Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.072275

ARTICLE

On the Resilience of Traffic Features under Concept Drift in Hidden Service Fingerprinting

Xiaoyun Yuan[1,2,3,*], **Zhengge Yi**[1,2], **Jingxi Zhang**[1,2] **and Hairui Zhang**[3]

[1]School of Cyberspace Security, Information Engineering University, Zhengzhou, China
[2]Key Laboratory of Cyberspace Situation Awareness of Henan Province, Zhengzhou, China
[3]Former Army Artillery and Air Defense Academy Zhengzhou Campus, Zhengzhou, China

*Corresponding Author: Xiaoyun Yuan. Email: yuanxyzz@outlook.com

Received: 23 August 2025; Accepted: 06 January 2026; Published: 23 July 2026

ABSTRACT: Website Fingerprinting (WF) has emerged as a promising technique for identifying user access patterns to Hidden Services (HS). Despite growing interest in WF for HS, the absence of well-established foundations and systematic guidelines for feature selection undercuts the robustness of WF techniques in the face of concept drift. To address this gap, we present an empirical study focusing on feature resilience under concept drift in WF for HS. Specifically, we categorize features into network-specific and network-agnostic groups and quantify their information leakage potential via mutual information. We further assess each feature's resilience to concept drift by evaluating its distributional stability through Kullback-Leibler (KL) divergence. To support our analysis, we construct a dedicated dataset capturing HS traffic across multiple time windows. Our evaluation reveals that although often overlooked by existing studies, network-specific features—especially time-based ones—not only leak more identifiable information but also exhibit greater stability against distributional shifts over time. We further investigate the underlying root causes of this phenomenon, offering insights that expand the theoretical underpinnings of WF and guide future feature selection strategies.

KEYWORDS: Website fingerprinting; concept drift; Tor; feature analysis; information leakage; hidden services

1 Introduction

The Onion Router (Tor) provides millions of users worldwide with a low-latency and censorship-resistant platform for internet access [1]. Hidden Services, hosted within the Tor network, accept only inbound connections through Tor, which are identified via unique ".onion" URLs, and are operated independently from any publicly visible IP addres. Although Tor ensures end-to-end anonymity and encrypts payload content, it cannot fully obscure traffic metadata such as volume, timing, and packet direction. Such metadata remain vulnerable to WF attacks, enabling adversaries to infer which HS are being accessed and illustrating the potential for increasingly precise WF techniques.

Regardless of whether features are manually engineered based on domain knowledge in traditional machine learning or automatically learned via deep learning (DL), the robustness of these features against concept drift [2,3] has never been systematically verified. For instance, in DL models, if the underlying feature representation changes substantially over time, as in NetCLR [4] and Holmes [5], previously trained models can become obsolete, necessitating periodic fine-tuning and retraining for specific websites, which poses a resource-intensive labor. Other evaluations [6–9] overlook the highly dynamic nature of Tor

traffic—particularly for HS. In such scenarios, the inherent unpredictability of Tor circuit paths, fluctuations in server-side load, and temporal variations in user behavior contribute to significant shifts in feature distributions over time. Consequently, evaluating feature resilience under realistic, time-evolving conditions is essential to preserve the practical utility of WF in HS scenarios.

To address this gap, we adopt a typical approach to feature selection in website fingerprinting, which consists of the following steps. First, mutual information is computed for each feature to assess how much information it leaks about the target website, serving as an indicator of its discriminative power. Next, the temporal robustness of each feature is evaluated by measuring the Kullback–Leibler (KL) divergence across multiple time windows, capturing its stability under concept drift. Finally, by jointly considering both metrics, one can identify features that are not only highly informative but also temporally stable—i.e., features that remain both revealing and reliable in dynamic traffic environments.

To support our analysis, we construct a purpose-built dataset capturing HS access traffic under multiple controlled drift scenarios. Experimental results show that network-specific features not only leak more identifying information but also exhibit greater stability under distributional shifts compared to their agnostic counterparts. This finding is counterintuitive, as such features were traditionally considered less robust across general web environments. Moreover, in the context of end-to-end deep learning approaches to website fingerprinting, the importance of this feature type is often underappreciated or entirely overlooked [4,10].

We further investigate the architectural and protocol-level causes of this phenomenon, providing foundational insights that bridge the empirical gaps in WF research and support the development of more resilient, theory-driven WF models. Our results inform not only the design of future WF classifiers but also shed light on potential weaknesses in current HS anonymity guarantees, offering guidance for both attack and defense strategies.

The main contributions of this work are as follows:

1. To the best of our knowledge, this is the first systematic investigation into the behavior of Website Fingerprinting features under concept drift, with a specific focus on Tor-based hidden services.
2. We propose a novel feature-based comparative methodology that offers intuitive insights into the mechanics of existing website fingerprinting attacks and defenses. By adopting this approach, we gain new insights and address several critical open questions in the field. In particular, we analyze the reasons behind the varying success rates of different attacks under concept drift scenarios.
3. We conduct a comprehensive analysis of commonly used website fingerprinting features under conditions of concept drift. Our results identify a subset of features that maintain high stability and discriminative power over time—referred to as *robust features*. Leveraging these features enhances both the accuracy and resilience of website fingerprinting classifiers in the presence of concept drift. These findings provide practical guidance for designing more adaptive and effective attack and defense strategies.

Roadmap. We begin with a review of related work on WF in Section 2. Section 3 introduces the threat model, covering Tor Hidden Services and a formal model of concept drift. Section 4 presents an overview of traffic features and a framework for their evaluation. Section 5 outlines our evaluation methodology, which includes data collection and feature measurement. Finally, Section 6 concludes the paper with a summary of findings and suggestions for future research.

2 Related Work

Website Fingerprinting refers to the process of analyzing network traffic generated by a web-browsing client to identify the destination website being visited, even when encryption and proxy services are used to conceal this information. In prior research, various traffic features have been proposed that enable an adversary to distinguish between different websites. Since traffic identification fundamentally relies on the selection of informative features, the extraction and selection of such features remain among the most critical and challenging aspects of website fingerprinting. The following paragraphs provide an overview of the most influential WF attacks targeting the Tor network that have been developed in recent years. Our discussion is organized according to the types of traffic features each attack leverages.

2.1 WF Using Manually Crafted Features

One of the most important aspects of a successful website fingerprinting attack lies in the choice of the features used to train ML classifiers [6,7,11]. Among these, three attacks earned particular relevance by significantly outperforming earlier WF attempts, achieving over 90% accuracy when classifying websites in the closed-world setting. Specifically, the k-NN [7] leverages the k-Nearest Neighbors classifier and a feature set with over 3000 features, including the total transmission bandwidth and elapsed time, the number of incoming and outgoing packets, or packet ordering and bursts. The CUMUL attack makes use of a Support Vector Machine and 104 hand-crafted features, including the number of incoming and outgoing packets and total bandwidth used in each direction. The k-FP [6] introduces a combination of features used in previous attacks with novel traffic characteristics, leading to a systematic analysis of 150 features. The classifier works by building a probabilistic model that compares the observed packet sequence against stored fingerprints.

These early approaches demonstrated that carefully selected and engineered features can significantly enhance classification performance. However, they also revealed limitations such as high computational overhead due to large features set and reduced effectiveness in more realistic open-world settings where many classes are unknown during training.

2.2 WF Using Automatically Learned Features

In recent years, a trend has emerged of applying deep learning techniques to website fingerprinting, enabling the automatic extraction of features without relying on domain-specific knowledge. These methods typically employ neural network architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), or Long Short-Term Memory (LSTM) networks to learn complex patterns directly from raw or minimally processed traffic traces [8,10,12]. These models not only reduce the need for extensive feature engineering but also demonstrate superior generalization capabilities across different datasets and environments.

Despite their promising results, deep learning-based WF attacks still face challenges related to model interpretability, computational cost, and sensitivity to concept drift. For example, to work accurately, DL-based WF requires a dataset of significant size. Given that collecting a dataset and training a model must be done regularly, it is clear that the cost in terms of time and resources is substantial. These issues motivate ongoing research into hybrid approaches that combine the strengths of both manually crafted and automatically learned features.

2.3 WF for Hidden Service

Hidden Services (HS) serve as the primary mechanism for deploying applications on the Tor network. In recent years, some researchers have begun to investigate Website Fingerprinting targeting HS. Kwon et al. [13]

achieved fast and effective detection of hidden services by using only the first 20 cell sequences, with an accuracy rate of 99%. In [14], the authors collected access traffic from 482 HS and evaluated the performance of three WF methods: CUMUL, k-NN, and k-FP. The authors of [15] proposed a WF method called 2ch-TCN, in which both the direction and timing of data packets are used as input features. A two-channel temporal convolutional network is then employed to enhance classification performance. In [16], the authors extracted burst-based features from HS traffic and built a CNN-based classifier that also achieved high accuracy.

However, hidden service (HS) web pages differ significantly from those of regular websites on the surface web. They typically feature simpler structures, limited content diversity, and often rely on similar development templates. Additionally, the domain names of hidden services were updated from version 2 to version 3 in 2021. Consequently, extracting discriminative and effective features from HS traffic for accurate classification remains a significant challenge for researchers.

2.4 WF under the Condition of Concept Drift

Concept drift is a well-known challenge in machine learning applications, particularly in security domains where data distributions evolve rapidly due to changes in user behavior, network protocols, or adversarial tactics [3]. In the context of website fingerprinting, concept drift arises when the statistical properties of traffic features change over time, potentially undermining the effectiveness of both attack and defense strategies. Prior work has shown that even slight variations in packet timing or burst size can significantly impact classification accuracy [17].

However, few studies have explicitly examined how concept drift affects feature stability or evaluated mitigation strategies tailored to this scenario. Understanding how concept drift impacts feature dynamics in website fingerprinting is essential for designing robust classifiers and effective defensive measures. This motivates our study, which aims to evaluate feature robustness over time and provide insights into drift-aware modeling strategies in this critical privacy-preserving context.

3 Threat Model

3.1 Website Fingerprinting for Hidden Services

Tor began supporting hidden services in 2004, providing the foundational technology for the emergence of the Tor dark web. Today, the Tor dark web is one of the largest and most prominent darknet platforms, hosting a significant amount of sensitive and malicious content. Tor hidden services are online services that can only be accessed within the Tor dark web through notable domain names ending in `.onion`. The core components of the Tor dark web architecture include the client, directory servers, hidden service directories, onion routers (ORs), and the hidden service server itself, as illustrated in Fig. 1.

- **Client:** The client is a local program running on the user's operating system, known as the Onion Proxy (OP). The OP encapsulates user data into Tor cells and applies multi-layer encryption, providing anonymous proxy services for various TCP applications.
- **Onion Router (OR):** Onion routers act as relay nodes within the Tor dark web. By default, a Tor anonymous circuit consists of three ORs: Entry Node, Middle Node, and Exit Node. The Entry Node is typically chosen from among trusted Guard nodes to enhance security.
- **Entry Node (Guard Node):** This is the first relay in the circuit that receives traffic directly from a Tor client. For security reasons, Tor clients typically select their Entry nodes from a small set of stable and trusted relays called Guard nodes. This helps prevent certain types of attacks by ensuring consistent entry points into the Tor network.

- **Middle Node:** This relay serves as an intermediate hop in the circuit, passing traffic between the Entry and Exit nodes. Middle nodes add an additional layer of encryption and help ensure that no single relay knows both the source and destination of the traffic.
- **Exit Node:** In regular Tor circuits, this would be the final relay that connects to the destination server. However, in hidden services, the Exit nodes serve a different role—they act as the last relay before reaching either the hidden service directory or the hidden service itself, maintaining the end-to-end encryption within the Tor network.
- **Hidden Server:** Hidden services provide TCP-based applications such as Web and IRC services. These servers are protected by Tor's anonymity features and can only be accessed via a Tor client to utilize their TCP application services.
- **Hidden Service Directory:** Hidden service directory servers store and provide clients with information about hidden service introduction points (IPO) and public keys. This information is essential for establishing connections to hidden services.

When a client accesses a hidden service, it first establishes a three-hop circuit to connect to the hidden service directory server and retrieves the introduction point and public key information. The client then selects a rendezvous point (RPO) as the meeting point for communication between the client and the hidden service. The RPO information is sent to the hidden service via the introduction point. Both the client and the hidden service establish circuits to reach the rendezvous point, completing the six-hop circuit setup before initiating communication. Through this six-hop circuit, Tor users access hidden services while ensuring that no single node simultaneously knows both the Tor client's IP address, the hidden service's IP address, and the content of the transmitted data, thereby maintaining the anonymity of both parties.

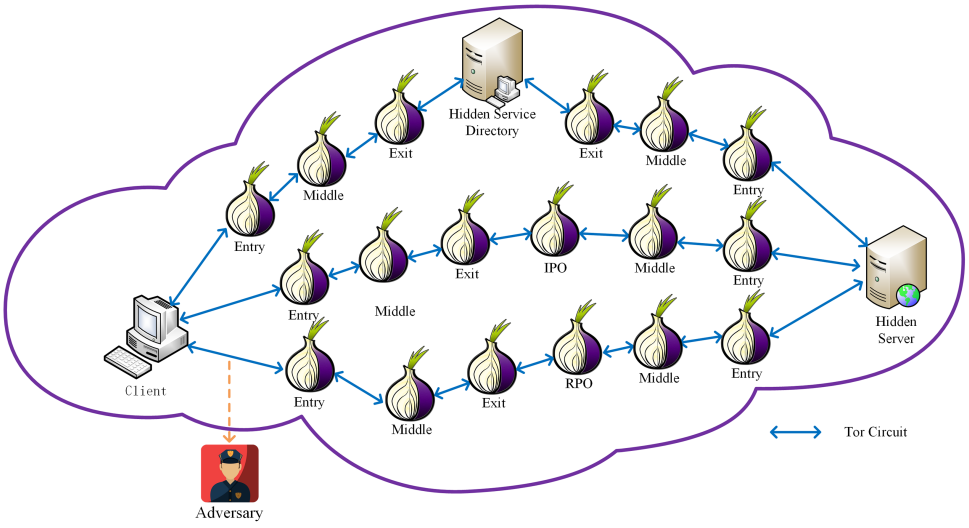

Figure 1: Tor hidden service.

A WF attack on the client side of Tor allows an adversary to observe a user's encrypted traffic to infer which website the user is visiting. They may monitor the communication link between the victim and the entry node or may control the entry node selected by the client to connect to the Tor network. The attack can be divided into two phases. During the training phase, the attacker visits each website multiple times, collecting traffic traces for each visit. Next, the collected traffic traces are labeled with their corresponding websites, and the features of these traces are extracted using manual or automatic methods. The classifier is trained on these collected traces. In the deployment phase, the attacker applies the classifier to new, unlabeled

traffic traces collected from victim-initiated communications and attempts to identify the websites visited by the client based on the classifier's output.

However, WF faces greater challenges under conditions of concept drift. Since the content and structure of target websites may change over time, features extracted during the training phase may become ineffective during deployment, resulting in a significant decrease in the classifier's classification accuracy. In the following, we present a formal modeling of concept drift.

3.2 Formal Modeling of Concept Drift

Most existing studies rely on a strong assumption that training and deployment data satisfy a condition of joint probability distribution stationarity:

$$P_{\text{train}}(\mathbf{X}, \mathbf{y}) \equiv P_{\text{deploy}}(\mathbf{X}, \mathbf{y}) \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{T \times d}$ represents a d-dimensional feature matrix over T time steps, and $\mathbf{y} \in \{0,1\}^C$ is a vector representing C classes of website labels. However, this stationarity assumption often fails to hold in real-world network environments, leading to significant degradation in model performance.

This discrepancy between training data and actual deployment data distributions is referred to as **concept drift**. Specifically, when the joint distribution between input features and target variables changes over time, existing predictive models fail to adapt to new data patterns, resulting in decreased classification accuracy, as illustrated in Fig. 2.

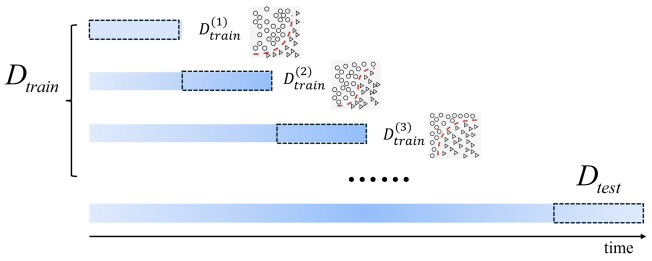

Figure 2: Concept drift illustration.

Formal Modeling and Mathematical Expression

In practical application scenarios, data is usually collected sequentially over time in a streaming format. Let the stream of data be represented as: $\mathbf{X} = \left\{x^{(1)}, x^{(2)}, \ldots, x^{(T)}\right\}$, where each sample at time point t, $x^{(t)} \in \mathbb{R}^m$, represents m input feature variables observed at that moment. The corresponding label sequence is: $\mathbf{y} = \left\{y^{(1)}, y^{(2)}, \ldots, y^{(T)}\right\}$. Traditional methods train models on historical data $\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{t}$ and attempt to make predictions on future test sets $D_{test}^{(t)} = \left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=t+1}^{t+\tau}$. The definition of concept drift between two timestamps t and $t+1$ is:

$$\exists x : p_t(x, y) \neq p_{t+1}(x, y) \tag{2}$$

where $p_t(x, y)$ denotes the joint probability distribution at time t.

To address this dynamic change, the prediction model should be formalized as an objective function capable of adapting to temporal evolution:

$$\min_{f^{(t)}, f^{(t+1)}, \ldots, f^{(t+\tau)}} \sum_{i=t}^{t+\tau} l\left(f^{(i)}(x^{(i)}), y^{(i)}\right) \tag{3}$$

where $f^{(i)}$ is the mapping function learned from historical training data $D_{train}^{(t)} = \left\{(x^{(i)}, y^{(i)})\right\}_{i=t-k}^{t-1}$ within a window size of k.

4 Feature Analysis

In this section, we present an overview of our approach to feature analysis. We begin by introducing the set of traffic features commonly used in website fingerprinting, along with their classification into different types based on their characteristics and sources. Following this, we describe the evaluation framework designed to assess the effectiveness, stability, and discriminative power of these features—particularly in the presence of concept drift.

4.1 Overview of Traffic Features

Traffic features form the foundation for distinguishing websites in website fingerprinting. They are commonly categorized into three types: temporal features, which capture timing behaviors such as inter-packet arrival times, burst durations, and session length and can vary based on a webpage's structure and content; directional features, which reflect the direction of packet flow (client-to-server or server-to-client) and help identify asymmetries that may indicate specific web activities; and volume-based features, which characterize the amount of data exchanged during a browsing session, including metrics such as total packet count, payload size, and packet size distribution. Each feature type contributes uniquely to the identification process and may demonstrate different levels of robustness under dynamic network conditions.

Real-world network environments are dynamic and unpredictable, with various factors influencing traffic behavior, such as traffic bursts, traffic engineering, partial network failures, and infrastructure updates. These environmental variations manifest primarily in the timing and ordering of packets. Website fingerprinting typically relies on two main categories of features:

1. **Network-specific features:** These are features that are highly dependent on the quality of the local network connection, such as packet timings and fragmentation patterns.
2. **Network-agnostic features:** These features are not affected by local network congestion. Examples include message lengths, transmission directions, and burst characteristics.

Network-specific features are closely tied to the local system and network conditions, making their generalization across different network environments unclear. Due to these limitations, recent studies tend to disregard timing information and instead focus on network-agnostic features. The latter can be highly characteristic, as the application data layer often determines the resulting ciphertext lengths.

Hayes and Danezis emphasized the importance of function and attribute selection for performing accurate WF, and proposed a manual function selection approach [6]. The types to which the first 20 features belong are shown in Table 1.

Table 1: Classification of features into network-specific and network-agnostic category.

ID	Feature Description	Category
1	Number of incoming packets	Both
2	Number of outgoing packets as a fraction of the total number of packets	Both
3	Number of incoming packets as a fraction of the total number of packets	Network-Agnostic
4	Standard deviation of the outgoing packet ordering list	Network-Agnostic
5	Number of outgoing packets	Both
6	Sum of all items in the alternative concentration feature list	Network-Agnostic

(Continued)

Table 1 (continued)

ID	Feature Description	Category
7	Average of the outgoing packet ordering list	Network-Agnostic
8	Sum of incoming, outgoing and total number of packets	Both
9	Sum of alternative number packets per second	Network-Specific
10	Total number of packets	Both
11–18	Packet concentration and ordering features list	Both
19	The total number of incoming packets stats in first 30 packets	Both
20	The total number of outgoing packets stats in first 30 packets	Both

4.2 Framework for Feature Evaluation

To systematically evaluate the usefulness of each feature, we propose a comprehensive framework that considers multiple dimensions, as shown in Fig. 3. Different features may carry different amounts of information. A common practice in evaluating the effectiveness of website fingerprinting is to extract traffic features from target websites, train a classifier using these features, and assess performance based on the classifier's accuracy. However, classification accuracy is inherently dependent on the choice and performance of the classifier rather than solely on the intrinsic quality of the features themselves.

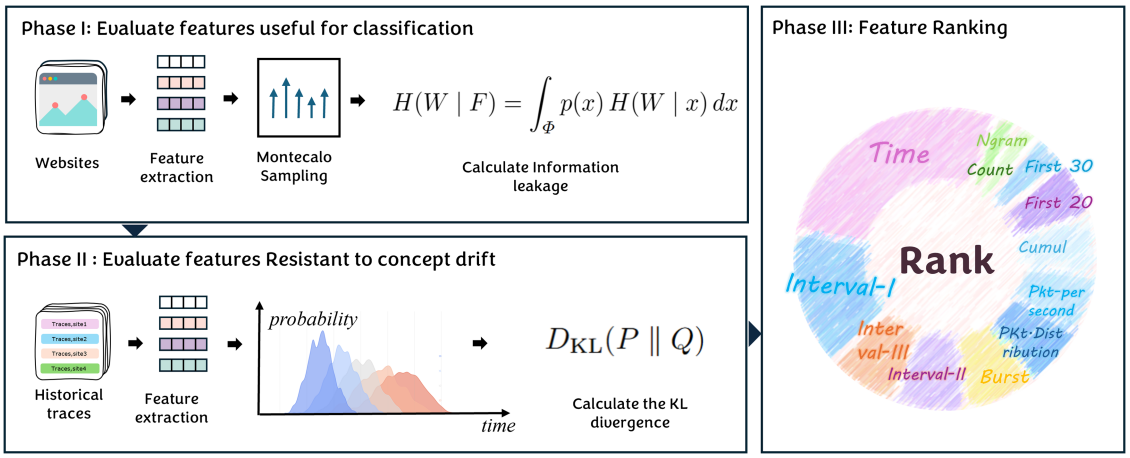

Figure 3: Feature evaluation method under concept drift.

Some features may carry substantial discriminative information but fail to be effectively utilized by a particular classifier. Therefore, it is essential to adopt classifier-agnostic feature evaluation methods to accurately assess the impact of concept drift on website fingerprinting features. Such approaches provide a more reliable measure of feature stability and informativeness over time. To assess the effectiveness of feature evaluation, we introduce concepts from information theory.

Phase I: Evaluate Features Useful for Classification

In information theory, given two random variables W: Indicates the website category (i.e., which website the user is visiting). F: Represents the features extracted from the traffic. The conditional entropy $H(W \mid F)$ quantifies the average uncertainty about the visited website W given the observed traffic features F. A lower value of $H(W \mid F)$ indicates that the features are highly informative about the website identity, while a higher value suggests greater ambiguity.

$H(W \mid F)$ quantifies the remaining uncertainty about the visited website after observing the traffic features. It is defined as:

$$H(W \mid F) = \int_{\Phi} p(x)\, H(W \mid x)\, dx \tag{4}$$

where $p(x)$ denotes the probability density of the feature vector x, $H(W \mid x)$ represents the uncertainty in the website class given a specific feature observation x, and Φ is a weighting or normalization factor that accounts for the contribution of different regions in the feature space.

If a feature F can effectively reveal the identity of a website W (e.g., through unique traffic patterns), then the conditional entropy $H(W \mid F)$ will be small. Conversely, if a feature does not provide significant information about the website, then $H(W \mid F)$ approaches the original entropy $H(W)$, indicating that the feature is either useless or has low discriminative power.

This can be naturally connected to *mutual information*, which quantifies the amount of information that the observed feature F provides about the website W. It is defined as:

$$I(W; F) = H(W) - H(W \mid F) \tag{5}$$

where $I(W; F)$ denotes the mutual information between W and F, $H(W)$ is the entropy of the website distribution, and $H(W \mid F)$ is the uncertainty remaining about W after observing F. A higher value of $I(W; F)$ indicates that the feature F is more informative about the visited website.

Phase II: Evaluate Features Resistant to Concept Drift

When concept drift occurs, the conditional entropy of website fingerprinting features—conditioned on the class label (i.e., website identity)—tends to change over time. To assess this behavior and identify features that remain both informative and stable under evolving conditions, we employ a combination of information-theoretic measures.

A key component of our evaluation is the measurement of information leakage using *adaptive kernel density estimation* (KDE). This method models the probability density function (PDF) of each feature or category of features by allowing each kernel to determine its optimal bandwidth independently. As a result, adaptive KDE can effectively capture both continuous and discrete distributions without prior assumptions about their underlying structure.

To compute the conditional entropy $H(C \mid \bar{f})$, we apply the Monte Carlo integration method. This approach approximates the definite integral by randomly sampling points from the domain of the function:

$$H(C \mid \bar{f}) \simeq \sum_{i=1}^{k} H(C \mid \bar{f}^{(i)}) \tag{6}$$

where $\bar{f}^{(1)}, \bar{f}^{(2)}, \ldots, \bar{f}^{(k)}$ denote the sampled feature vectors drawn from the distribution of $\bar{f}$.

Another critical measure in our analysis is the *Kullback-Leibler (KL) divergence*, which quantifies the difference between two probability distributions. Specifically, it measures how one distribution diverges from a reference distribution. For two discrete probability distributions P and Q, the KL divergence from P to Q is defined as:

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{7}$$

In the context of website fingerprinting, KL divergence enables us to evaluate the dissimilarity between feature distributions under different temporal or environmental conditions. For instance, if P represents

the feature distribution for a set of websites at time t_1, and Q corresponds to the same set at time t_2, then $D_{\mathrm{KL}}(P \parallel Q)$ provides a quantitative measure of how much the feature distributions have shifted over time. A lower value indicates higher stability and robustness of the feature concerning concept drift.

Phase III: Feature Ranking

Following the identification of features that remain stable under concept drift, we proceed to rank these features based on their overall effectiveness for classification. This ranking process aims to prioritize features that not only maintain high discriminative power but also exhibit strong robustness across different network conditions.

To achieve this, we define a composite scoring function that combines two key metrics:

- **Discriminative Power:** Measured using mutual information $I(C; f)$ between the class label C (i.e., website identity) and a given feature f. A higher mutual information score indicates that the feature is more informative about the class.
- **Stability over Time:** Quantified by the KL divergence $D_{\mathrm{KL}}(P \parallel Q)$ between feature distributions before and after concept drift, as computed in Phase II. Features with lower KL divergence are considered more stable.

  The composite score $S(f)$ for a feature f is defined as:

$$S(f) = \alpha \cdot I(C; f) - (1 - \alpha) \cdot D_{\mathrm{KL}}(P \parallel Q) \tag{8}$$

where $\alpha \in [0, 1]$ is a tunable parameter that controls the trade-off between discriminative power and stability. In our experiments, we set $\alpha = 0.5$ to weight both aspects unless otherwise specified equally.

Once all features are scored, we sort them in descending order of $S(f)$ to obtain a ranked list of features. The top-ranked features represent those that are most suitable for building classifiers that are resilient to concept drift. These features are then selected for use in the subsequent classification phase. This ranking mechanism provides a principled approach to guide feature selection in dynamic environments, such as Tor-based hidden services, where both accuracy and long-term generalization are crucial.

5 Evaluation

On 15 July 2021, Tor discontinued support for version 2 (v2) hidden service addresses. Therefore, we collected traffic from version 3 (v3) hidden services to construct our dataset. We first describe the data collection process, followed by an analysis of the features' potential for information leakage and their robustness against concept drift. Finally, we tested the top-ranked features using machine learning models, and the results demonstrated that the selected feature set significantly outperformed randomly selected features.

5.1 Data Collection Methodology

The website dataset used in this paper was collected by visiting the landing pages of the top 3000 websites worldwide, as listed on Alexa. Each website is visited 20 times using Google Chrome Version 61 on a desktop machine. Selenium WebDriver is used for web browser automation. When visiting a website, we set a 60-s timeout before closing the Chrome browser. Pages that fail to load within this period will be marked as a failure.

A successful visit is defined as an instance. In total, we successfully visited 2712 websites at least once within the 60 s. The total number of instances was 44,944. We collected data over two weeks. We noticed that the network is highly volatile. About 30% of all hidden services running an HTTP(S) server switch from

online to offline and vice versa within two weeks, as depicted in Fig. 4. Therefore, the websites that we can use to analyze concept drift are only 27.7% of the original.

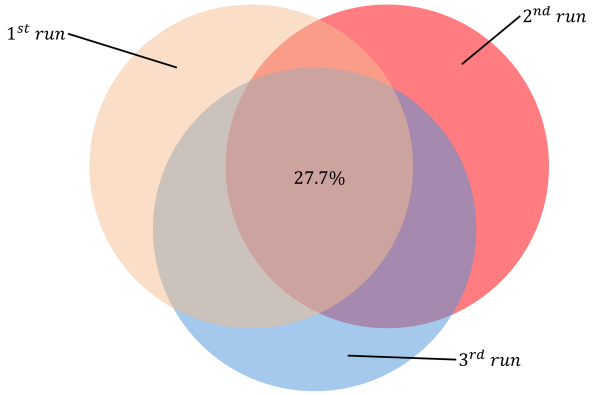

Figure 4: The Venn diagram shows the difference between three runs two week apart.

The 27.7% represents the proportion of websites that remained consistently accessible across all three runs (R1, R2, and R3) of our data collection process. From our initial set of 2712 websites, only approximately 751 websites (27.7%) maintained stable accessibility throughout the entire study period. This intersection of consistently available websites across all three runs forms the basis for our concept drift analysis, as it provides the necessary temporal continuity for meaningful comparison.

5.2 Feature Measuring

A packet sequence P can be represented as:

$$P = \langle (t_1, \ell_1), (t_2, \ell_2), \ldots, (t_n, \ell_n) \rangle$$

In this representation:

- t_i denotes the inter-packet time difference between the i-th and $(i-1)$-th packet, with $t_1 = 0$;
- ℓ_i represents the byte length of the i-th packet.

The length of the sequence, denoted $|P|$, is equal to n. We use P_t and P_ℓ to denote the subsequences containing only the inter-packet timing values and only the packet lengths, respectively. Packet direction is indicated by the sign of ℓ_i: a positive value denotes an outgoing packet, while a negative value indicates an incoming packet. Features are extracted from the package sequence. A classifier achieves success in distinguishing between two classes by identifying consistent differences in their feature representations. To facilitate comparison and analysis, we introduce the 14 feature categories that are commonly employed in state-of-the-art website fingerprinting attacks.

5.2.1 Evaluate Features Useful for Classification

In our data collection process, we gathered network traffic in two distinct formats: ‘whole’ and ‘pcap’. The primary distinction between these formats lies in their scope of capture. The ‘whole’ format provides a comprehensive record of the entire connection process, including all preliminary communication with Tor directory authorities, initial circuit establishment, handshake procedures, and the subsequent data transfer. This format offers researchers complete visibility into the end-to-end connection process but results in larger file sizes. In contrast, the ‘pcap’ (Packet Capture) format focuses exclusively on the actual data transfer phase,

excluding the preliminary setup processes and directory authority communications. It begins recording only after the circuit is established, resulting in more concentrated data collection and smaller file sizes. While the 'pcap' format is more storage-efficient and suitable for analyzing specific communication patterns, the 'whole' format proves invaluable for researchers studying the complete connection establishment process and overall Tor network behavior.

Fig. 5 illustrates the information leakage across different feature categories for two data formats: pcap and whole. The upper section depicts the information leakage patterns using the pcap format, encompassing features such as packet count, time, n-gram, transposition, multiple interval features, and others like packet distribution, burst traffic, first 20/30 packets, last 30 packets, packets per second, and CUMUL. Among these features, some show relatively stable information leakage trends, while others exhibit higher leakage at specific points.

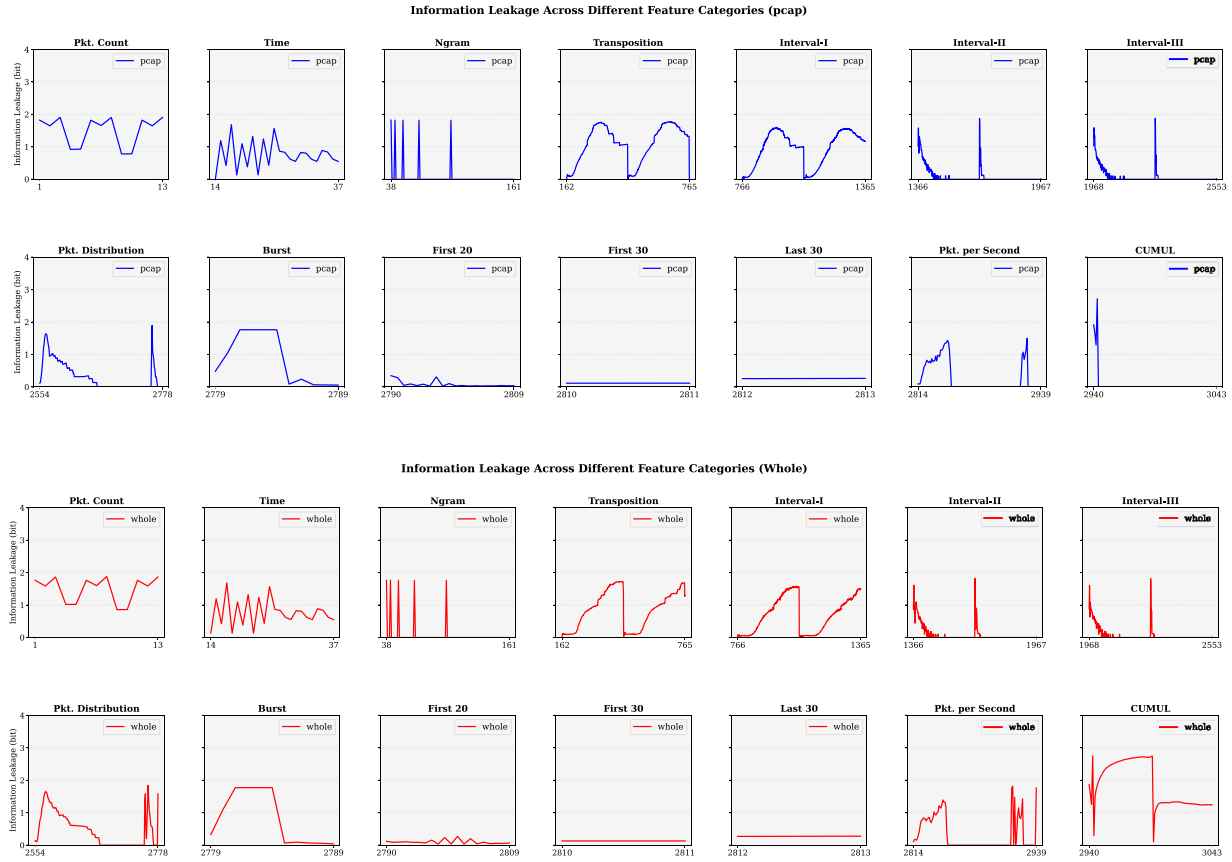

Figure 5: Comparative analysis of information leakage across feature categories for Pcap and whole formats.

The lower section corresponds to the analysis of information leakage under the whole format. Overall, compared to the pcap format, the whole format generally shows higher levels of information leakage. Particularly in features like packet count, n-gram, transposition, and the three distinct interval features, the whole format demonstrates higher leakage levels with greater variability. This suggests that the whole format, due to containing more interaction details and circuit establishment process information, is potentially more prone to information leakage.

Root Cause Analysis: IP circuits initiated by hidden services have significantly longer lifespans compared to other types of circuits. Specifically, the Duration of Activity (DoA) for these circuits typically lasts

around one hour [13], ensuring that hidden services remain continuously accessible via their IP addresses. These results highlight the advantages of using the whole format when analyzing onion services. Compared to regular circuits, those established over the Tor network for communicating with hidden services exhibit distinct behavioral patterns. A key characteristic of hidden services is their unique communication pattern, which involves three specific types of outgoing cells: two extend commands and one introduce command. Additionally, Tor assigns dedicated circuits for each hidden service connection, further enhancing the uniqueness and persistence of this type of communication.

5.2.2 Evaluate Features Resistant to Concept Drift

The three subplots presented in Fig. 6 depict the top 20 most stable features in resisting concept drift across multiple websites during different testing rounds—specifically, from Round 1 to Round 2, from Round 2 to Round 3, and the overall period from Round 1 to Round 3. These heatmaps provide a visual representation of how each feature maintains its performance under varying degrees of concept drift, with color intensity ranging from light red (indicating low stability) to dark blue (representing high stability).

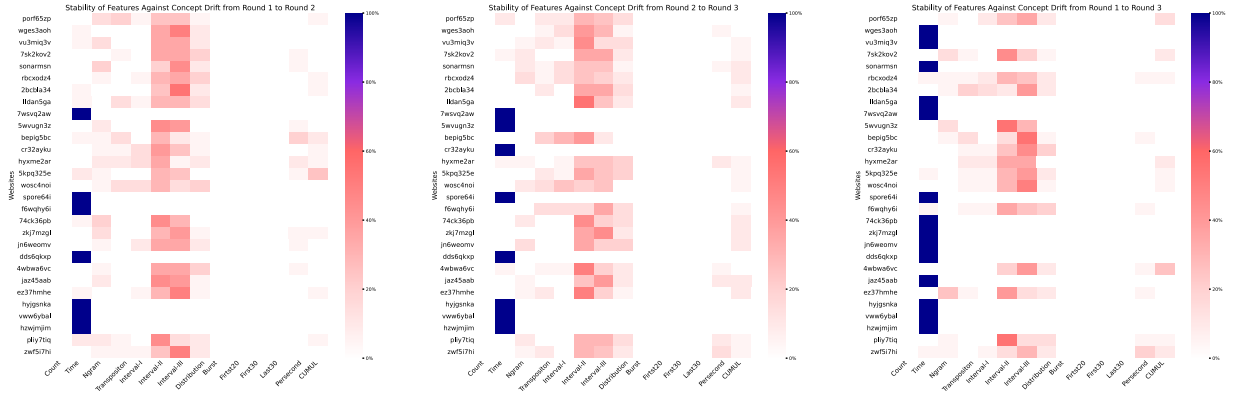

Figure 6: Stability of features against concept drift.

In the first subplot (Round 1 to Round 2), several features show consistent resilience against concept drift across the majority of websites, as evidenced by the prominent dark blue regions. Network-specific features, such as "Time," "Interval-I," and "Interval-II," demonstrate extreme stability throughout this period, indicating their robustness in adapting to both gradual and abrupt changes in data distribution. In contrast, certain features, such as "First20" and "First30", exhibit reduced stability on specific websites, highlighting their sensitivity to variations in traffic patterns.

The second subplot (Round 2 to Round 3) and the third subplot (Round 1 to Round 3) display similar patterns in feature stability, indicating a consistent resistance to concept drift over time. Furthermore, as illustrated in Fig. 5, time-based features in whole-format packets demonstrate a higher propensity for information disclosure. However, WF based on deep learning often overlook this feature. Therefore, they remain an underrecognized yet crucial feature category in traffic analysis.

In summary, these visualizations offer valuable insights into the comparative stability of different features when exposed to concept drift across diverse environments. The findings can inform the selection and design of robust features in real-world applications where data distributions are subject to change over time.

Root Cause Analysis: Upon loading a web page, the client initially requests the main HTML document. This document typically triggers subsequent requests for associated resources, such as scripts, images,

stylesheets, and other embedded components. These resources may be hosted on different servers. For each resource, the client establishes an HTTP connection to the respective server and retrieves the content sequentially or in parallel, depending on the browser's policies and the server's constraints.

In contrast, when accessing hidden services, most resources are usually fetched from a single server or a small number of servers. This architectural difference has a significant impact on traffic patterns. In regular websites, the use of multiple concurrent connections introduces a degree of packet order randomization, as different connections often retrieve distinct subsets of resources during each page load. As a result, packet sequences can vary considerably across different visits to the same website.

Timing features elucidate the temporal intervals and logical relationships between packets. In Tor circuits, the pathways utilized by clients frequently differ from those observed by attackers during model training in terms of latency, congestion, and bandwidth constraints. This discrepancy poses challenges for timing-based identification techniques, making it difficult to achieve stable performance in ordinary web browsing contexts. Conversely, for hidden services, timing features exhibit notable robustness, enabling them to maintain high discriminatory accuracy even in the presence of concept drift.

The reason lies in the relatively fixed communication paths involved in accessing hidden services. Tor establishes a persistent circuit architecture for hidden services, which minimizes latency variations resulting from path reconfiguration. This structural consistency enhances the stability of timing features across multiple visits. Consequently, while timing-based features are generally susceptible to environmental changes in traditional website fingerprinting settings, they exhibit superior discriminative capability and resilience to concept drift in the case of hidden services.

Experimental Validation of Feature Ranking

To evaluate the effectiveness of our feature ranking method, we conduct an experiment in which we train and test a classifier using subsets of features selected based on their scores $S(f)$. The goal is to assess whether higher-ranked features indeed lead to better classification performance—especially under concept drift scenarios.

We perform this evaluation in two settings:

- **Stable Environment:** Training and testing data are collected under similar conditions (i.e., no concept drift).
- **Drift Environment:** Training data is collected at an earlier period while testing data is collected after noticeable changes in traffic patterns (i.e., concept drift has occurred).

For both settings, we construct multiple feature subsets:

- *Top-ranked features*: The top 10%, 20%, and 50% of features according to $S(f)$.
- *Randomly ranked features*: A control group where features are randomly selected.
- *All features*: A baseline using the full set of available features.

We use a Random Forest classifier [6] for evaluation due to its robustness and ability to handle high-dimensional feature spaces. Classification accuracy and F1-score are used as performance metrics.

Fig. 7 illustrates the classification accuracy across different feature subsets in both stable and concept drift environments. It shows that models trained on the top-ranked features consistently achieve higher accuracy than those trained on randomly selected features, especially under drift conditions. Similarly, Fig. 8 compares F1-scores and further confirms the superior robustness of the top-ranked feature subset.

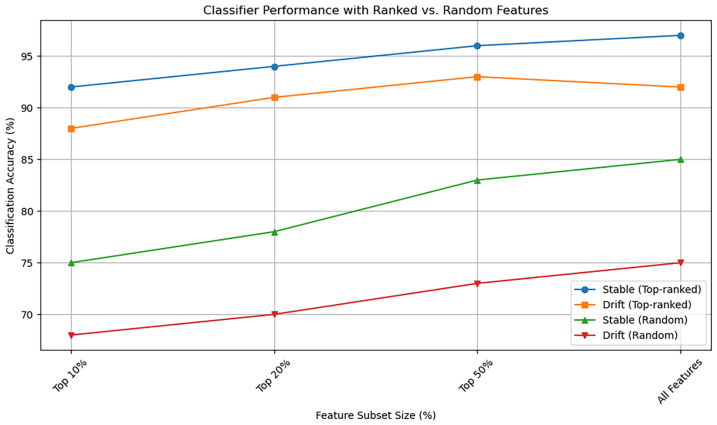

Figure 7: Classification accuracy comparison between top-ranked and random features.

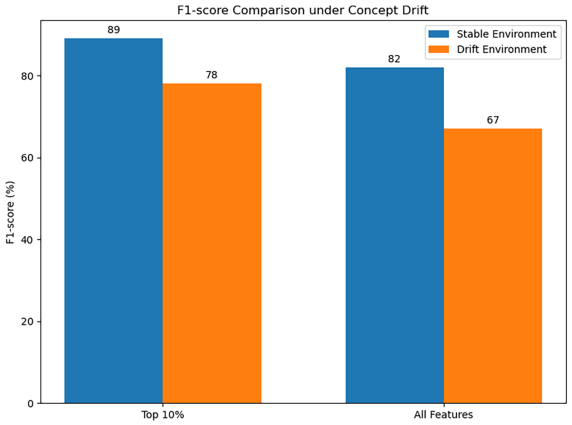

Figure 8: F1-scores of models trained on the top 10% ranked features vs. all features. The performance drop is significantly smaller for the top-ranked subset in the presence of concept drift.

The results show that models trained on top-ranked features consistently outperform those trained on randomly selected features across all subset sizes. More importantly, in the drift environment, the performance degradation of models using top-ranked features is significantly smaller than that of models using random or all features. This demonstrates that our ranking method successfully identifies features that are not only discriminative but also robust to environmental changes. These findings validate the utility of our composite scoring function and support its use in guiding feature selection for website fingerprinting under dynamic conditions.

6 Discussion

While deep learning methods have demonstrated remarkable performance in various classification tasks, their end-to-end nature often creates a “black box” effect, making it challenging to interpret how they arrive at their decisions. This lack of interpretability motivated our choice to initially employ traditional machine learning approaches for feature analysis. By using machine learning methods, we can systematically examine and understand the significance of individual features in the classification process, providing valuable insights into which network characteristics are most influential in identifying hidden services. This understanding is crucial for two reasons: first, it helps establish a more transparent and explainable

foundation for our analysis; second, it paves the way for developing more sophisticated hybrid models that combine the interpretability of traditional machine learning with the powerful pattern recognition capabilities of deep learning. Our approach allows us to leverage the best of both worlds—the explainability of machine learning for feature analysis and the potential for enhanced performance through future integration with deep learning techniques. This methodological choice aligns with our goal of not just achieving high classification accuracy, but also understanding the underlying patterns and characteristics that define different types of hidden services.

7 Conclusion

We proposed a feature analysis framework for website fingerprinting that identifies features which are both discriminative and robust to concept drift. By categorizing traffic features into temporal, directional, and volume-based types, and evaluating their stability using information-theoretic measures (e.g., conditional entropy, KL divergence), we introduced a composite scoring function to rank feature effectiveness over time.

Experiments on Tor hidden services show that timing-based features outperform others in classification accuracy and resilience to concept drift. This stems from: (1) fixed six-hop circuit paths that reduce delay variation; (2) consistent request-response patterns enhancing timing regularity; (3) longer circuit lifetimes minimizing behavioral fluctuations; (4) centralized content delivery reducing external network noise; and (5) sustained stability even under content or behavior evolution. These properties make temporal features particularly effective and durable for long-term fingerprinting attacks.

Acknowledgement: We gratefully acknowledge the technical support provided by the network operations team at the Information Engineering University for their assistance in setting up the experimental testbed.

Funding Statement: This work was supported by Henan Province Key Research, Development and Promotion Project on Tackling Key Scientific Problems (Grant No. 252102211087), the National Natural Science Foundation of China (U23A20305, and 62172435), the National Key R&D Program of China (Grant No. 2022YFB3102900), the Innovation Scientists and Technicians Troop Construction Projects of Henan Province (Grant No. 254000510007).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Xiaoyun Yuan and Zhengge Yi; methodology, Xiaoyun Yuan; validation, Jingxi Zhang; formal analysis, Zhengge Yi; writing—original draft preparation, Xiaoyun Yuan; writing—review and editing, Hairui Zhang; visualization, Jingxi Zhang. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available on request from the corresponding author.

Ethics Approval: This study does not involve human or animal subjects. Therefore, ethical approval is not applicable. All experiments were conducted in a controlled laboratory environment using simulated network traffic and publicly accessible Tor hidden services for research purposes. No personal or sensitive user data was collected or analyzed.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Arora A, Garman C. Improving the performance and security of Tor's onion services. Proc Priv Enhancing Technol. 2025;2025(1):531–52. doi:10.56553/popets-2025-0029.
2. Arora S, Rani R, Saxena N. A systematic review on detection and adaptation of concept drift in streaming data using machine learning techniques. WIREs Data Min Knowl Discov. 2024;14(4):1–27. doi:10.1002/widm.1536.

3. Malialis K, Li J, Panayiotou CG, Polycarpou MM. Incremental learning with concept drift detection and prototype-based embeddings for graph stream classification. In: 2024 International Joint Conference on Neural Networks. Piscataway, NJ, USA: IEEE; 2024. p. 1–7.
4. Bahramali A, Bozorgi A, Houmansadr A. Realistic website fingerprinting by augmenting network traces. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2023. p. 1035–49.
5. Deng X, Li Q, Xu K. Robust and reliable early-stage Website fingerprinting attacks via spatial-temporal distribution analysis. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2024. p. 1997–2011.
6. Hayes J, Danezis G. k-fingerprinting: a robust scalable website fingerprinting technique. In: 25th USENIX Security Symposium, USENIX Security 16. Berkeley, CA, USA: USENIX Association; 2016. p. 1187–203.
7. Wang T, Cai X, Nithyanand R, Johnson R, Goldberg I. Effective attacks and provable defenses for website fingerprinting. In: Proceedings of the 23th Conference on USENIX Security Symposium. Berkeley, CA, USA: USENIX Association; 2014. p. 143–57.
8. Sirinam P, Imani M, Juarez M, Wright M. Deep fingerprinting: undermining website fingerprinting defenses with deep learning. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2018. p. 1928–43.
9. Bhat S, Lu D, Kwon A, Devadas S. Var-CNN: a data-efficient website fingerprinting attack based on deep learning. Proc Priv Enh Technol. 2019;2019(4):292–310. doi:10.2478/popets-2019-0070.
10. Rimmer V, Preuveneers D, Juarez M, Goethem TV, Joosen W. Automated website fingerprinting through deep learning. In: Proceedings 2018 Network and Distributed System Security Symposium; 2018 Feb 18–21; San Diego, CA, USA. p. 1–15.
11. Wang T, Goldberg I. On realistically attacking tor with website fingerprinting. Proc Priv Enhancing Technol. 2016;2016(4):21–36. doi:10.1515/popets-2016-0027.
12. Sirinam P, Mathews N, Rahman MS, Wright M. Triplet fingerprinting: more practical and portable website fingerprinting with N-shot learning. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2019. p. 1131–48.
13. Kwon A, AlSabah M, Lazar D, Dacier M, Devadas S. Circuit fingerprinting attacks: passive deanonymization of tor hidden services. In: 24th USENIX Security Symposium (USENIX Security 15). Berkeley, CA, USA: USENIX Association; 2015. p. 287–302.
14. Overdorf R, Juarez M, Acar G, Greenstadt R, Diaz C. How unique is your. onion? An analysis of the fingerprintability of tor onion services. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York, NY, USA: ACM; 2017. p. 2021–36.
15. Wang M, Li Y, Wang X, Liu T, Shi J, Chen M. 2ch-TCN: a website fingerprinting attack over tor using 2-channel temporal convolutional networks. In: 2020 IEEE Symposium on Computers and Communications (ISCC). Piscataway, NJ, USA: IEEE; 2020. p. 1–7.
16. Rahman MS, Sirinam P, Mathews N, Gangadhara KG, Wright M. Tik-Tok: the utility of packet timing in website fingerprinting attacks. Proc Priv Enh Technol. 2020;2020(3):5–24. doi:10.2478/popets-2020-0043.
17. Deng X, Yin Q, Liu Z, Zhao X, Li Q, Xu M, et al. Robust multi-tab website fingerprinting attacks in the wild. In: 2023 IEEE Symposium on Security and Privacy (SP). Piscataway, NJ, USA: IEEE; 2023. p. 1005–22.