks," *IEEE Access*, vol. 10, pp. 18516–18525, 2022.

[152] Y. Zhang, L. Yang, J. Jia, S. Ying, and Y. Zhou, "RAP: A lightweight application layer defense against website fingerprinting," in *Security and Privacy in New Computing Environments*, W. Shi, X. Chen, and K.-K. R. Choo, Eds. Cham, Switzerland: Springer, 2022, pp. 254–270.

[153] Z. Ling, G. Xiao, W. Wu, X. Gu, M. Yang, and X. Fu, "Towards an efficient defense against deep learning based website fingerprinting," in *Proc. IEEE Conf. Comput. Commun.*, May 2022, pp. 310–319.

[154] J.-P. Smith, L. Dolfi, P. Mittal, and A. Perrig, "QCSD: A QUIC client-side website-fingerprinting defence framework," in *Proc. 31st USENIX Secur. Symp.* Boston, MA, USA, Aug. 2022, pp. 771–789.

[155] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3905–3911, doi: 10.24963/ijcai.2018/543.

[156] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[157] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2242–2251.

PEIDONG LIU received the B.E. and M.E. degrees from Beihang University, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His current research interests include deep learning, website fingerprinting, and encrypted network traffic classification.

LONGTAO HE was born in China, in 1974. He received the B.E., M.S.E., and Ph.D. degrees from the Harbin Institute of Technology, China. He is currently a Senior Engineer (professor level) with the National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC). His research interests include internet measurement, network traffic classification, internet security, and mobile security.

ZHOUJUN LI received the Ph.D. degree from the National University of Defense Technology, Hunan, China, in 1999. He is currently a Professor with the School of Computer Science and Engineering, Beihang University, Beijing, China. His main research interests include concurrency theory and process algebra, formal analysis and verification of security protocols, information security, and data mining.